

**École polytechnique de Louvain**

# **Intégration d'une visionneuse implémentant le protocole IIIF dans un logiciel de description archivistique**

Auteur: **Louis CARLIER**  
Promoteurs: **Yves DEVILLE, Aurore FRANÇOIS**  
Lecteurs: **Xavier GILLARD, Caroline DERAUW**  
Année académique 2023–2024  
Master [60] en sciences informatiques

## Résumé

L'objectif de ce mémoire est centré sur l'intégration d'une visionneuse compatible avec les normes IIIF au sein du logiciel AtoM, un outil de description archivistique notamment utilisé par le service des Archives de l'UCLouvain. Dans un premier temps, ce travail aborde les concepts et enjeux permettant d'appréhender au mieux la phase de développement. Il décrit les spécificités d'AtoM et sa gestion actuelle des objets numériques et explique également en détail les normes IIIF ainsi que leur impact sur la visualisation et l'interopérabilité des ressources numériques sur le Web. Il s'ensuit une description de la phase de développement couvrant les étapes clés telles que la mise en place d'une instance dédiée d'AtoM, les divers choix de mise en œuvre ainsi que l'implémentation de la nouvelle visionneuse. Cette phase aboutit sur le résultat concret de l'intégration de cette dernière, permettant ainsi l'utilisation du protocole IIIF directement depuis l'environnement d'AtoM.

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Le logiciel de description archivistique AtoM</b>	<b>6</b>
2.1	Présentation générale . . . . .	6
2.2	Présentation technique . . . . .	8
2.3	Visionneuse actuelle et informations archivistiques . . . . .	10
<b>3</b>	<b>L’International Image Interoperability Framework</b>	<b>15</b>
3.1	Présentation . . . . .	16
3.2	Spécifications . . . . .	17
3.3	Visionneuses IIIF existantes . . . . .	21
<b>4</b>	<b>Limitations actuelles et besoins</b>	<b>24</b>
4.1	Problématique . . . . .	24
4.2	Objectif . . . . .	25
4.3	Acteurs . . . . .	25
4.4	Besoins fonctionnels . . . . .	25
4.5	Besoins non fonctionnels . . . . .	26
<b>5</b>	<b>Implémentation open source de la visionneuse IIIF Mirador dans AtoM</b>	<b>28</b>
5.1	Mise en place d’AtoM . . . . .	28
5.2	Choix de la visionneuse . . . . .	30
5.3	Codebase d’AtoM . . . . .	31
5.4	Intégration de Mirador . . . . .	32
<b>6</b>	<b>Résultats et expérimentations</b>	<b>35</b>
6.1	Résultats . . . . .	35
6.2	Expérimentations sur les performances . . . . .	38
<b>7</b>	<b>Conclusion</b>	<b>41</b>

<b>Bibliographie</b>	<b>42</b>
<b>Annexes</b>	<b>44</b>

# Chapitre 1

## Introduction

De nos jours, une grande partie des institutions culturelles et académiques possédant un service de gestion d'archives mettent également à disposition leur catalogue de manière numérique. Cette numérisation s'inscrit dans le contexte d'évolution de notre société au fil des dernières décennies et comprend de multiples avantages significatifs. Premièrement, cela fluidifie l'échange d'informations et renforce la coopération entre les institutions, services et personnes. De plus, l'accès et la recherche au sein des archives sont grandement simplifiés, ce qui représente un gain de temps et d'effort pour les utilisateurs internes ou externes au service[1]. L'UCLouvain reflète d'ailleurs cette tendance en mettant à disposition son catalogue d'archives via le logiciel AtoM.

Parmi les différentes innovations facilitant et représentant cette transition numérique, l'International Image Interoperability Framework (IIIF) joue un rôle important. Les normes définies par le IIIF offrent une approche standardisée pour la représentation, la diffusion et la manipulation d'images en haute qualité sur le Web. Les outils les plus répandus utilisant ces normes sont ce qu'on appelle des visionneuses IIIF, ces dernières permettent la manipulation avancée de ressources directement depuis une interface sans avoir à télécharger le document dans son espace personnel.

L'objectif de ce mémoire est l'intégration d'une visionneuse compatible avec le protocole IIIF au sein de l'environnement d'AtoM. En effet, la représentation des documents numériques dans AtoM est très rudimentaire et n'intègre pas certaines innovations modernes telles que celles proposées par le framework IIIF. Au cours des différents chapitres, tous les sujets essentiels à la bonne appréhension des concepts et enjeux relatifs à la problématique seront abordés permettant ainsi une compréhension complète de l'intégration.

Dans un premier temps, nous analyserons les spécificités relatives au logiciel AtoM ainsi que sa gestion actuelle des ressources numériques. Nous aborderons également en détail les différentes normes IIIF ainsi que leur impact sur la visualisation et l'interopérabilité des ressources numériques. Après cela, nous décrirons le cahier des charges relatif au projet, de sorte à définir correctement la problématique ainsi que l'objectif et les besoins qui lui sont liés. Ensuite, nous détaillerons le processus de développement réalisé ainsi que les différents choix d'implémentation effectués. Enfin, dans le dernier chapitre, nous constaterons plus concrètement le résultat obtenu suite à la phase d'intégration.

# Chapitre 2

## Le logiciel de description archivistique AtoM

Les logiciels de description archivistique jouent un rôle essentiel dans la gestion actuelle des archives, ils offrent aux institutions des outils afin d'organiser et de rendre accessible leurs catalogues. AtoM, est l'un de ces logiciels et il est utilisé par de nombreuses institutions à travers le monde, y compris les Archives de l'UCLouvain[2] et des institutions internationales de grande envergure telles que l'OTAN[3].

Dans ce chapitre, nous examinerons donc en détail AtoM en explorant ses fonctionnalités et caractéristiques, sa structure technique ainsi que sa gestion actuelle de documents d'archives numériques. En comprenant mieux AtoM et sa structure actuelle, nous serons mieux préparés pour analyser et comprendre les limitations auxquelles il est confronté, et ainsi trouver des solutions adéquates.

Nous baserons ici la majorité de notre analyse sur la version 2.4 d'AtoM, qui est utilisée par les Archives de l'UCLouvain.

### 2.1 Présentation générale

AtoM, pour "Access to Memory"[4] est une application Web open source pour la description et l'accès aux archives basées sur les normes du Conseil International des Archives (CIA), notamment ISAD(G) pour les descriptions archivistiques (nous définirons ces "descriptions" en détail par la suite). Il propose un environnement multilingue et multi-dépôts afin de répondre facilement aux besoins variés des utilisateurs. Il met également à disposition des outils de recherche puissants et efficaces afin d'aider au mieux les utilisateurs à localiser et gérer le contenu.

Le code d'AtoM est diffusé sous la licence GNU Affero General Public Licence (A-GPL 3.0), permettant ainsi à tout un chacun de le consulter, de le modifier ou de l'adapter selon les besoins. De plus, AtoM est continuellement mis à jour et amélioré par une communauté active de développeurs et d'utilisateurs, garantissant sa fiabilité dans un environnement en constante évolution.

Dans AtoM, plusieurs niveaux de privilèges sont définis par défaut[6], offrant ainsi des rôles distincts aux utilisateurs en fonction de leurs responsabilités et de leurs autorisations dans le système :

- **Chercheur** : utilisateur anonyme avec un accès en lecture seule, il peut rechercher et parcourir les descriptions archivistiques publiées.
- **Administrateur** : il bénéficie de privilèges étendus, y compris l'installation, l'importation, la création, la modification et la suppression des enregistrements. Il peut également personnaliser AtoM, gérer les comptes utilisateurs et créer de nouveaux rôles.
- **Éditeur** : il peut créer, modifier, afficher des brouillons et exporter des descriptions, mais n'a pas accès à l'onglet administrateur. Il a également droit à la modification des statuts de publication des objets d'information.
- **Contributeur/Donateur** : il peut créer, modifier, visualiser des brouillons et exporter des descriptions, mais ne peut pas modifier le statut de publication des objets d'information.
- **Traducteur** : il est autorisé à modifier le contenu du système AtoM, mais avec des privilèges limités à la traduction et à la visualisation des documents. Il peut également rechercher, parcourir et traduire les descriptions publiées, ainsi que les éléments de l'interface utilisateur et le contenu de la base de données. Il a accès aux brouillons des descriptions, mais ne peut pas les modifier.

Pour notre analyse, nous réduisons ces niveaux de privilèges à deux catégories d'utilisateurs : les archivistes (les utilisateurs possédant les droits d'administrateur) et les visiteurs. Cette simplification est justifiée par le fait que ces deux types d'utilisateurs couvrent efficacement les cas d'utilisation d'AtoM pour la problématique liée à ce mémoire. De plus, il est important de noter que les privilèges dans AtoM sont personnalisables, permettant ainsi aux institutions de s'adapter à leurs propres exigences et structures.

## 2.2 Présentation technique

Au niveau de son architecture, AtoM est composé des éléments suivants :

### Frontend (client) :

- **Pages HTML et serveur Web** : le serveur Web (Apache dans notre cas) met à disposition les pages HTML à travers le navigateur Web et toutes les interactions de l'utilisateur avec le système se font via le navigateur.

