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

D4.5 Initial Multiplatform interoperability and scalability framework

Version:	**3**
Date:	**25/05/2019**
Authors:	**Kendraio**
Type	**Report**
Dissemination level	**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).

Bloomen is being 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 will be designed and developed given the various requirements that are elicited in WP2. Deriving schemas from the requirements. Using existing standards and standardising across the pilots. In particular, the appropriate blockchain data structures will be implemented for allowing common information reference framework for integrating the different ICT modules. This information framework will be based on standard representation formats through JSON, XML, RDF, languages and frameworks, ensuring thus a standardized and general purpose form of exchanging data.

Open and easily manageable APIs will be implemented for bridging functional components of different integrated systems. These APIs will facilitate interoperability and communication between components within the whole Bloomen architecture. The APIs will be standards-based HTTP RESTful APIs as well as 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 should be standardised across the pilots. The APIs themselves will be documented in an open format (Open API/Swagger).

This will enable a standard way for a high-level communication and interaction protocol between diverse systems ensuring thus the necessary interoperability for the Bloomen system. The output of this activity will be a standard based interoperability layer ready to be used for validation. This includes data formats, schemas, API descriptions and tools for validation.

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

This information is intended to guide decision making by technical partners in order to increase interoperability. It is not intended that all the elements here described are implemented as they are listed here as an industry technical overview.

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 so far as part of this deliverable. Although much has been achieved, there is still a lot of work to do on implementing this across the system and the pilots, in particular in the area of scalability. This is the first of two reports. The follow up report at the end of the project will document the continuation of this work throughout the rest of the project.

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 to be 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

High-level operations, based on the requirements for the Bloomen system as a whole, as they relate to interoperability, are presented in the following sections.

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 be a great benefit for users to have a single identity across multiple pilots, however, the bigger challenge is to maintain a single identity across the various systems that each pilot needs to connect to. For example, if the Music pilot needs to connect to multiple Collectives (CMOs) it will be easier for a user to have a single identity across those services. One way we can achieve this is by using a unique identifier like the ISAN (International Standard Name Identifier). This will allow for easier mapping of external entities to the Bloomen internal schema. The use of unique identifiers should be used not only for individuals but also for organisations and other entities.

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

Music: Users must be authorised by their respective CMO Member administrators.

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 will need to handle 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:

- Check the asset is not already registered.
- Register the asset
- Create a blockchain hash and receives 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 are to communicate with a number of services and each of these services may use a different schema and a different workflow. The only way this can work and become scalable is 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 needs to be to the internal Bloomen format and should use 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 will normally make use of industry standards for identifiers like ISRC and ISWC for music, and those should be 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 will not be supported natively by Collectives (CMOs) and even when they do, they are unlikely to be consistent.

Some of those requirements are:

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

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

A notification system that is common to all the pilots will work along with this metadata layer to ensure consistent delivery of notifications and management of preferences and relationship between users and organisations.

The rest of this metadata may be use case specific, but commonalities between the use cases should always be explored.

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 should extend the internal Bloomen schema with metadata that will support 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 can be hardcoded, however, using an interoperability layer will make it easier to replace the backend system or support multiple ones.

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 recommend 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 will allow us to develop robust low-level functionality separate from the real-life business processes that are likely to be harder to achieve as a result of complex real-life systems, with limited automation support. As such, much of the achieved use cases will be a proof of concept (this may not be relevant to use cases fully developed within the Bloomen project). Having the business logic driven by metadata will allow for faster adoption of changes and new opportunities, allowing for the development of real-life use cases in a scalable way.

One other area that is worth exploring is that all rights data across all pilots are derived from the same legal framework of Authors' Rights and Neighbouring rights and as such it may be possible to abstract those.

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. Those are all new concepts that do not currently exist and are unlikely to 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 Claim

- A Claim is also referred to as "change request" in some documentation
- Music
    - See Music pilot Schema (below).
    - The workflow is described in 3.1.5.5 and 3.1.5.6 of D2.2.
    - There is no industry standard workflow for claiming and conflict resolution.
- Pilot: Music, Photo

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 will determine the level of 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 will be assigned to each content item and will be pushed to the blockchain.

There may be existing data standards and schemas which could be used, and these are noted in this section where applicable, with more details on relevant standards given in the later References section.

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. Potential 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.

Existing standards (referenced in section 5):

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

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.

Possible 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
- Photo ID - there is no clear industry ID
- Video - there is no clear industry ID

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 at this stage.

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 at this stage.

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
  - Syncronisation
    - 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 possible 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. Potential 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 (Accepted/Rejected/Pending)

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

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.

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).

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

4.2 Quorum RPC

The Bloomen architecture uses JSON-RPC[1], 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

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

HTTP, or any other message passing protocols. It uses JSON (see references) as a data format. 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

5 Reference

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

5.1 Technologies

5.1.1 OpenZeppelin

OpenZeppelin[2] is an open-source library for secure smart contract development that is hosted on GitHub[3]. 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
- Code is maintained by Zeppelin[4] the company, but community-audited to high standard, and with regular audit-checks

[2] https://openzeppelin.org/
[3] https://github.com/OpenZeppelin/openzeppelin-solidity
[4] https://zeppelin.solutions/

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

5.1.4 JSON

JSON (RFC 4627[5]) is a lightweight, text-based data interchange format based on the JavaScript Programming Language Standard. JSON defines a small set of formatting 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.

