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

D4.6 Final Multiplatform interoperability and scalability framework

Version:	Final
Date:	28/08/2020
Authors:	Kendraio
Type:	Report
Dissemination:	Public

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

Table of Contents

1 Overview

This deliverable documents the interoperability and scalability framework of Bloomen. Here we describe how all Bloomen components talk to each other, both data formats and functional calls (APIs). In this 2nd cycle we document progress from the development of the Bloomen system as it relates to the interoperability and scalability framework that was planned from the 1st cycle.

Bloomen has been developed in order to be as open as possible. Interoperability is a cornerstone in the design and development of the Bloomen system. This is achieved using open source tools, and with data formats that adhere to open standards, or are described using open standards.

In this task, a specific and open data model scheme was designed to support the various requirements from WP2. Deriving schemas from the requirements. Using existing standards and standardisation across the pilots. In particular, the appropriate blockchain data structures are implemented for allowing common information reference frameworks for integrating the different ICT modules. This information framework was based on standard representation formats through JSON, XML, RDF, languages and frameworks, ensuring thus a standardized and general purpose form of exchanging data.

An open and easily manageable API was developed for bridging functional components of different integrated systems. This API facilitates interoperability and communication between components within the whole Bloomen architecture. The API is a standards-based HTTP RESTful APIs, in addition to the RPC style API that is provided by the blockchain nodes for direct programmatic access to transaction functionality of the blockchain network. The exchange of data between components is standardised across the pilots. The API is documented in an open format (Open API/Swagger).

This interoperability framework enables a standard way for a high-level communication and interaction protocol between diverse systems ensuring thus the necessary interoperability for the Bloomen system.

1.1 Motivation

The main purpose of the interoperability aspect of this task is to identify commonality to ensure consistency across the whole system.

Other benefits:

- interoperability with third parties (promotes adoption)
- adoption of existing standards (open) rather than re-inventing our own reduces costs associated with creating and maintaining software

The analysis presented here was used to guide decision making by technical partners in order to increase interoperability. The elements that have been implemented are documented below.

1.2 Methodology

The analysis and planning for interoperability and scalability applies across the whole Bloomen architecture. We have summarised the work conducted across the whole project as part of this deliverable. This is a follow up report that documents the continuation of the work and provides an update to the state of interoperability and scalability that was documented in the first cycle of this task.

We have chosen to analyse interoperability and scalability from the initial requirements, and through the design and implementation of the various system components. This starts with an analysis of the functions, processes and workflows set out in the requirements phase of the project, as well as further requirements discovered through the iterative development processes that are being used. From this analysis of functionality, we have documented the requirements for data models, schemas, and formats used across the whole project. In particular, the use of standard identifiers has been encouraged to promote interoperability and compatibility with other systems.

A reference section is provided that catalogues existing technologies, standards, and formats that are of interest to the Bloomen project, and all three pilots.

2 Functions and Workflows

The following sections present high-level operations, based on the requirements for the Bloomen system as a whole, as they relate to interoperability. Details are included, where applicable, on the interoperability between the pilots and/or between the components of the system. Additional interoperability with external or 3rd party systems is also documented where applicable.

2.1 Internal Facing

This section details categories of activity, for the purpose of identifying commonalities between features. These are taken from the requirements, but grouped on the basis that they are either:

- directly related to interoperability within the Bloomen system, or
- related similar features from each use case
- operate on identified data model

2.1.1 Register users

The following requirements relate to registering users with the system:

- UC-MUSIC-1 Register a user through a CMO
- UC-PHOTO-1 Creator and User Identity
- UC-WTV-1 - Access WebTV System

It could have been a great benefit for users to have a single identity across multiple pilots, however, the bigger challenge was to maintain a single identity across the various systems that each pilot needs to connect to. This allowed for easier mapping of external entities to the Bloomen internal schema. The use of unique identifiers was used not only for individuals but also for organisations and other entities.

The use of unique identifiers reduced the number of errors, conflicts and the need for manual processes making it more scalable and cost-effective, resulting in a system that allows content creators to keep more of the revenue generated from their works.

2.1.2 Register and upload content/assets

The following requirements relate to registering content and assets with the system:

- UC-MUSIC-2 Register Sound Recording or Musical Work
- UC-MUSIC-3 Register Sound Recording or Musical Work in batches
- UC-PHOTO-2 Upload image and create Blockchain hash

The system handles the uploading of recordings, musical works for the music pilot and images for the photo pilot. While the two pilots handle different types of data, the overall process is common:

1. Check the asset is not already registered.
2. Register the asset
3. Create a blockchain hash and receive a unique identifier.

The above suggests that the main difference between the two pilots is the data format (schema) and the type of resource. It is, however, important to note that both pilots communicate with a number of services and each of these services use a different schema and a different workflow. This works by using an interoperability layer with two main functions:

1. Map between schemas so data can be automatically transformed between standards.
2. Allowing to configure the workflows so they allow for handling of the differences and exceptions between the target systems.

The mapping to the internal Bloomen format uses open standards wherever possible and, as we are communicating with external entities, Collectives (CMOs), Dropbox, these external standards are not in our control. However, those systems make use of industry standards for identifiers like ISRC and ISWC for music, and these are used for the identification and mapping of resources.

2.1.3 Edit metadata

The following requirements relate to editing metadata within the Bloomen system:

- UC-MUSIC-5 Edit core metadata of a musical asset
- UC-MUSIC-6 Request edit of core metadata of a musical asset
- UC-MUSIC-7 Link musical assets
- UC-MUSIC-8 Merge duplicate musical assets

Many of the requirements for the above are not supported natively by Collectives (CMOs) and even when they are, they are not consistent.