### Backend (serveur) :

- **Code PHP** : le code PHP permet de gérer les demandes et réponses entre le client Web (l'utilisateur), la logique de l'application et la base de données.
- **Base de données MySQL** : AtoM utilise une base de données relationnelle MySQL afin de stocker ses informations.

### Framework et Services :

- **Symfony** : Symfony est un framework PHP open source utilisé ici pour organiser les différentes parties composant AtoM. Il fournit un ensemble de composants permettant de faciliter la conception et la maintenance d'applications Web.
- **Elasticsearch** : Elasticsearch est un serveur de moteur de recherche et d'analyse pour les applications. Il n'est pas intégré directement dans le code d'AtoM mais doit être déployé en parallèle. Leurs interactions se font via une API REST.

Pour des informations détaillées sur les versions spécifiques des technologies utilisées dans ce mémoire, veuillez vous référer à la section dédiée à l'implémentation de la visionneuse IIIF Mirador dans AtoM.

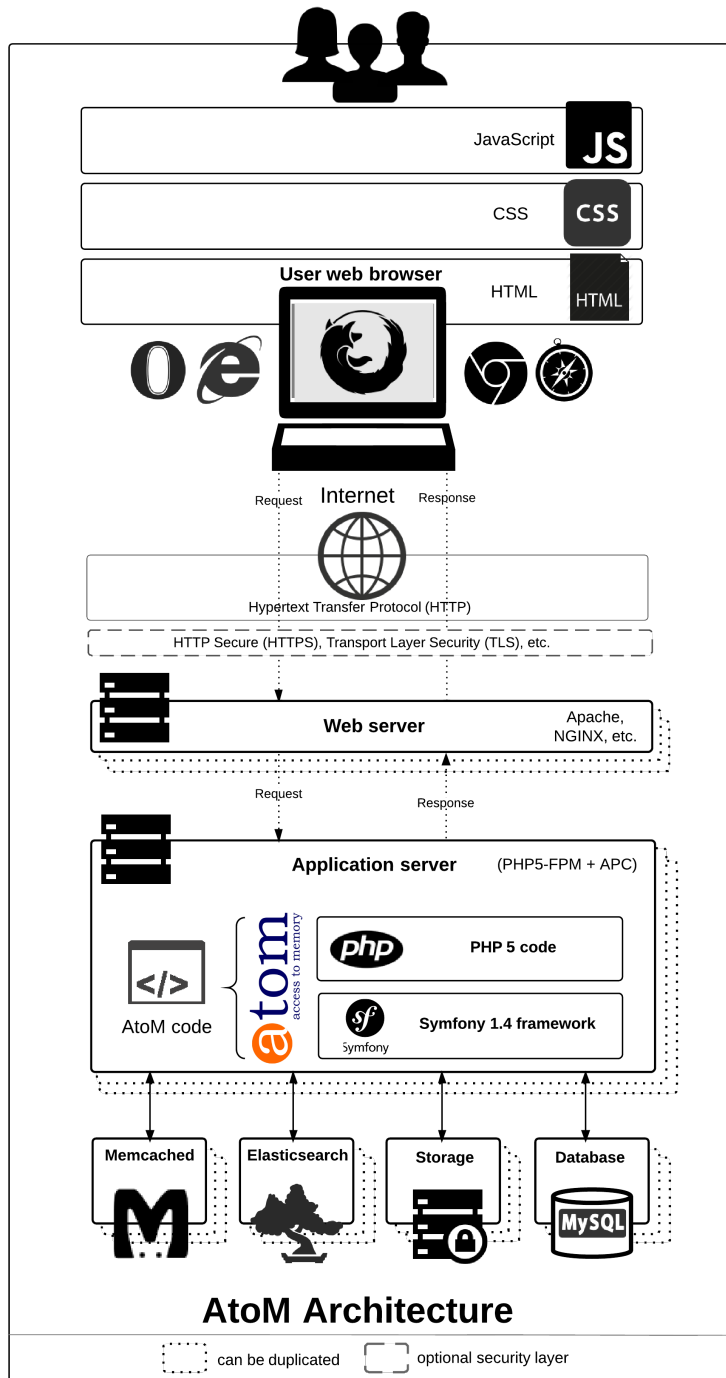


Figure 2.1: Architecture d'AtOM[5]

## 2.3 Visionneuse actuelle et informations archivistiques

Passons maintenant en revue les fonctionnalités mises à disposition par AtoM pour la gestion de ses documents numériques. Nous diviserons cette analyse en deux parties distinctes, celle dédiée aux archivistes disposant de droits d'administration et celle réservée aux visiteurs, afin de mettre en évidence les actions possibles selon les autorisations accordées.

De plus, pour garantir une approche générale, nous nous baserons sur l'instance de démonstration mise à disposition par AtoM[7]. Cela nous fournira une vue générale des fonctionnalités et de la visionneuse actuelle d'AtoM, tout en reconnaissant que les configurations peuvent varier en fonction des institutions et des installations spécifiques.

### Pour les visiteurs :

Lorsque qu'un visiteur se retrouve à explorer un catalogue archivistique, il a accès à une arborescence lui permettant de naviguer aisément à travers la structure de classification définie.



Figure 2.2: Exemple d'arborescence dans AtoM

Lorsqu'il décide de sélectionner un élément, deux cas de figure sont alors possibles : soit cet élément est une entité de plus haut niveau (comme une collection, une sous-collection, une série, etc.), soit il s'agit d'une pièce individuelle. Dans le premier cas, si cet élément contient une ou plusieurs descriptions archivistiques liées à des documents numériques, alors l'utilisateur aura accès à un carrousel d'images lui donnant un aperçu de ces documents comme illustré avec la Figure 2.3.

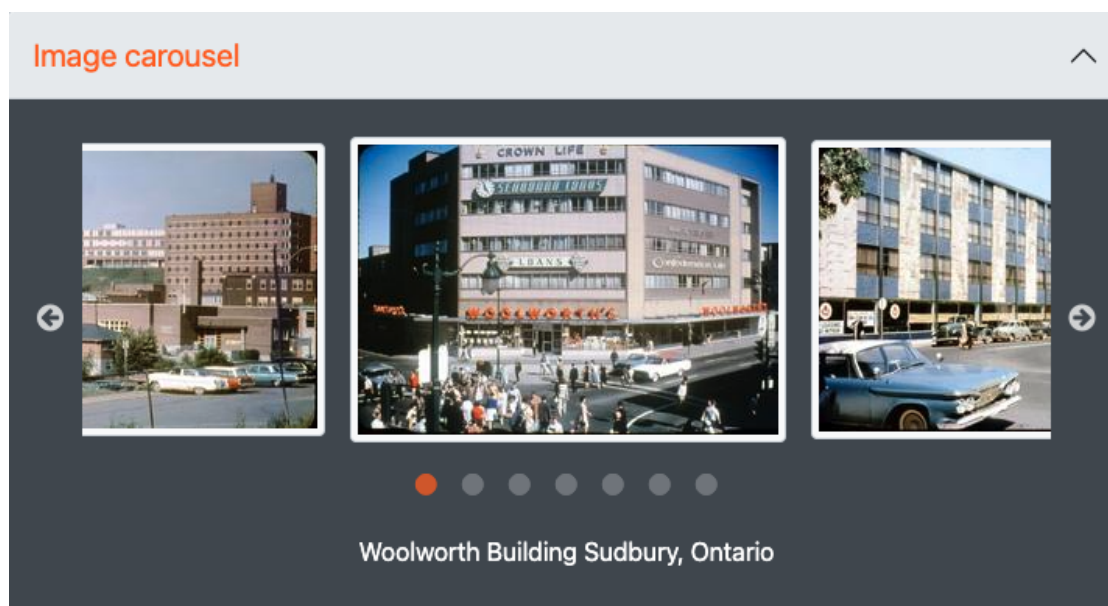




Figure 2.3: Carrousel d'images dans AtOM


Irving Steinberg Sudbury Slide Collection » Woolworth Building Sudbury, Ontario





Clipboard

 Add


Explore


 Reports

 Browse as list

 Browse digital objects

Export

 Dublin Core 1.1 XML

 EAD 2002 XML

Related places

[Canada](#) » [Ontario](#) » [Sudbury, Greater](#)

Physical storage

Hollinger box: C-005

**Identity area**

Reference code	ON00120 007-2
Title	Woolworth Building Sudbury, Ontario

Figure 2.4: Description archivistique liée à un document numérique dans AtOM

Dans le second cas, lorsque l'élément sélectionné est une pièce individuelle, si celle-ci est liée à un document numérique quel qu'il soit, alors sa représentation sera disposée avec le contenu de description de l'archive en question comme on peut le constater sur la Figure 2.4. L'utilisateur peut également si les droits lui sont accordés, cliquer sur cette représentation afin d'ouvrir la version complète du document. Ainsi, il aura la possibilité de visualiser des fichiers de plusieurs pages tels que des PDF. Cependant, si l'utilisateur souhaite effectuer des manipulations plus avancées sur le document numérique, il devra généralement enregistrer le document sur sa propre machine et l'ouvrir dans une application tierce appropriée.