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.

- Ref: COALA IP
  - https://www.w3.org/2016/04/blockchain-workshop/slides/McConaghy-COALA_IP-short.pdf
  - COALA IP Protocol
    - A community-driven minimum-viable set of data for IP licensing (RDF schema definitions, JSON-LD)

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

        - A free and open messaging protocol for license transactions (LCC, Interledger, IPLD)
- Ref: Schema.org
    - Implementations are in all main languages:
    - https://github.com/digitalbazaar/jsonld.js
    - Operations:
        - Compact - Expand - Flatten - Frame (Query by example / Force structure)
        - Normalize (Useful for hashing same data with multiple representations)

5.2 Existing Identifiers

Industry standard identifiers which may be useful within the Bloomen project are described in the following sections.

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.[6]
- Relevant pilot(s): Music
- This is currently a part of the Music pilot requirements.

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.[7]
- Relevant pilot(s): Music
- This is currently a part of the music pilot requirements.

[6] https://isrc.ifpi.org/en
[7] http://www.iswc.org/

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.[8]
- 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.[9]
  EIDR is an implementation of a Digital Object Identifier (DOI).

DOI (Digital object identifier)

- https://www.doi.org
- Is a persistent identifier or handle used to identify objects uniquely, standardized by the International Organization for Standardization (ISO).[10]

- 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.[11]
- 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

[8] https://en.wikipedia.org/wiki/Interested_Parties_Information
[9] https://en.wikipedia.org/wiki/Digital_object_identifier
[10] https://en.wikipedia.org/wiki/Digital_object_identifier
[11] https://en.wikipedia.org/wiki/International_Standard_Music_Number

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 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.[12]
- 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.[13]
- 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.

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

5.3 Metadata and other standards

The following are a number of standards that should be explored.

5.3.1 XMP and IPTC

XMP (Extensible Metadata Platform)

- XMP standardizes a data model, a serialization format and core properties for the definition and processing of extensible metadata. It also provides guidelines for embedding XMP information into popular image, video and document file formats, such as JPEG and PDF, without breaking their readability by applications that do not support XMP.[14]

IPTC (Information Interchange Model)

- XMP has largely superseded IIM's file structure, but the IIM image attributes are defined in the IPTC Core schema for XMP.[15]

- Pilot: Web TV
- THis may be useful in both determining the WebTV schema but also may be useful for interoperability with other systems.

5.3.2 MXF

- MXF (Material Exchange Format): A container format for professional digital video and audio media defined by a set of SMPTE standards[16]
- Pilot: WebTV
- MXF has full timecode and metadata support, and is intended as a platform-agnostic stable standard for future professional video and audio applications and may be useful if assets need to hold their own metadata or the Bloomen system need to handle assets that already hold MXF data.

5.4 Existing Schemas

5.4.1 Schema.org

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

- Party
  - https://schema.org/Person
  - schema.org/Organization
- Creation

[14] https://en.wikipedia.org/wiki/Extensible_Metadata_Platform
[15] https://en.wikipedia.org/wiki/IPTC_Information_Interchange_Model
[16] https://en.wikipedia.org/wiki/Material_Exchange_Format

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

5.4.2 DDEX

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

5.4.3 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 will be further investigated during the second iteration of the specific document. The input gathered during the pilots will be assessed and feed the further development of the specific 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[17]. Quorum uses Ethereum blockchain, which is ideal for Business to Customer applications that require 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[18]. 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[19] 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[20]. 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

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

[18] 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)

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

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

contrary, PoW approaches are probabilistic and have to spend significant amount of time solving the cryptographic puzzle. Consequently, these models have high transaction latencies and therefore a low transaction rate[21].

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 performant Javascript-based server. Node.js implementations can scale to handle 1000s of connections from all clients, including mobile clients, web portals, and Apps.

7 Conclusion

This document is meant to be read as a description of the current state of interoperability and scalability within the Bloomen project. It is also to be read as an advisory, putting forth recommendations and possible additions to foster better interoperability and scalability in the future.

It remains to be seen which recommendations are adopted by the project and, of course, each facet of interoperability and scalability will have its own merits and prioritisation dependent on requirements and resources.

The need for interoperability is crucial if Bloomen is to be relevant to an external audience. Interoperability will also reduce costs of project maintenance by leveraging services and libraries already implemented by third parties. There is no need to reinvent the wheel. So, practising interoperability enables us to concentrate and maximise our efforts on innovation and usability.

The need for scalability is clear if Bloomen is to have any relevance in the coming new world of media distribution. The sheer number of creative people making professional works is incredible. And now that the bug has bitten the numbers are only set to rise. People like consuming media and they also like making it. Bloomen needs to able to scale to meet that challenge.

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