Some of those requirements are:

- Permission management
- Reputation mechanisms
- Handling of notifications
- Changelog
- Asset relationships that are not supported by the Collectives (CMOs).

In order to achieve this, we add the required metadata to the Bloomen internal schema and add workflow processes around the CMO's workflows.

The rest of this metadata may be use case specific, but commonalities between the use cases were explored and consolidated.

2.1.4 View/search/browse

The following requirements relate to viewing, searching and browsing content:

- UC-MUSIC-4 Search assets
- UC-PHOTO-3 View images through API in CMS
- UC-WTV-1 Access WebTV System
- UC-PHOTO-8 View image analytics

In order to create more generic functionality that will scale with increased use cases (more target systems / Collectives), we extend the internal Bloomen schema with metadata that supports the low-level functions. This metadata will include:

- Endpoints of the APIs
- Mapping of input parameters
- Mapping of output parameters

As some of the use cases are specific to a single service ("View images through API in CMS") those were hardcoded for the specific pilot.

2.1.5 Rights metadata

The following requirements relate to the management of rights metadata:

- UC-MUSIC-9 Claim rights over a musical asset
- UC-MUSIC-10 Claim rights over a musical asset in batch
- UC-MUSIC-11 Resolve claim conflicts
- UC-MUSIC-15 Rights Metadata Privacy Control
- UC-MUSIC-12 Rights Management
- UC-MUSIC-13 Rights Versioning
- UC-MUSIC-14 Rights Collaboration
- UC-PHOTO-9 Rights management for photos
- UC-PHOTO-10 Rights Versioning for photos
- UC-PHOTO-11 Rights Collaboration for photos
- UC-PHOTO-12 Rights Metadata Privacy Control for photos
- UC-WTV-4 Rights Versioning for media
- UC-WTV-5 Rights Collaboration for media
- UC-WTV-6 Rights Metadata Privacy Control for media

The above requirements can be broken down into a number of generic areas:

- Control and management of rights data (this includes Collaboration)
- Workflows
- Privacy Control
- Versioning
- Digital signatures / blockchain

Some of the real-life processes involved with claims and conflicts are complex and even more so if involving multiple Systems, like dealing with more than one Collective (CMO).
As such we recommended the following:

1. Develop low level functionality that supports the above requirements.

2. Separation of lower level functionality from the workflows
3. Make the workflows driven by metadata as much as possible.
4. Use interoperability mapping to the Bloomen internal schema for all inputs and outputs.

The separation of the business layer allowed us to develop robust low-level functionality separate from the real-life business processes. As such, much of the achieved use cases were a proof of concept. Having the business logic driven by metadata allowed for faster adoption of changes and new opportunities, allowing for the development of real-life use cases in a scalable way.

2.1.6 Blockchain transactions

The following requirements relate to management of blockchain transactions:

- UC-PHOTO-4 Smart contracts
- UC-PHOTO-5 Pay for image through Smart Contract
- UC-PHOTO-6 Add licenced consumer
- UC-PHOTO-7 Vote contributor “Reputation over time”
- UC-PHOTO-13 Security and Privacy
- UC-WTV-2 The Core B2C Relationship
- UC-WTV-3 The Core B2B Relationship

The above requirements focus on the smart contract and blockchain transactions. These were all new concepts that did not need much in the way of interoperability at this stage.

Scalability, however, is essential for wide adoption of these systems. This is covered below in section 6.1 Bloomen blockchain scalability analysis.

OpenZepplin (see references section below) provides an open-source framework for implementation of smart contracts.

2.2 External Facing

This section lists processes that are part of the Bloomen system, but related to operations that are with third parties. They are detailed here because they relate to external interoperability of the Bloomen system.

2.2.1 Music

The Music pilot enabled users to manage Claims:

- A Claim is also referred to as "change request" in some documentation
- The workflow is described in 3.1.5.5 and 3.1.5.6 of D2.2.
- Declaring rights and resolving conflicts for musical works is done via CWR, a common standard for publishers and CMOs[1]. For Sound Recordings, there are the recent Recording Data and Rights Standards (formerly MLC Standards). [2]

The Music App communicates with the following external systems:

- It connects with the decentralized system of Alastria, which is a permissioned blockchain network. The interoperability occurs in a two-way schema by deploying different smart contracts and storing copyright claim information on-chain (in the smart contract storage), and by retrieving the claim and user data from the blockchain ledger.
- It interacts with the Bloomen API in order to fetch music asset data from the API's MongoDB inside the application itself, when an end-user is browsing the different music assets. This interoperability is one-way, since the Music App only reads from the Bloomen API.
- It interacts with the Kendraio application through the Bloomen API to load or edit repertoire (musical works and sound recordings metadata) to the system.
- It interacts with the end-user device. This is a two-way interoperability case since the end-user can experience the full application functionalities and capabilities, while they can not only interact by providing the obvious navigation and claim management mouse input, but also they are able to upload a CSV type of file with copyright claims that are inserted as new input to the system.

Data interoperability:

- Adding data to the system can be done both manually by using the graphical interface or in bulk, loading formatted files. There are two types of input data: the collections of musical assets, which can be imported or edited through the Kendraio app or directly using the Bloomen API, and the rights claims data, which can be added through the Music app. The Kendraio app features a flexible system of adaptors that makes it possible to potentially load data in any standard format (CWR or DDEX) into the Bloomen platform using the Bloomen API.