Dans tous le cas, qu'elle soit liée ou non à un document numérique, une description archivistique dans AtoM regroupe un ensemble d'informations sur un ou plusieurs documents d'archives. Le Conseil international des archives définit une description archivistique comme suit : *" Représentation précise d'une unité de description, et de ses composantes éventuelles, obtenue en sélectionnant, en analysant et en ordonnant toute information permettant d'identifier, de gérer et de localiser les documents d'archives et d'expliquer leur contenu et le contexte de leur production. L'expression désigne à la fois le processus de représentation et son résultat. "[8]. Une description est divisée en sept parties principales distinctes :*

- **Zone d'identification :** cette section fournit des informations essentielles permettant d'identifier de manière unique l'unité de description (cote<sup>1</sup>, titre, niveau de description, date de création du document d'archives, etc.).
- **Zone du contexte :** elle identifie le ou les producteurs de l'unité de description, c'est-à-dire des informations sur l'organisme ou la personne responsable de la création des archives, ainsi que sur le contexte historique plus large.
- **Zone du contenu et de la structure :** elle contient une brève description du contenu du document d'archives et de sa structure interne.
- **Zone des conditions d'accès et d'utilisation :** cette section spécifie les conditions dans lesquelles les documents peuvent être consultés et utilisés (dispositions juridiques, conventions particulières, règlements intérieurs, etc.).
- **Zone des sources complémentaires :** elle répertorie les sources complémentaires pouvant être utiles pour la compréhension ou la contextualisation du document d'archives.

---

<sup>1</sup>La cote est un ensemble de symboles (lettres, chiffres, signes) identifiant un fonds, un dossier ou une pièce et correspondant à sa place dans le classement ou à son lieu d'entreposage. <https://www.filigranearchives.com/a-propos/des-archives-cest-quoi/>

- **Zone des notes** : section permettant d'inclure des renseignements ou des commentaires qui ne rentrent pas dans les catégories précédentes.
- **Zone du contrôle de la description** : elle comprend des informations qui identifient de manière unique la description elle-même ainsi que comment et par quelle institution la description a été créée et conservée.

Dans AtoM, les informations dédiées aux éventuels documents numériques se trouvent dans une section supplémentaire nommée "**Objet numérique - métadonnées**". On y retrouve par exemple : le type de fichier ainsi que sa taille, le type de support, le type MIME, etc.

En plus de tout cela, l'interface permet d'exporter les métadonnées d'une description via deux standards XML différents : le Dublin Core 1.1, qui fournit une structure assez générale et simpliste (titre, date, créateur, sujet, etc.) et l'EAD 2002 (Encoded Archival Description), entièrement compatible avec la norme ISAD(G) et fournissant des informations plus complètes sous un format structuré et spécifique pour les documents d'archives.

### **Pour les archivistes :**

Après authentification, les archivistes ont accès à des éléments supplémentaires sur leur interface leur permettant par exemple de modifier les informations relatives aux descriptions énoncées plus haut ou de paramétrer les spécificités de l'instance de leur institution.

En ce qui concerne les documents numériques, la liaison avec une description archivistique peut être réalisée de deux façons distinctes : soit en ajoutant un document directement depuis un emplacement local, soit en fournissant un lien URL vers un document externe.

### Lier objet numérique

14 - 43. Looking north toward downtown Sudbury, Ontario

Téléverser un objet numérique ^

File

Choisir un fichier    Aucun fichier choisi

The maximum size of file uploads is 520 MB.

Lier à un objet numérique externe ^

Url

http://

Annuler    Ajouter

Figure 2.5: Liaison d'un document numérique à une description dans AtoM

Lorsqu'une description est déjà liée à un objet numérique, l'archiviste a alors la possibilité de supprimer ledit objet ou de modifier certaines des données relatives à celui-ci telles que le type de média utilisé ou les différentes représentations du document.

## Chapitre 3

# L'International Image Interoperability Framework

Les normes et protocoles sont essentiels dans le domaine de la gestion et de l'accès aux archives car ils garantissent l'interopérabilité, la compatibilité et la pérennité des données et des métadonnées. En adoptant des standards communs, les institutions garantissent que leurs archives restent accessibles et exploitables sur le long terme et ce indépendamment des différentes avancées technologiques. De plus, les normes facilitent l'échange de données entre différentes plateformes et systèmes, ce qui favorise la collaboration et le partage de connaissance entre les institutions.

L'International Image Interoperability Framework (IIIF) est un exemple important de ces normes, il offre une approche standardisée pour la présentation et la manipulation d'images et de documents numériques. Dans ce chapitre, nous explorerons en détail ce protocole IIIF ainsi que les différents concepts qui lui sont attachés. Nous aborderons également les outils développés pour intégrer ce protocole dans divers environnements, offrant ainsi des fonctionnalités avancées de visualisation et de manipulation des documents numériques. En analysant ces outils, nous découvrirons comment le IIIF contribue à offrir une expérience utilisateur riche et interactive, tout en améliorant la gestion et l'accès aux documents d'archives numériques.

Pour mener à bien mon étude, j'ai dans un premier temps consulté la documentation officielle du IIIF comme point de départ. Ensuite, je me suis en grande partie appuyé sur deux documents : "Les protocoles IIIF" par Régis Robineau[9] ainsi que "Suggested measures for deploying IIIF in Swiss cultural heritage institutions" par Julien Antoine Raemy et René Schneider[10]. Les nombreuses vidéos de la chaîne Youtube officielle du IIIF sont également une source d'information très riche sur le sujet[11].

## 3.1 Présentation

Premièrement, prenons comme point de départ de notre analyse la façon dont le IIF se définit lui-même : *"IIF est un ensemble de normes ouvertes pour diffuser en ligne et à grande échelle des objets numériques de haute qualité et attribués. Il s'agit également d'une communauté internationale qui développe et met en œuvre les API IIF. IIF est soutenu par un consortium de grandes institutions culturelles"*[12].

Pour commencer, la "communauté" mentionnée fait référence à cinq entités principales :

- **Consortium IIF** : créé en 2015 pour pérenniser et piloter l'initiative IIF, le Consortium IIF (IIF-C) joue un rôle crucial dans la coordination du développement et de la promotion du IIF à l'échelle mondiale. Il compte actuellement 68 membres à travers le monde<sup>1</sup>, travaillant ensemble pour étendre l'adoption et l'utilisation des normes IIF.
- **Groupes communautaires** : ces groupes sont au nombre de onze et se réunissent pour discuter de questions liées aux thèmes et sujets d'actualité dans le domaine du IIF.
- **Groupes de spécifications techniques** : composés de quatre groupes agréés, ils sont chargés de mettre à jour les spécifications IIF existantes ou d'en créer de nouvelles.
- **Fournisseurs de logiciels** : ce sont les sociétés fournissant des services prenant en charge les normes IIF.
- **Institutions participantes** : principalement composées d'institutions du secteur du patrimoine culturel, celles-ci ont adopté IIF et contribuent à son expansion.

La création du IIF découle de problématiques communes et récurrentes à de nombreuses institutions. Tout d'abord, le premier obstacle fut le manque d'interopérabilité entre les systèmes, limitant ainsi la capacité des institutions à échanger efficacement des données entre leurs plateformes. De plus, le développement d'outils et de visionneuses spécifiques pour chacune d'entre elles entraînait des coûts élevés ainsi qu'une complexité de maintenance accrue. La Figure 3.1 illustre bien cette problématique de duplication et d'étanchéité des systèmes souvent mentionnée comme des "silos". Enfin, alors que les chercheurs utilisaient de plus en plus les pratiques numériques dans leurs travaux de recherche, les ressources numériques

---

<sup>1</sup><https://iif.io/community/consortium/members/>

existantes ne subvenaient pas toujours à leurs besoins. Ces différents problèmes ont soulevé l'importance et la nécessité de développer des normes et des pratiques communes afin de simplifier la gestion et l'accès aux ressources numériques.

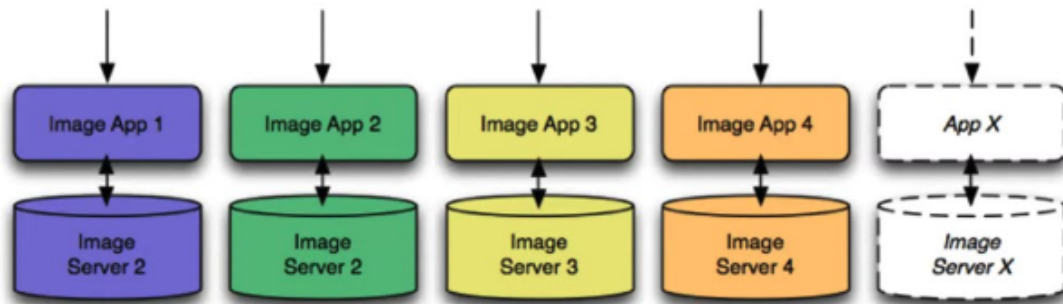


Figure 3.1: Le problème des silos[13]

Après avoir abordé les principaux acteurs et les motivations qui ont conduit à la création du IIIF, il est maintenant important de définir plus concrètement ce qu'est le IIIF et ce qu'il offre. Le IIIF propose une approche standardisée et un cadre technique commun pour la présentation et la manipulation d'images et de documents numériques sur le Web. Concrètement, il permet aux utilisateurs d'accéder à des objets numériques, sous différents formats et en haute qualité. Il n'y a pas de dépendance vis-à-vis d'un fournisseur, ce qui permet d'utiliser n'importe quel logiciel compatible pour visualiser et manipuler les documents. De plus, le format des documents IIIF (les Manifestes) encourage l'utilisation de métadonnées riches sur les ressources, simplifiant ainsi la recherche et l'exploration de celles-ci.

## 3.2 Spécifications

Passons maintenant à l'aspect plus technique des normes IIIF. Le cadre normatif de IIIF établit des interfaces de programmation applicative (API) standardisées, celles-ci offrent une méthode uniforme pour la description et l'accès aux images sur le Web. Ce sont justement ces APIs qui vont permettre de surmonter le problème des silos abordé plus haut, en permettant aux institutions de partager efficacement des données entre leurs plateformes, sans dépendre de solutions propriétaires. L'API IIIF fonctionne comme une couche intermédiaire interagissant entre les clients et les serveurs, ce qui facilite grandement l'échange de données comme nous pouvons le constater sur la Figure 3.2.

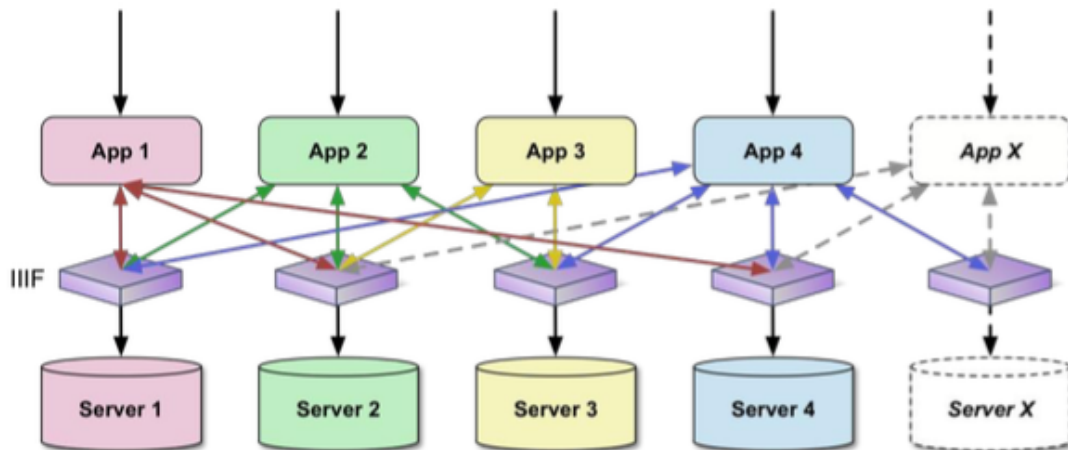


Figure 3.2: API IIIF dans le modèle client-serveur[10]

Actuellement, les APIs IIIF sont au nombre de six et chacune joue un rôle spécifique dans la gestion et l'accès aux ressources numériques :

- **Image API (3.0)** : elle spécifie un service Web qui renvoie une image en réponse à une requête HTTP ou HTTPS. Elle rend accessible et paramétrable les images IIIF par le biais de leurs URI.
- **Presentation API (3.0)** : le rôle de cette API est de fournir toutes les informations de description des objets numériques. Elle permet la visualisation des documents dans un environnement riche.
- **Authorization Flow API (2.0)** : cette spécification sert de passerelle entre une application cliente et un système de contrôle d'accès. Son but est de guider l'utilisateur dans l'accès à une ressource grâce à un ensemble d'opérations.
- **Content Search API (2.0)** : cette API permet la recherche au sein de ressources IIIF, notamment en se basant sur les annotations présentes dans celles-ci.
- **Change Discovery AP (1.0)** : elle met en place un mécanisme permettant la découverte et ensuite l'utilisation de ressources IIIF. Un exemple significatif de son application est la création de moteurs de recherche et de portails, tels que ceux lancés par Cultural Japan<sup>2</sup> ou Biblissima<sup>3</sup>.

<sup>2</sup><https://cultural.jp/>

<sup>3</sup><https://iiif.biblissima.fr/collections/>

- **Content State API (1.0)** : elle permet de faire référence à une ressource ou à une partie d'une ressource en pointant sur celle-ci dans un format compact dont la vue peut être initialisée par un client.

Dans le cadre de ce mémoire, nous examinerons plus en profondeur l'Image API ainsi que la Presentation API. En effet, ce sont en particulier ces deux APIs que nous manipulerons et il est essentiel de comprendre leurs fonctionnements afin d'aborder au mieux leurs utilisations en situation réelle.

## Image API :

L'Image API fournit une méthode standardisée permettant la récupération et la manipulation d'images à distance en réponse à des requêtes HTTP ou HTTPS standards. Plus concrètement, les images sont accessibles via une syntaxe d'URI spécifique et il existe deux types de requêtes distinctes pouvant être réalisées :

- La requête permettant la récupération de l'image réelle en tant que telle, dont voici le modèle :

```
{scheme}://{server}/{prefix}/{identifiant}/
{region}/{size}/{rotation}/{quality}.{format}
```

Ce mécanisme permet à un utilisateur de solliciter et de manipuler une image en fonction de paramètres tels que la région, la taille, la rotation, la qualité et le format. Illustrons et testons son utilisation avec un exemple<sup>4</sup> issu d'une collection des archives de l'UCLouvain[14]:

```
1      https://archives.uclouvain.be/iiif-img/575093/
      full/200%2C134/0/default.jpg

2      https://archives.uclouvain.be/iiif-img/575093/
      600,170,120,120/200,/0/gray.jpg
```

Ici, dans la seconde version, les paramètres ont été modifiés afin de ne récupérer que la partie où se trouve la tête de Georges Lemaître. L'image a également été redimensionnée pour faire 200 pixels de largeur et la qualité a été fixée en niveaux de gris. Le résultat est représenté par la Figure 3.3.

<sup>4</sup><https://archives.uclouvain.be/ark%3A/33176/dli000000ff3M#?c=0&m=0&s=0&cv=0&xywh=195%2C-15%2C680%2C544>



Figure 3.3: Paramétrisation d'une image IIF par le biais de son URI

- La requête fournissant des informations sur l'image, suivant ce modèle :

```
{scheme}://{server}/{prefix}/{identifiant}/info.json
```

De cette façon, les métadonnées techniques relatives à l'image sont retournées au format JSON. Vous trouverez à l'annexe B le contenu du fichier "info.json" relatif à l'exemple précédent.

## Presentation API :

Contrairement à l'Image API qui se concentre principalement sur la distribution d'une seule image à la fois, la Presentation API fournit les métadonnées nécessaires à la représentation d'une ressource complète dans une interface. L'ensemble de ces données est compris dans un fichier qu'on appelle un "Manifeste". Un Manifeste est un fichier au format JSON-LD<sup>5</sup>, que l'on peut se représenter comme un contenant virtuel, formant l'unité de distribution de base dans l'écosystème IIF. Les données qui composent un Manifeste sont très diverses et permettent d'offrir une représentation très détaillée de la ressource et de son contexte. Elles comprennent : des métadonnées descriptives (titre, auteur, etc.), techniques (format, comportement de navigation, etc.) et de structure (index, table des matières, etc.), des références à des liens vers des ressources externes ou à des services Web tels que d'autres APIs IIF ainsi que des annotations (commentaires, tags, etc.).

<sup>5</sup><https://json-ld.org/>

Au niveau de sa structure, le tronc commun d'un Manifeste IIIF reste généralement identique. Premièrement, un Manifeste est composé de plusieurs sections appelées "Canevas". Un Canevas est en quelque sorte une "vue" d'un objet, il fournit un cadre de référence pour la disposition du contenu. De plus, ces différents Canevas sont eux-mêmes encapsulés dans ce qu'on appelle des "Séquences", permettant de regrouper différents Canevas entre eux. Enfin, il est également possible de regrouper un ensemble de Manifestes, la ressource permettant cela est appelée une "Collection", il s'agit simplement d'un document JSON listant les liens vers différents Manifestes.

Vous trouverez aux annexes C.1 et C.2 un exemple<sup>6</sup> du contenu d'un Manifeste provenant encore une fois d'une collection des archives de l'UCLouvain[15] ainsi qu'une illustration sur ce même exemple permettant de comprendre de manière plus visuelle les rôles distincts des deux APIs décrites ici.

### 3.3 Visionneuses IIIF existantes

Compte tenu de la popularité croissante du IIIF dans le domaine de la gestion et de l'accès aux ressources numériques, de nombreuses solutions logicielles ont vu le jour pour répondre aux besoins variés des institutions. Ces solutions comprennent notamment : des serveurs d'images et d'annotations, des outils de création et d'édition de Manifestes, des visionneuses, etc.

Nous allons ici porter notre attention sur les visionneuses IIIF, dont le but principal est de proposer aux utilisateurs une visualisation riche et interactive des ressources numériques. Ces visionneuses utilisent principalement l'API Image et Presentation où la première est utilisée pour récupérer les images dans différents formats et résolutions, tandis que l'API de Presentation est utilisée pour obtenir la structure et le contenu de la ressource à afficher pour l'utilisateur. Comme représenté par la Figure 3.4, voici le scénario d'interaction typique client-serveur lors de l'utilisation d'une visionneuse IIIF par un utilisateur :

1. Pour commencer, le client envoie une requête au serveur pour récupérer le Manifeste JSON de la ressource qu'il souhaite accéder.
2. En second lieu, le serveur répond à la demande du client en envoyant le Manifeste.