[1] https://members.cisac.org/CisacPortal/documentLink.do?sw=off&id=35801&usg=AOvVaw2pEt61wYheUVRu-KX-hhwf

[2] https://kb.ddex.net/pages/viewpage.action?pageId=1344848
https://kb.ddex.net/display/HBK/Updating+a+Claim
https://kb.ddex.net/display/HBK/Handling+Conflicts

- The Music App user management is organised in three types of users; the Super admin, the Admin and the User. The system administrator creates the Super admin users. The Super admins are able to manage an Admin user request (accept or reject) and allow or deny access into the system. In a similar manner, when a User requests access, the corresponding Admin user is managing their request.
- The Bloomen tools are also providing APIs for third party platforms to read and write data on the blockchain. This allows for secure and proper copyrights management for all parties involved.
- The platform uses the international standard identifiers of the musical assets managed by Bloomen Music: ISWC for the Musical Works and ISRC for the Sound Recordings (see 5.2), which facilitates the matching of any data with third party systems.

2.2.2 Photo

Blockchain technology enables a number of very important technical options for Bloomen Photo: The ability to onboard users (photographers) who we do not know nor “trust”. The upload of photos with their meta-data, meaning that we know who made the photos (or who claimed it), when, where, and other information. Further, and this would be a breakthrough in digital asset management: Photo licences can be connected to the creator AND the licensee - meaning that five years after a purchase the retrieval of the license terms is possible and can be done quickly. This is the “business” view, now follows the technical view.

The Photo Pilot, as part of the Bloomen Platform, is built with interoperability and ease of integration by external users and services in mind. It is based on a REST API so whoever wants to be part of the system just has to attain a JWT token, following the authentication workflow, and is free to interact with any parts of the pilot needed. The endpoints for interacting with the API as well as the data schemas are provided by the platform through swagger docs.

The user management for the pilot is based on an RBAC model with each role having different rights and permissions and with the possibility to expand on these in the future. For the Photo Pilot the main roles are these of the publisher and the photographer.

Lastly the Photo Pilot uses the blockchain wallet provided by the Bloomen platform which could be accessed at a lower level through direct connection with the blockchain layer of the platform, or for ease of use through the API methods as shown in the swagger docs of the platform.

2.2.3 WebTV

The 2nd iteration of the WebTV pilot was developed in such way that it could be integrated with various outside systems:

- The Bloomen WebTV tools communicate with the decentralized system of Alastria, which is a permissioned blockchain network. The interoperability occurs in a two-way schema by deploying different smart contracts and storing copyright claim information on-chain (in the smart contract storage), and by retrieving the claim and user data from the blockchain ledger.
- The Mobile Wallet (BloomenApp) is meant to be used for allowing devices to preview content that is protected by copyrights. This means that there is a functionality for a cryptographic handshake between the wallet and the device (through a dedicated app or a web browser), which is done through a zero token transfer between the wallet and the browser, which essentially tells the browser that the user is allowed to watch the content.
- The Bloomen tools are also providing APIs for third party platforms (WebTV Providers) to read and write data on the blockchain. This allows for secure and proper copyrights management for all parties involved.

3 Schemas and Formats

This section looks at the data models from the requirements, with a specific view to the needs for interoperability. The choice of schemas and formats used for the data models reflects the interoperability both internally within Bloomen, and externally with other systems.

The requirements for the three pilots are specified in D2.2 where data models are identified. This section explains each data model, with needs pulled from all three pilots. The required properties are derived from the requirements, and common properties are identified.

It is essential for interoperability that common identifiers are used. We present relevant identifiers below with each data model, with further information included later in the document in the References section. In addition to supporting industry standard identifiers, a hashed unique identifier is assigned to each content item and pushed to the blockchain.

3.1 User

Requirements for the user data model are mentioned in use cases across all three pilots, as each pilot has users within the system. The operation of users across pilots

differs, but there are some commonalities that are applicable to the discussion of interoperability.

In D2.2 each of the pilots specify a data model that contains all necessary information for the user of the system.

Common properties across all three pilots include:

- Authorisation data
- Name and Address
- Role (Consumer, Contributor)

In addition to the common properties the Photo use case specifies further properties:

- Settings (privacy, payments, etc)
- Reputation (reputation of the user, can be applied to both creators and media organisations over time)

The WebTV use case specifies additional properties:

- Financial information
- Wallet data and public addresses

The use of standard identifiers allows Bloomen systems to interoperate with third parties, as well as reconcile user objects with external sources of data, if required. Identifiers to be used include:

- IPI (Interested party information) is a unique identifying number assigned by the CISAC which is applicable to the Music pilot. More information is detailed below in section 5 - References.
- The DDEX Party Identifier standard describes an identification system used to identify each sender and receiver of a DDEX message and how the identification allocation process is undertaken.

3.2 Assets/Metadata

An assets data model is mentioned in all three pilots. From the requirements in D2.2 it is clear that each of the pilots have different needs around the storage of assets.

From the music pilot, the “music” asset represents Musical Works (MW) or their Sound Recordings (SR). They are defined by a set of core metadata (international identifiers, title, contributors, etc), and they have rights holders attached to them.

The requirements in D2.2 for the photo pilot specifies the following properties for an asset:

- URL (public file URL)
- Type of asset: UGC, photo, special photo
- Rights (list of users that have rights using this file)
- Owner (the owner of the file)
- Date/time added
- Price (price to pay for publishing rights)
- Usage rights time (how long?)
- Usage rights region (where in the world?)
- Analytics (number of views, likes, etc)
- Keywords
- Description
- Geo-coordinates
- Hash for organisation

The WebTV pilot specifies an entity that represents copyrighted content available for commercialization, distribution and access, with the following properties:

- Video Content:
- Video Title
- Year of production
- Production company name
- Available subtitles (by language)
- File location (URL)
- Tags for indexing
- Video Analytics (Views, Likes, etc)
- Hash

The music pilot requires the registration and management of metadata about assets only, the actual assets themselves are not required for operation of the system. The photo and WebTV pilots require the digital assets to be available.