---

<sup>6</sup><https://archives.uclouvain.be/ark%3A/33176/dli000000Cj30#?c=0&m=0&s=0&cv=0&xywh=-132%2C-1308%2C5175%2C3579>

3. Après analyse du Manifeste reçu, le client envoie une requête au serveur afin de récupérer les informations sur une image ou une plage d'images au sein de l'objet numérique qu'il souhaite afficher.
4. Le serveur répond alors à la demande du client en envoyant les informations d'image demandées, toujours au format JSON.
5. Ensuite, le client envoie une requête au serveur pour récupérer les données d'image réelles.
6. Si tout se passe bien, le serveur répond à la demande du client et ce dernier peut alors afficher l'image dans la visionneuse.

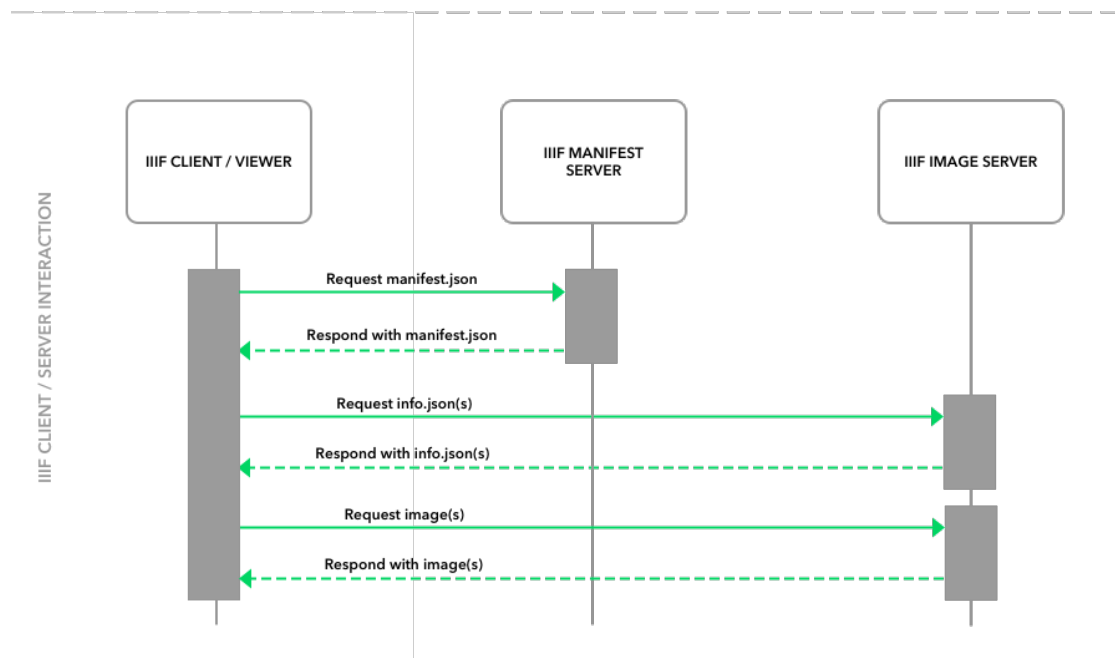


Figure 3.4: Interaction client-serveur au sein d'un environnement IIIIF[10]

Examinons et comparons à présent les principales solutions disponibles sur le marché des visionneuses compatibles. Il en existe de différentes catégories, chacune offrant des fonctionnalités plus ou moins avancées et des avantages spécifiques adaptés à différents besoins et cas d'utilisation. Lorsqu'on effectue des recherches sur le sujet, on constate rapidement que certains noms reviennent fréquemment parmi les solutions de visionneuses les plus populaires : Universal Viewer, Mirador, Tify, IIIIF Curation Viewer, OpenSeadragon, Annona, Clover, etc. Parmi ces

options, Universal Viewer (UV)<sup>7</sup> et Mirador<sup>8</sup> se démarquent particulièrement dans le cadre de ce mémoire notamment pour leur caractère intégrable, leur adoption répandue ainsi que leur capacité à répondre aux besoins spécifiques en matière de manipulation d'images. De plus, elles ont toutes deux obtenu d'excellents résultats sur l'échelle d'utilisabilité du système (System Usability Scale) et ont offert une très bonne expérience utilisateur aux participants lors de tests réalisés sur celles-ci [16]. Comparons à présent les principales spécificités de ces deux outils :

### **Caractéristiques communes :**

- Prise en charge de Manifestes IIIF
- Fonctionnalités de zoom et de navigation
- Affichage riche de métadonnées
- Différents modes de visualisation des ressources
- Maintenance active et documentation fournie
- Open source

### **Universal Viewer :**

- + Rotation d'images

### **Mirador :**

- + Comparaison d'images intégrée à l'interface
- + Prise en charge complète des annotations
- + Deux types d'espace de travail (Élastique et Mosaïque)
- + Possibilité d'ajouter des plugins<sup>9</sup>

---

<sup>7</sup><https://universalviewer.io/>

<sup>8</sup><https://projectmirador.org/>

<sup>9</sup>Les différents plugins se trouvent sur le profil Github officiel du projet Mirador : <https://github.com/ProjectMirador>

# Chapitre 4

## Limitations actuelles et besoins

Ce chapitre est dédié à l'analyse de notre problème, il vise à décrire les tenants et les aboutissants de ce mémoire à la manière d'un cahier des charges. Ainsi, nous exposerons notre problématique, l'objectif que nous poursuivons ainsi que les acteurs et les besoins relatifs à ceux-ci.

### 4.1 Problématique

AtoM est une application Web très complète et efficace en ce qui concerne la recherche et la publication de descriptions archivistiques dans un catalogue. Cependant, la gestion actuelle de ces documents numériques présente des limitations notables. En effet, sa visionneuse est considérée comme rudimentaire par rapport aux standards actuels de consultation d'archives en ligne, notamment en ce qui concerne l'expérience utilisateur et les fonctionnalités possibles. Pour illustrer ces propos, nous pouvons observer les pratiques adoptées par des institutions telles que FranceArchives[17] ou la bibliothèque d'Harvard[18], qui ont intégré les spécifications IIIF pour la gestion et la consultation de leurs documents numériques d'archives.

La principale problématique avec la consultation des documents numériques dans AtoM réside dans l'incapacité des utilisateurs à visualiser et à explorer les images d'archives au-delà de la simple consultation d'images ou de PDF. En d'autres termes, il n'est pas possible pour les utilisateurs de zoomer sur les images, de les comparer ou d'effectuer d'autres manipulations interactives qui sont devenues des attentes standards dans de nombreuses plateformes de consultation d'archives actuelles.

En plus de cela, cette situation est accentuée par le fait que de nombreuses

institutions utilisent des systèmes indépendants afin de stocker et de gérer leurs documents numériques et leurs métadonnées. Ces systèmes sur mesure, bien que permettant de répondre aux besoins précis de chaque institution, entraînent des coûts élevés de maintenance et de modernisation. D'autre part, pour les utilisateurs, l'accès aux documents sous différents standards provenant de différentes institutions peut être fastidieux car nécessitant l'utilisation de plusieurs interfaces et visionneuses.

Enfin, bien que ce mémoire ait été réalisé en étroite collaboration avec le service des Archives de l'UCLouvain, il est important de préciser que la volonté d'intégrer le cadre IIF dans AtoM n'est pas une aspiration isolée. Il suffit de se pencher quelques instants sur le forum de discussion des utilisateurs d'AtoM pour constater que nous sommes confrontés à une réelle demande dans la communauté des institutions qui utilisent cet outil[19].

## 4.2 Objectif

L'objectif de ce mémoire est d'explorer les possibilités offertes par le cadre IIF et de travailler sur l'intégration open source d'une visionneuse compatible dans AtoM afin d'améliorer la gestion et la consultation d'archives numériques.

## 4.3 Acteurs

Deux types d'acteurs principaux seront amenés à utiliser cette solution :

- Les archivistes : ils sont les professionnels responsables de la gestion des archives au sein des institutions utilisant AtoM. Leur rôle couvre la création, la description, l'administration et la conservation des archives numériques.
- Les visiteurs : ils comprennent les chercheurs, les historiens, les étudiants et d'autres personnes curieuses intéressées par la consultation de documents d'archives. Leur objectif premier est d'avoir accès aux informations contenues dans les documents d'archives numériques afin de mener des recherches, qu'elles soient académiques, professionnelles ou personnelles.

## 4.4 Besoins fonctionnels

Les besoins fonctionnels décrivent les actions concrètes que les différents types d'utilisateurs doivent pouvoir accomplir.

## Pour les visiteurs :

- **Visualisation des Manifestes IIIF depuis une visionneuse intégrée :** permet aux visiteurs d'accéder et de visualiser les Manifestes IIIF liés à des descriptions archivistiques directement depuis AtoM.
- **Capacité de zoomer sur les documents :** possibilité depuis la visionneuse de zoomer sur les documents numériques pour les examiner en détails.
- **Navigation aisée :** la visionneuse doit permettre une navigation fluide entre les différentes pages d'un document numérique.
- **Comparaison de documents :** possibilité de comparer côte à côte deux ou plusieurs documents depuis une même interface.
- **Différents modes de visualisation :** fournir aux visiteurs différents modes de visualisation et de disposition des documents.
- **Accès aux métadonnées des documents :** permet aux visiteurs d'accéder aux métadonnées associées aux documents numériques.

## Pour les archivistes :

- **Activation/désactivation de la visionneuse IIIF :** donne la possibilité d'activer ou de désactiver la visionneuse IIIF depuis les paramètres d'AtoM. Lorsque la visionneuse est désactivée, le comportement de base d'AtoM reprend le dessus, offrant une visualisation statique des documents numériques sans les fonctionnalités avancées offertes par la visionneuse.
- **Ajout de Manifestes IIIF :** permet aux archivistes de lier des Manifestes IIIF à des descriptions archivistiques et ainsi de les rendre accessibles depuis la visionneuse.
- **Modification/suppression de Manifestes :** permet aux archivistes de modifier certaines données ou de supprimer les liens entre les Manifestes IIIF et les descriptions archivistiques, comme cela est possible pour les documents qui ne sont pas des Manifestes.

## 4.5 Besoins non fonctionnels

Les besoins non fonctionnels se concentrent sur les aspects de qualité et de contraintes du système, bien que ces besoins ne soient pas directement liés aux

fonctionnalités spécifiques du système, ils jouent un rôle crucial dans la réussite globale du développement.

- **Conception visuelle et ergonomie** : il est important d'assurer une conception visuelle attrayante et une ergonomie optimale en accord avec le style d'AtoM afin de garantir une expérience utilisateur fluide et cohérente.
- **Performance** : la solution doit garantir de bonnes performances en termes de temps de chargement et de réactivité de l'interface utilisateur.
- **Open source** : toute intégration et développement doivent être réalisés en open source.
- **Intégration conforme aux standards d'AtoM** : l'intégration doit respecter les normes et les standards de développement du codebase existant d'AtoM, tout en tenant compte des indications définies pour la contribution et la collaboration au projet.

# Chapitre 5

## Implémentation open source de la visionneuse IIIF Mirador dans AtoM

Dans ce chapitre, nous parcourons les différentes étapes clés nécessaires à l'intégration de la visionneuse Mirador au sein de l'environnement d'AtoM. Nous couvrirons l'installation de l'instance d'AtoM dédiée au développement, l'implémentation concrète de Mirador, ainsi que les différents choix et considérations entrant en ligne de compte dans le processus d'incorporation.

Le projet nécessitant un développement open source, l'intégralité du code réalisé se trouve dans un dépôt Github public que vous pouvez retrouver à l'annexe A. Il s'agit d'un fork<sup>1</sup> du projet AtoM officiel<sup>2</sup>.

### 5.1 Mise en place d'AtoM

À la base du processus de développement et avant même de pouvoir parler d'une quelconque intégration, il est bien entendu essentiel de procéder à l'installation initiale d'AtoM[20]. Pour ce faire, je me suis dirigé initialement vers la version 2.4 d'AtoM utilisée par les archives de l'UCLouvain mais les différentes étapes décrites ici restent en grande partie valables pour les versions ultérieures de l'application.

Pour commencer, le code d'AtoM doit être installé dans un environnement contenant Apache (NGINX est également possible). En effet, AtoM étant développé

---

<sup>1</sup>Un fork est une copie d'un projet existant permettant de travailler sur sa propre version du projet sans modifier directement l'original.

<sup>2</sup><https://github.com/artefactual/atom>

en PHP et utilisant Elasticsearch pour la recherche et l'indexation des données, il est essentiel pour ces technologies d'être utilisées sur un serveur Web comme Apache permettant leur exécution. Il est également primordial de s'assurer que la version de PHP requise par la version d'AtoM choisie est installée sur le même serveur.

Ensuite, avant de commencer l'ajustement de l'application sur le serveur, il est important d'assurer la disponibilité d'une base de données MySQL pour AtoM. Cela inclut la création d'une base de données dédiée, l'attribution des droits d'accès appropriés ainsi que la configuration des paramètres de connexion dont nous aurons besoin lors de la phase de configuration.

De plus, Elasticsearch ainsi que Java, nécessaire au fonctionnement d'Elasticsearch, doivent être installés en parallèle d'AtoM. Il est également important de s'assurer que le service est bien démarré avant de lancer la phase de configuration.

Enfin, une fois toutes les étapes précédentes réalisées, le paramétrage de l'application peut commencer. Cette phase peut se réaliser soit en exécutant l'installateur en ligne de commande, soit en accédant directement à AtoM via le navigateur Web et en suivant le processus de configuration à partir de là. Dans les deux cas, toutes les informations nécessaires à l'accès à la base de données préalablement créée sont demandées (emplacement, port, identifiants, etc.). Cela permet la génération automatique de toutes les tables de base de données nécessaires pour AtoM ainsi que l'ajout des données initiales. Ensuite, de la même manière, les détails relatifs à Elasticsearch sont demandés (emplacement, port, index de recherche, etc.). Enfin, la configuration finale de l'application implique de remplir les informations restantes telles que le titre du site, la description, l'URL de base du site ainsi que les identifiants permettant la connexion au profil administrateur par défaut.

La configuration de l'environnement de développement utilisée dans le cadre de ce mémoire est la suivante :

- AtoM 2.4
- Apache 2.4.54
- PHP 5.6.4
- MySQL 5.7.39
- Elasticsearch 1.7.6

- Java 1.8
- macOS Ventura

## 5.2 Choix de la visionneuse

Lors de la réflexion sur la visionneuse à intégrer, les deux options principales énoncées plus haut, Universal Viewer et Mirador semblaient initialement toutes deux prometteuses. Cependant, Mirador s’est avéré être le choix le plus judicieux pour plusieurs raisons.

Tout d’abord, Mirador offre une option essentielle qui manque à UV : la capacité de comparer des ressources directement depuis la visionneuse. En effet, cette fonctionnalité intégrée s’est avérée être décisive dans notre choix et offre un avantage significatif pour le projet, notamment dans le cadre de la collaboration avec les archives de l’UCLouvain. Effectivement, la mise en parallèle de ressources telles que des documents historiques ou des manuscrits permet une analyse comparative facile et rapide des documents, offrant ainsi une expérience utilisateur accrue.

Un autre aspect qui a également pesé dans la balance est la flexibilité de Mirador. Son architecture est conçue de façon à permettre une grande personnalisation de la visionneuse[21], offrant ainsi une meilleure maintenabilité et évolutivité du système dans le temps. Cette notion est cruciale pour garantir que le système et l’intégration puissent évoluer et s’adapter aux besoins futurs des utilisateurs et éventuels changements. De plus, les différents plugins disponibles pour Mirador vont également dans ce sens et tendent à renforcer son évolutivité future.

En ce qui concerne le choix de la version de Mirador, cette dernière se décline en deux versions majeures. Bien que la version 2 existe depuis de nombreuses années et est en production dans de nombreuses institutions, le choix s’est orienté vers la version 3. En effet, la version 3 de Mirador, publiée en 2020, bénéficie quant à elle encore d’un développement actif et d’une adoption croissante au sein de la communauté.

Vous trouverez à l’annexe D différentes représentations annotées issues d’une instance Mirador mise en place localement permettant d’illustrer plus visuellement les fonctionnalités et caractéristiques de cette visionneuse.

## 5.3 Codebase d'AtoM

Le "codebase d'AtoM" fait ici référence à l'ensemble du code source relatif à la plateforme. Cela comprend tous les fichiers, modules et autres composants qui constituent le logiciel mais également la façon dont est conçue son architecture. L'analyse de ce codebase est essentielle afin de comprendre le fonctionnement interne de l'application ainsi que les fonctionnalités disponibles et les différentes possibilités d'extension.

Voici un tour d'horizon non exhaustif des principales sections utiles dans ce mémoire constituant l'architecture du code d'AtoM :

- **Modules** : les différents composants de l'interface utilisateur d'AtoM sont structurés en modules distincts. Ils sont situés dans le dossier "apps/qubit/modules" où chaque sous-dossier est nommé explicitement et correspond à un module (exemple : le dossier "/digitalobject" est le module gérant les fichiers relatifs à la représentation ainsi qu'à la gestion des objets numériques). Chaque module contient ensuite deux dossiers, "/actions" et "/templates" permettant respectivement de jouer le rôle de contrôleur et de vue. En effet, le framework Symfony utilisé ici propose un modèle MVC (Model-view-controller) permettant de séparer l'interface représentant les informations pour l'utilisateur (les vues) avec la partie logique des interactions effectuées (les contrôleurs). Ce modèle de conception favorise grandement la réutilisation de composants dans le système et permet ainsi une meilleure maintenance et aptitude aux modifications.
- **JavaScript** : le dossier "/js" situé à la racine du code de l'application regroupe les différents fichiers JavaScript utilisés permettant notamment la gestion de fonctionnalités dynamiques pour l'application.
- **Plugins** : différents types de plugins sont disponibles dans AtoM[22], ils permettent notamment la modification du thème graphique de l'application ou l'ajout de fonctionnalités supplémentaires. Parmi ces plugins, certains sont dédiés à la gestion des normes de métadonnées, jouant ainsi un rôle essentiel dans la structuration des informations archivistiques au sein d'AtoM (par exemple : le plugin "/sfIsadPlugin" contient tout le code relatif à la mise en forme des données suivant les règles ISAD(G)).
- **Modèles** : les "modèles" présents dans le dossier "/lib/model" correspondent ici au dernier élément du motif d'architecture MVC mentionné plus haut. Concrètement, ils sont responsables de la manipulation et du traitement des données de l'application, ainsi que de leurs interactions avec la base de

données. Ils jouent donc également un rôle important dans l'architecture d'AtoM car ils assurent la gestion des données qui alimentent les vues et sont manipulées par les contrôleurs.