Common properties across the data models include:

- A hash for identification
- Title
- External IDs for reconciliation with other systems

The photo and WebTV use case also have file location (URL) as a common property, but this does not exist in the music use case.

Standard identifiers for use by assets in the Bloomen system include (further details are provided in the references section below):

- ISRC - for sound recordings
- ISWC - for musical works

Existing standards (further details in the references section below):

- DDEX provides a suite of standards that are applicable to the music assets
- CWR is an addition standard for Musical works
- Relevant Schema.org schemas are: Music Recording, Music Composition, Photograph, Movie, and Episode

3.3 Rights Metadata

Rights metadata is mentioned across all pilots, and relates to many requirements, as well as required for tasks being undertaken in Work Package 3 related to Copyright management, monitoring and reporting.

In the photo pilot, there are properties of the asset metadata that are related to rights:

- Price (price to pay for publishing rights)
- Usage rights time (how long?)
- Usage rights region (where in the world?)

3.4 Merged asset

Music pilot requirements reference a merged assets data model that contains all the information relative to an asset that has been merged into another, whereby the original asset becomes deprecated. As this is specific to the music pilot, this is not considered for interoperability.

3.5 Asset Link

The Music pilot requirements reference a data model for an asset link. This is an entity that represents a link between assets. Links can be between entities of the same type (i.e. SR to SR / MW to MW), or entities of different types (i.e. MW to SR).

The properties of this asset include:

- Source asset
- Target asset
- Relation (e.g. underlying composition, radio edit, remaster...)

As this is specific to the music pilot, this is not considered for interoperability.

3.6 Rights holder

The requirements for all pilots from D2.2 include data model requirements for rights holders.

From the requirements for the music pilot, the following properties are listed for a rights holder:

- For Sound Recording
  - Rights holder
  - Rights Holder Proprietary Id
  - Rights Owner
  - Rights Owner Proprietary Id
  - Territories
  - Start Date
  - End Date
  - Split
  - Use Types
- For Music Works
  - Rights Holder
    - Name
    - IPI Name Number
    - Role
  - Rights Holder Original Publisher
    - Name
    - IPI Name Number
  - Rights Holder Proprietary Id
    - Territories
    - Start Date
    - End Date
    - Mechanical
      - Affiliation Society
      - Ownership Split
      - Collection Split
    - Performance
      - Affiliation Society
      - Ownership Split
      - Collection Split
    - Synchronisation
      - Affiliation Society
      - Ownership Split
      - Collection Split

From the requirements for the WebTV pilot, the following properties are listed for a rights holder:

- Name
- Contact Information
- Role
- Rights Type
- Territory
- Start Date
- End Date

This suggests a common rights holder data model across all pilots utilising the same set of property definitions.

The set of identifiers for rights holders mirrors the standard identifiers for users. This allows Bloomen systems to interoperate with third parties, as well as reconcile rights holders with external sources of data, if required. Identifiers to be used include:

- IPI (Interested party information) is a unique identifying number assigned by the CISAC which is applicable to the Music pilot. More information is detailed below in section 5 - References.
- The DDEX Party Identifier standard describes an identification system used to identify each sender and receiver of a DDEX message and how the identification allocation process is undertaken.
- Photographer ID

Existing standards (referenced in section 5):

- The DDEX set of standards specifies a “Party” schema, which is used to represent any business with an interest in digital media content.

3.7 Rights Claim

The Music pilot requirements reference a data model for a rights claim. This is an entity that represents a claim over a musical asset.

The properties of this asset include:

- Rights Holder
- Musical Asset
- Status (Claimed/Conflict)

As this is specific to the music pilot, this is not considered for interoperability.

3.8 Transaction

A data model is required for all transactions within the system. The requirements for the Photo and WebTV pilots are specific about the properties required within this transaction:

For transactions in the Photo pilot the following properties are required:

- From
- To
- Date
- Amount

For the WebTV pilot, transactions are entities which represent store of value, means of reimbursements as well as cryptographic delivery of the content. Properties include:

- Virtual Currency
- Transaction Info (Source, Destination, Amount, Timestamp, Transaction Hash)
- Video Server Delivery Access Control

Common properties for all transactions are the source (from) and destination (to), the date (timestamp), and the amount. A generated unique hash is used to identify transactions.

Existing standards:

- The Ethereum ERC20 Token Standard provides a technical standard for the implementation of smart contracts. More information is included below in the Technologies references section.

3.9 Pilots

3.9.1 Music

The following data models are used in the Music Pilot:

- Regarding the interaction with the Bloomen API, the data is fetched in JSON format and then, the NgRx Angular Framework3 is used in order to instill the music asset data into the application as Typescript Objects4 and deliver it in the user interface (UI).

[3] https://ngrx.io/docs

[4] https://www.typescriptlang.org/docs/handbook/basic-types.html#object

- Regarding the interaction with the Alastria Network5, the pilot uses JSON-RPC calls with JSON format messages (see section 4.2 for an explanation).
- As far as the bulk claim ingest is concerned, the pilot receives a CSV file from the end-user, and then utilizes the Papa Parse library6 for efficiently parsing the CSV to internal data formats that are necessary for the back-end functionalities.