## 5.4 Intégration de Mirador

Dans le processus d'intégration, la première étape consiste bien évidemment à installer Mirador au sein du projet AtoM initialement configuré. Cela peut être réalisé de plusieurs manières, mais la documentation de Mirador conseille de passer par un gestionnaire de paquets Node.js tel que npm. Si vous récupérez le code du mémoire depuis GitHub, il suffit d'exécuter la commande "npm install" dans le répertoire racine du projet installé afin de télécharger automatiquement les dépendances (et donc Mirador) grâce au fichier de configuration "package.json" inclus dans le dépôt.

Le développement de la solution a été réalisé dans la continuité des bonnes pratiques et architectures définies par AtoM. Voici dans les grandes lignes, les principaux composants mis en place :

### Prise en charge de Manifestes IIIF :

Dans un premier temps, il faut permettre aux Manifestes d'être reconnus dans le système en tant que type de fichier à part entière, de sorte que le comportement de l'interface soit adapté en fonction de leur présence. Pour ce faire, la classe "DigitalObjectShowComponent" étant responsable de la logique d'affichage des objets numériques en fonction de leur type a été modifiée. Cette classe se trouve dans les contrôleurs du module "/digitalobject". Ces modifications incluent la vérification du type de fichier via une méthode "isIIIFManifest" ainsi que l'appel au composant responsable de l'affichage des Manifestes IIIF :

```
$this->showComponent = 'showIIIFManifest';
```

En effet, dans AtoM, chaque type d'objet numérique (image, vidéo, etc.) possède une vue et un contrôleur qui lui est propre, permettant ainsi d'adapter le comportement de l'interface en conséquence. De ce fait, une classe "DigitalObjectShowIIIFManifestComponent" ainsi qu'un fichier "\_showIIIFManifest.php" ont été ajoutés au module afin de gérer spécifiquement les Manifestes .

## Affichage de Mirador :

Les composants créés plus haut sont donc à proprement parler responsables de l'affichage de la visionneuse au sein d'AtoM. En effet, notre contrôleur va appeler les ressources essentielles telles que le paquet contenant Mirador, le fichier CSS ainsi que le script "mirador.js" permettant l'instanciation de notre visionneuse :

```
$this->response->addJavaScript('../node_modules/mirador/dist/mirador.min.js', 'last');  
$this->response->addJavaScript('mirador', 'last');  
$this->response->addStylesheet('mirador', 'last');
```

Le script comprend une fonction qui est ensuite appelée dans la vue, celle-ci prend plusieurs paramètres dont "jsonlink" fournissant le lien vers le Manifeste à afficher précédemment détecté par le système :

```
function renderMiradorViewerComponent(jsonlink, catalog, settings)
```

De cette manière, chaque fois que l'application voudra afficher la version complète d'une ressource s'avérant être un Manifeste, celle-ci sera représentée dans Mirador.

## Synchronisation du catalogue :

L'interface graphique de Mirador met à disposition un composant intéressant, son "catalogue". Il s'agit d'une liste de Manifestes répertoriés au sein d'une instance Mirador (la deuxième représentation de l'annexe D illustre cela). Afin de faciliter la comparaison des ressources dans notre environnement comme énoncé dans les besoins fonctionnels du projet, une synchronisation a été réalisée entre ce catalogue et l'arborescence présente dans AtoM. Les différentes méthodes gérant la récupération des éléments dans l'arborescence se trouvent dans la classe "QubitDigitalObject" (cf. "lib/model/"). De plus, la fonction "renderMiradorViewerComponent" présentée plus haut possède un paramètre "catalog", ce dernier est un tableau censé contenir les éléments de l'arborescence actuelle de la page. Ainsi, cela permet de faire le lien entre les différentes couches d'abstraction de l'application.

## Paramètres :

Pour permettre aux administrateurs de configurer certains éléments liés à la visionneuse, une section spécifique dédiée à la prise en charge et l'affichage du IIIF a été

mise en place dans les paramètres généraux de l'application. À la manière de ce qui a été implémenté pour les objets numériques, une classe "SettingsIIIFViewerAction" ainsi qu'une vue "iiifViewerSuccess.php" ont été mises en place dans le module "/settings". Le paramètre "settings" de la fonction "renderMiradorViewerComponent" est une chaîne JSON spécifiant certaines des configurations définies par les administrateurs ou par l'utilisateur lui-même (comme par exemple la langue choisie pour AtoM qui est ainsi synchronisée pour définir celle de la visionneuse).

Au niveau de l'interface, trois sous-sections ont été ajoutées :

```
case 'mirador':
    $this->createOrUpdateSetting($this->miradorSetting,
        'iiifviewer_mirador', $field->getValue());
...
case 'miradorCatalog':
    $this->createOrUpdateSetting($this->miradorCatalogSetting,
        'mirador_catalog', $field->getValue());
...
case 'miradorView':
    $this->createOrUpdateSetting($this->miradorViewSetting,
        'mirador_view', $field->getValue());
...
```

Ces trois champs permettent de définir :

- **L'activation/désactivation de Mirador** : permet de désactiver la prise en charge du IIIF dans le système comme définie plus haut.
- **Le type de synchronisation du catalogue** : offre la possibilité de sélectionner quelle méthode de récupération des données de l'arborescence va être utilisée pour la génération du catalogue.
- **La type de vue par défaut de Mirador** : propose le choix entre la vue "galerie" et la vue "simple" lors du chargement initial d'une ressource dans la visionneuse pour les utilisateurs.

L'approche utilisée facilite grandement l'ajout futur de nouvelles configurations dans AtoM pour la visualisation IIIF ainsi que le transfert des configurations vers Mirador.

# Chapitre 6

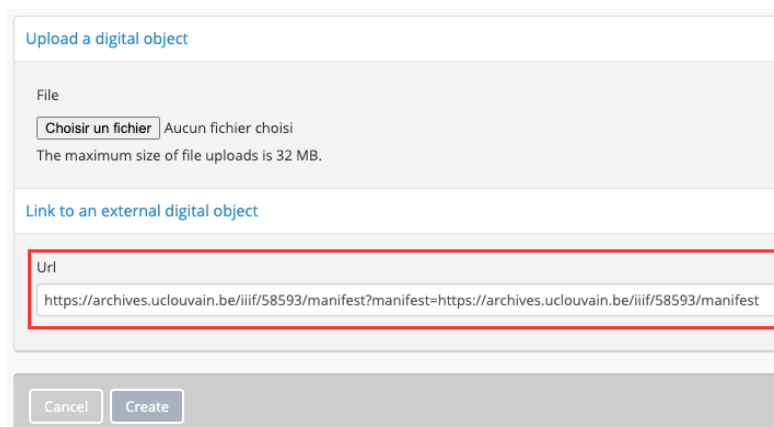
## Résultats et expérimentations

Au travers de ce chapitre, nous passerons en revue l'aboutissement des différentes fonctionnalités développées. Nous examinerons également certaines expérimentations réalisées pour évaluer les performances de notre intégration, en particulier l'impact de la synchronisation avec le catalogue de Mirador.

### 6.1 Résultats

Voici les résultats obtenus à la suite du processus de développement conformément aux différents points définis dans le chapitre précédent :

#### Prise en charge de Manifestes IIIF :



The screenshot shows a web form titled "Upload a digital object". It has two main sections. The first section, "File", contains a button labeled "Choisir un fichier" and the text "Aucun fichier choisi" and "The maximum size of file uploads is 32 MB.". The second section, "Link to an external digital object", contains a text input field labeled "Url" with the value "https://archives.uclouvain.be/iiif/58593/manifest?manifest=https://archives.uclouvain.be/iiif/58593/manifest". The input field is highlighted with a red border. At the bottom of the form are two buttons: "Cancel" and "Create".

Figure 6.1: Ajout d'un Manifeste via son URI

Comme représenté sur la Figure 6.1, lorsqu'un utilisateur possédant les droits d'accès nécessaires souhaite ajouter un Manifeste en tant qu'objet numérique pour une description archivistique, il suffit à ce dernier de se rendre sur la page de liaison de la description et d'ajouter l'URI de ce Manifeste comme cela peut être réalisé pour tous les autres types de ressources (l'ajout de Manifestes "locaux" est également possible). Si le Manifeste fourni s'avère être corrompu ou non conforme, celui-ci ne sera pas détecté comme tel et le comportement par défaut du système sera appliqué.