An example of the JSON data describing a recording:


```
{
  "ISRC":"BER181131702",
  "mainArtist":"EX TEMPORE",
  "featuredArtists":[
        "FLORIAN HEYERICK"
  ],
  "title":"DURCH DIE HERZLICHE BARMHERZIGK",
  "versionTitle":null,
  "duration":227,
  "yearOfRecording":2002,
  "territoryOfRecording":"BE",
  "languageOfPerformance":null,
  "originalReleaseDate":"2002-01-01",
  "originalReleaseLabel":"OUTHERE SA",
  "creators":[
        "GOLDBERG JOHANN GOTTLIEB"
  ],
  "isVideo":false,
  "releases":[
        {
        "title":"GOLDBERG JL BACH & KREBS",
        "artist":null,
        "ICPN":"5400439003170",
        "numberOfTracks":null,
        "label":"RICERCAR",
        "duration":null,
        "isCompilation":false
        }
  ]
}
```


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

[6] https://www.papaparse.com/

An example of a JSON document describing a musical work:


```
{
  "ISWC":"T9204649558",
  "originalTitle":"SHAPE OF YOU",
  "creators":[
        {
        "name":"SHEERAN ED",
        "IPINameNumber":"00583552527",
        "role":"ComposerLyricist"
        },
        {
        "name":"MCDAID JOHN",
        "IPINameNumber":"00412720203",
        "role":"ComposerLyricist"
        },
        {
        "name":"MAC STEVE",
        "IPINameNumber":"00257395141",
        "role":"ComposerLyricist"
        },
        {
        "name":"COTTLE TAMEKA D",
        "IPINameNumber":"00338239158",
        "role":"ComposerLyricist"
        },
        {
        "name":"BURRUSS KANDI L",
        "IPINameNumber":"00338170958",
        "role":"ComposerLyricist"
        },
        {
        "name":"BRIGGS KEVIN",
        "IPINameNumber":"00344353278",
        "role":"ComposerLyricist"
        }
  ],
  "alternativeTitles":[
        "SHAPE OF YOU (STORMZY REMIX)",
        "SHAPE OF YOU [OFFICIAL LYRIC VIDEO]",
        "SHAPE OF YOU (LIVE FROM THE 59TH GRAMMYS)",
        "SHAPE OF YOU - ACOUSTIC"
  ],
  "associatedPerformers":[
        "WALK OFF THE EARTH",
        "WALE FEATURING ED SHEERAN",
```

```
        "SING2PIANO",
        "SHEERAN ED",
        "MADILYN BAILEY",
        "GALANTIS",
        "FAME ON FIRE",
        "ED SHEERAN",
        "BOYCE AVENUE"
  ],
  "associatedISRCs":[
        "GBAHS1600463",
        "GBAHS1700245",
        "GBAHS1700200",
        "GBAHS1700196",
        "GBAHS1700651"
  ]
}
```


Bloomen Music Pilot implements a CSV Format, which includes the following columns:

- ISC,
- countries,
- startDate,
- endDate,
- types,
- splitPart,
- rightHolderRole,
- rightHolderProprietaryID,
- title

3.9.2 Photo

The Photo Pilot uses the User, Assets and Transactions schemas as these are described above (3.1, 3.2 and 3.8 respectively)

Other schemas used by the pilot include:

Assignment:

- organisation - Organisation name where the user is invited in
- email - User email where invitation is sent
- accepted - User accepted the invitation, default: false
- user - User
- invitedBy - User

- createdAtUTC - When assignment notification was sent

Media:

- url - The media file URL
- description - Description of the media resource
- keywords
- hash - Unique hash of the media resource
- type - Description of the resource, default is "photo"
- price - Price of the media resource, default is 0
- attribution - Uploader wants credit for media resource usage
- owner - Hash of media resource owner
- rights
- rightsTime - Usage rights time (how long?)
- rightsRegion - Usage rights region (where in the world?)
- analytics
- geo
- metadata
- createdAtUTC - When media resource was uploaded (type: Date)

Organisation:

- name - Organisation name
- hash - Organisation hash
- walletAddress - Blockchain wallet address of organisation's funds
- group - Organisation belongs to this group

The pilot also uses a large list of DTOs (data transfer object) for interacting with the platform which can be viewed in detail in the bottom part of the swagger docs.

3.9.3 WebTV

The WebTV pilot did not interoperate via the API but connected directly to the blockchain via the RPC API. The common data types and formats used by the API are not applicable in this case. Documentation of the data schemas used in the WebTV pilot is therefore outside of the scope of this document. Refer to the pilot specific documentation in this case.

4 APIs and Protocols

In this section we review the APIs and Protocols in use by the Bloomen system, both internally, and for communication with third parties and systems outside of the Bloomen architecture. Use of open and documented APIs allows for further interoperability.

Within the Bloomen architecture the functionality is exposed by two mechanisms, the Bloomen API is a REST based system, and the Quorum RPC exposes direct blockchain transaction functionality.

4.1 Bloomen API

The Bloomen API is a standards-based REST API that exposes functionality within the Bloomen system. It is documented using Open API/Swagger (see the references section for details on this technology).

The Bloomen API is used by the Photo and Music pilots but not the WebTV pilot.

Features from multiple components within the Bloomen system can be accessed via the API, including:

- Functionality related to the uploading of assets
- Updating asset metadata, and removal of assets
- Browsing and searching assets including full-text search
- Access management, User management and KYC
- Copyright management and licensing-contract management

See https://bloomen.herokuapp.com/api-docs for the Bloomen API definition.

No limits to scalability were detected whilst using the Bloomen API. The API is hosted in the cloud so we would not expect there to be any limits to scalability.

4.2 Quorum RPC

The Bloomen architecture uses JSON-RPC[7], provided by Quorum nodes, for direct connection with blockchain transactions. JSON-RPC is a stateless, lightweight remote procedure call (RPC) protocol. The specification defines several data structures and rules for how they are processed. It can be exposed over sockets, or HTTP, or any other message passing protocols. It uses JSON (see references) as a data format.

[7] https://github.com/ethereum/wiki/wiki/JSON-RPC

JSON is a lightweight data-interchange format. It can represent numbers, strings, ordered sequences of values, and collections of name/value pairs.

In addition to the JSON-RPC provided by Ethereum, Quorum exposes below two extra API calls for dealing with accounts and Quorum payloads.

Interoperability within the Bloomen architecture is possible between nodes, and other components that require communication with nodes via the RPC mechanism. For components that use JavaScript, this is simplified by use of the web3.js library, which gives a convenient interface for the RPC methods.

Within the Bloomen components, direct RPC access is used for features including:

- Cryptocurrency and micropayments
- Bloomen wallet management, including management of prepaid cards

All pilots used the Bloomen blockchain RPC.

5 Reference

This section provides a brief description of the technologies mentioned in the document.

5.1 Technologies

5.1.1 OpenZeppelin

OpenZeppelin[8] is an open-source library for secure smart contract development that is hosted on GitHub[9]. It provides implementations of standards like ERC20 and ERC721. Conforming to the appropriate standards (in particular ERC20), when developing Smart Contracts for Bloomen, is important for interoperability. The OpenZeppelin framework provides a starting point for Smart Contract development, which can be extended to meet our requirements.

Features of OpenZeppelin that are relevant to interoperability and scalability within Bloomen include:

- Open-source components
- Implementation of ERC20 standard
- Stable API allows easier upgrade between versions

[8] https://openzeppelin.org/

[9] https://github.com/OpenZeppelin/openzeppelin-solidity

- Code is maintained by Zeppelin10 the company, but community-audited to high standard, and with regular audit-checks

5.1.2 ERC20

ERC20 is a standard for smart contracts that operate on the Ethereum (and Ethereum clones like Quorum) blockchain. It specifies a mechanism for implementing tokens. By adhering to this standard the Bloomen project ensures that any tokens generated are interoperable with the wider blockchain ecosystem.

ERC-20 defines a common list of rules for Ethereum tokens. The OpenZeppelin framework in use by the Bloomen project provides a standards-based implementation of the specification.

5.1.3 OpenAPI (Swagger)

- https://swagger.io/specification
- Standard, language-agnostic interface to RESTful APIs
- both humans and computers to discover and understand the capabilities of the service without access to source code,
- understand and interact with the remote service with a minimal amount of implementation logic.
- can then be used by documentation generation tools to display the API,
- code generation tools to generate servers and clients in various programming languages,
- testing tools

A Swagger definition was made available throughout the development process and formed a key resource. It simplified cooperation across teams due to the standardised and well-understood methodology for documenting the API resources as they were being developed. The swagger definition was served from the API, so was always available and up-to-date with the current status of the data schemas and functions of the backend, and used by the Kendraio App, management workflows, and music and photo pilots.

5.1.4 JSON

JSON (RFC 4627[11]) is a lightweight, text-based data interchange format based on the JavaScript Programming Language Standard. JSON defines a small set of formatting

[10] https://zeppelin.solutions/

[11] https://www.ietf.org/rfc/rfc4627.txt

rules for structured data. It is widely supported across all platforms and languages, and is therefore an ideal candidate for encoding of data across the whole Bloomen architecture.

JSON format is used by the API and blockchain RPC as a consistent serialisation format for data being shared between components of the bloomen system.

5.1.5 JSON-LD

RDF is a way to express assertions in a schematic way. JSON-LD is a data structure to serialize RDF in JSON. It can be used to provide an interoperable way to transfer meaning in data with less ambiguity. JSON-LD is self documenting, and includes a normalising specification. This specification of how to normalise data represented as JSON-LD is essential for generating unique hash IDs of encoded metadata.

JSON-LD is not utilised by the Bloomen system in the final implementation. The data models and formats did not require this level of self-documentation or normalisation.

5.2 Existing Identifiers

As a reference, Industry standard identifiers which have been used within the Bloomen project are described in the following sections. More information is available at the provided links.

5.2.1 ISRC

ISRC (International Standard Recording Code)

- https://isrc.ifpi.org/en
- ISRC enables recordings to be uniquely and permanently identified. ISRC helps to avoid ambiguity and simplifies the management of rights when recordings are used across different formats, distribution channels or products. The ISRC for a recording remains a fixed point of reference when the recording is used across different services, across borders, or under different licensing deals.[12]
- Relevant pilot(s): Music
- This is currently a part of the Music pilot requirements.

[12] https://isrc.ifpi.org/en

5.2.2 ISWC

ISWC (International Standard Musical Work Code)

- http://www.iswc.org/en
- ISWC (International Standard Musical Work Code) is a unique, permanent and internationally recognized ISO reference number for the identification of musical works.[13]
- Relevant pilot(s): Music
- This is currently a part of the music pilot requirements.

5.2.3 IPI

IPI (Interested party information)

- https://www.ipisystem.org
- IPI is a unique identifying number assigned by the CISAC database to each Interested Party in collective rights management. It is used worldwide by more than 120 countries and three million right holders.[14]
- Relevant pilot(s): Music
- This could be useful for interoperability between Collectives (CMOs)

5.2.4 EIDR and DOI

EIDR (Entertainment Identifier Registry)

- https://eidr.org
- Is a global unique identifier system for a broad array of audio visual objects, including motion pictures, television, and radio programs. The identification system resolves an identifier to a metadata record that is associated with top-level titles, edits, DVDs, encodings, clips, and mash-ups. EIDR also provides identifiers for Video Service providers, such as broadcast and cable networks.[15] EIDR is an implementation of a Digital Object Identifier (DOI).

DOI (Digital object identifier)

- https://www.doi.org

[13] http://www.iswc.org/
[14] https://en.wikipedia.org/wiki/Interested_Parties_Information
[15] https://en.wikipedia.org/wiki/Digital_object_identifier

- Is a persistent identifier or handle used to identify objects uniquely, standardized by the International Organization for Standardization (ISO).[16]
- Relevant pilot(s): WebTV
- EIDR and DOI may be useful for the identification of assets relevant to the WebTV pilot. Its use will likely depend on the adoption by relevant parties.

5.2.5 ISMN

- ISMN (International Standard Music Number)
- https://www.ismn-international.org
- The International Standard Music Number or ISMN is a thirteen-character alphanumeric identifier for printed music developed by ISO.[17]
- Relevant pilot(s): Music
- ISMN are mostly used outside the US and have a number of implementation issues including its relation to the ISBN book standard. Its relevance to the Bloomen will depend on its adoption by the systems we will integrate with. Those being mainly Collectives (CMOs)

5.2.6 ISNI

ISNI (International Standard Name Identifier)

- http://www.isni.org/
- ISNI is the ISO certified global standard number for identifying the millions of contributors to creative works and those active in their distribution, including researchers, inventors, writers, artists, visual creators, performers, producers, publishers, aggregators, and more.
- Relevant pilot(s): Photo, Music, WebTV
- The ISNI is relevant for all 3 pilots and therefore is an excellent candidate for any parent superclass / common part of the data model.

5.2.7 ISLI

ISLI (International Standard Link Identifier)

- https://www.isbn-international.org/content/other-identifiers
- The International Standard Link Identifier identifies the links between different entities. A "source" entity can be linked to a "target" entity via the ISLI. ISLI can be applied to many different types of entity - from material objects to digital

[16] https://en.wikipedia.org/wiki/Digital_object_identifier
[17] https://en.wikipedia.org/wiki/International_Standard_Music_Number

resources to parties (e.g. natural or legal persons) or even abstract items (e.g. time, places). The ISLI does not change the content, access rights or ownership and can be used in conjunction with existing identifiers.[18]
- Relevant pilot(s): WebTV, Photo, Music
- The ISLI is relevant for all 3 pilots and therefore is an excellent candidate for any parent superclass / common part of the data model.

5.2.8 ISAN

ISAN (International Standard Audiovisual Number)

- http://www.isan.org/about/#what_is_isan
- ISAN provides a unique, internationally recognized and permanent reference number for each audiovisual work registered in the ISAN system.[19]
- Relevant pilot(s): Web TV
- This standard would allow Bloomen to identify DVDs, Video Recordings, Digital Footage, TV programmes etc hosted online or in hard copy format.

5.3 Existing Schemas

Some existing schemas were identified in the first iteration of this deliverable, which have not been core to the implementation of the Bloomen system for operating the pilots. Adapters are available within the Kendraio App and can be configured to provide interoperability with these standards if required.

5.3.1 Schema.org

Schema.org is a well-known standard for defining objects:

- Party
  - https://schema.org/Person
  - schema.org/Organization
- Creation
  - https://schema.org/CreativeWork
  - And its subtypes: Book, Movie, MusicComposition
- Place
  - https://schema.org/Place
- Action
  - https://schema.org/Action

[18] https://www.isbn-international.org/content/other-identifiers
[19] http://www.isan.org/about/#what_is_isan

5.3.2 DDEX

- DDEX: to communicate sound recordings http://ddex.net
- Relevant pilot(s): Music

5.3.3 CWR

- CWR: to communicate musical works http://members.cisac.org/CisacPortal/consulterDocument.do?id=22272
- Relevant pilot(s): Music

6 Scalability

This section summarizes the requirements for the Bloomen system components that relate to scalability, and discusses mechanisms to achieve the required service levels. The scalability requirement was further investigated during the second iteration of this task. The input gathered during the pilots was assessed and fed into further development of this task.

6.1 Bloomen blockchain scalability analysis

Scalability issues are a feature of blockchain solutions in general. Quorum, as used by Bloomen, offers some improvements. Performance of the blockchain network will be the main determinant of the Bloomen system. The factors that affect scalability include the number of transactions per second, and the amount of traffic on the network. The transaction speed can be configured according to smart contracts. To give better performance, Quorum uses a vote-based RAFT consensus algorithm. It also uses Istanbul BFT consensus algorithm.

Given the fact that Quorum by nature is a blockchain distributed database, makes it a highly scalable solution in any given application[20]. Quorum uses Ethereum blockchain, which is ideal for Business to Customer applications that require a highly scalable solution given the number of customers and nodes created for the purpose of the solution. It uses PoW (Proof of Work) algorithms to reach a consensus which may be resilient, but as far as the performance and scalability are concerned there are better solutions[21].

[20] Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system, May 2009

[21] Lewenberg, Y., Sompolinsky, Y., Zohar, A.: Inclusive block chain protocols. In: Boehme, R., Okamoto, T. (eds.) FC 2015. LNCS, vol. 8975, pp. 528–547. Springer, Heidelberg (2015)

In order to push even further the scalability and performance of the Ethereum blockchain, Quorum has adapted faster and more efficient algorithms for consensus. More specifically, it has adapted the BFT (Byzantine Fault Tolerance). As far as the scalability is concerned, there are two types of scalability that affect the blockchain solution, the scalability of Nodes in the system and the Scalability of the Users (Customers). As far as the Nodes are concerned the PoW performs better and can scale in more nodes. Given the fact that the Bloomen solution is not a public blockchain, the node scalability is not that important. On the other hand, user scalability is far more efficient in BFT[22] making it highly scalable given the nature of the Bloomen solution.

Furthermore, while PoW protects a public blockchain by deliberately introducing cryptographic difficulty, it is unnecessary and wasteful (excessive power consumption) in a permissioned setting where participants are known. Instead, BFT algorithms lead to faster consensus and provide immediate transaction finality, making them a suitable choice for permissioned blockchain implementations, such as Quorum[23].

Immediate transaction finality indicates that once the transaction is included in the block, it is confirmed and will not be rolled back, offering thus a high transaction rate. On the contrary, PoW approaches are probabilistic and have to spend a significant amount of time solving the cryptographic puzzle. Consequently, these models have high transaction latencies and therefore a low transaction rate[24].

The REST-based Bloomen API is a centralised server that offers up some features of the Bloomen system to certain client applications. It is based on Node.js, which is a high performance Javascript-based server. Node.js implementations can scale to handle 1000s of connections from all clients, including mobile clients, web portals, and Apps.

No limitations detected at the blockchain side, the technology used is mature and widely used across the globe so it is not easy to find limitations with a simple pilot.

[22] https://link.springer.com/chapter/10.1007/978-3-319-39028-4_9

[23] https://arxiv.org/pdf/1809.03421.pdf

[24] https://www.persistent.com/wp-content/uploads/2018/02/wp-understanding-blockchain-consensus-models.pdf

6.2 Blockchain scalability feedback from Bloomen Pilots

Scalability comments from the separate pilots:

6.2.1 Music

The Music Pilot (or Music App) combines scalability features from the same core of the application to the divergent data sources it interacts with.

The information imports from external sources affect the App's performance, either they originate from the Bloomen API, the Alastria Network or the end-user themselves. In particular, when fetching data from the Bloomen API, the import agility of the pilot matches that one of the Bloomen API, which is a cloud framework with no scalability limits (see 4.1). Music asset data is fetched inside the App with an insignificant milliseconds delay, an expected Internet traffic number.

Moreover, regarding the interoperability with the Alastria Network, the App is benefiting from the Quorum-based blockchain that scales to hundreds transactions per second (tps). In this context, when different end-users are exploiting the pilot's features by submitting blockchain transactions simultaneously, the blockchain network can support this order of tps. From the Music Pilot's point of view, a single user can submit one blockchain transaction at a time, due to the nature of the blockchain, i.e. a single address can submit one transaction at a time.

Furthermore, the App has no scalability issues concerning the user interaction. It exploits the rapidity of NgRx Angular Framework which serves perfectly Angular reactive applications. From simple user navigation to blockchain transaction submission, NgRx offers agility and quickness of interactions.

Finally, the Music Pilot stores efficiently the claim submitted information on the blockchain. This claim data is considered an allowed amount of data to store on a Solidity structure of a blockchain smart contract. At the time of writing, the overall pilot scales for hundreds of claims with no issues, while further effort is in progress in order to support the industry numbers.

6.2.2 Photo

Since all of the functionality of the Photo pilot is based on the Bloomen platform the scalability of the application is the same as the platform. The Photo pilot uses three distinct functionalities of the platform that have to be addressed, namely interaction with the REST API for generic functions of the app, media transfer to and from digital storage (Amazon s3), and interaction with the blockchain for wallet functionalities or right management through the smart contract.

The platform is hosted on heroku which is a cloud provider and there are no blocking methods in any scenario regarding the photo pilot, so the interaction with the API is fully scalable. The same goes for the media storage, since s3 runs on the amazon cloud platform as well. Finally the smart contract is deployed on Alastria which has proven to be a reliably scalable blockchain provider through all our tests so far.

6.2.3 WebTV

The WebTV pilot is a decentralized application that is hosted on the blockchain and that uses the bloomen wallet to give end users access to their information stored in smart contracts.

The development of the pilot has not required an initial data load since special care has been taken to minimize the information stored on the blockchain to create a sustainable system.

Load tests have been carried out with up to 200 concurrent users without detecting concurrency problems and obtaining response times in the validation of transactions of about 4-5 seconds (includes process, creation of block and confirmation of the different validator nodes of the blockchain).

Within the project, specific developments have been made to guarantee the scalability of the system by adding a cluster of 3 nodes to the blockchain that serve the project. This cluster can be modified to add or remove nodes according to the planned need.

At the same time, for the 2nd iteration of the WebTV pilot to run smoothly, the front-end that was provided to enable playback of content should also have addressed certain scalability issues. Since this is a website that can host more than 3,000 concurrent viewers, when it comes to video playback, no issues had to be addressed. The only area that required some work was the system that validated, via a Bloomen API, whether the user had access to watch audiovisual content.

To be on the safe side, the website required the user to click a button, confirming that they have made the payment via the mobile wallet, before the website would actually call the Bloomen API. This decision was made because the website has thousands of concurrent users, who if navigated on the Bloomen WebTV interface at the same time, could cause disruption on the server side of the WebTV platform provider.

7 Conclusion

This document is meant to be read as an update to the state of interoperability and scalability within the Bloomen project. The document also provides references for key technologies and standards that have been evaluated and considered as part of the design and development of the system. It documents the recommendations as adopted by the project and, of course, each facet of interoperability and scalability has its own merits and prioritisation dependent on requirements and resources.

As an update to the previous iteration of the deliverable, key developments have been documented as they relate to interoperability and scalability. This includes results from the implementation, testing and verification of the core system, the development of pilot systems, and the operation and evaluation of the three pilots.