## Affichage de Mirador :

Pour les utilisateurs visitant une description archivistique liée à un Manifeste IIIF, la traditionnelle représentation de la ressource est remplacée par la visionneuse Mirador, disposée et utilisable directement au sein de l'environnement :

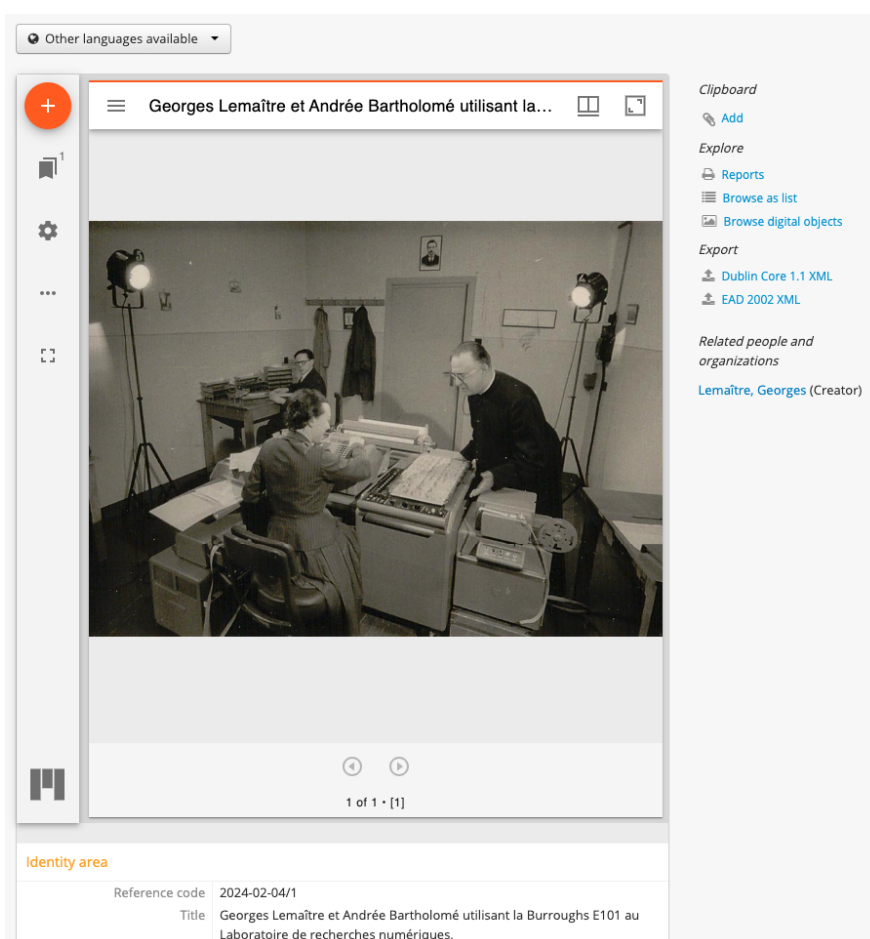


Figure 6.2: Mirador au sein de l'interface d'AtoM

Pour les administrateurs souhaitant modifier l'objet numérique d'une description, la représentation de Manifestes via Mirador est également prise en compte au sein de la page de modification.

## Synchronisation du catalogue :

En cliquant sur le bouton "+" situé dans le bord supérieur gauche de la visionneuse, l'utilisateur a la possibilité d'afficher le catalogue de Mirador reprenant les éléments présents dans l'arborescence dans laquelle il se trouve (en fonction des paramètres définis dans AtoM, plus ou moins d'éléments peuvent être présents) :

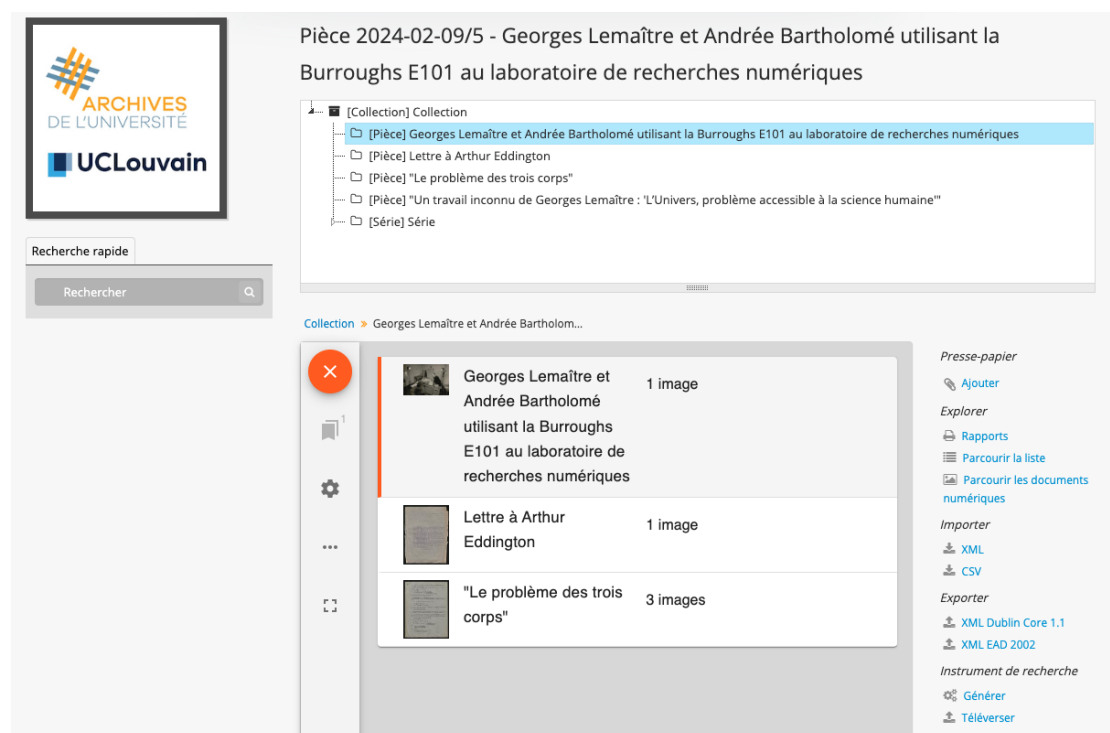


Figure 6.3: Synchronisation entre l'arborescence d'AtoM et le catalogue Mirador

De cette manière, s'il souhaite comparer le document actuel avec une ou plusieurs autres ressources présentes dans la liste, il lui suffit de cliquer sur celles-ci et la vue sera mise à jour en conséquence. L'utilisateur peut également, s'il le souhaite, ajouter un Manifeste externe à la comparaison en cliquant sur le bouton dans le coin inférieur droit de l'interface du catalogue (comme illustré sur la deuxième représentation de l'annexe D). À noter qu'une ressource ajoutée de la sorte sera stockée localement mais de manière non persistante et temporaire dans le système.

## Paramètres :

Voici à présent la section des paramètres d'AtoM accessibles aux administrateurs et dédiée à la représentation des ressources IIIF :

The screenshot shows the 'IIIF Viewer' settings page. On the left is a sidebar menu with the following items: Global, Site information, Default page elements, Default template, User interface labels, I18n languages, Finding Aid, Security, Permissions, Inventory, Digital object derivatives, DIP upload, Treeview, IIIF Viewer (highlighted), and Privacy Notification. The main content area is titled 'IIIF Viewer' and contains three sections:

- IIIF Viewer settings**: A 'Mirador' section with two radio buttons: 'No' and 'Yes' (selected).
- Mirador Catalog**: A section titled 'Mirador Catalog (elements in the viewer resources)' with three radio buttons: 'Direct Children From Same Parent' (selected), 'All Children From Same Parent', and 'All Children From Root'.
- Mirador view**: A section titled 'Mirador default window view' with two radio buttons: 'Single' (selected) and 'Gallery'.

A 'Save' button is located at the bottom of the settings area.

Figure 6.4: Paramètres IIIF dans AtoM

Lorsqu'une nouvelle configuration est sélectionnée et sauvee, les modifications apportées sont immédiatement appliquées pour tous les utilisateurs de l'instance d'AtoM.

Vous trouverez aux annexes E.1 et E.2 des comparaisons illustrant les changements apportés avec ces paramètres.

## 6.2 Expérimentations sur les performances

Si l'on désire aborder des aspects de performance et de chargement des éléments dans notre système, il est légitime de se questionner sur l'impact concret de la prise en compte et de la synchronisation d'un grand nombre de Manifestes depuis le catalogue. En effet, les services d'archives comme celui de l'UCLouvain peuvent mettre à disposition des collections composées de centaines voir de milliers de descriptions d'archives, ce qui implique la gestion d'autant de Manifestes possibles

pour notre catalogue Mirador. C'est pourquoi plusieurs expérimentations, variant le nombre de Manifestes, ont été réalisées afin d'évaluer l'impact sur notre intégration.

Tout d'abord, des tests ont été réalisés au niveau du chargement de la visionneuse lors de l'affichage d'une description. C'est une façon simple de mesurer l'impact d'un grand catalogue sur l'initialisation de Mirador. Les résultats ont été analysés en effectuant des moyennes sur plusieurs exécutions pour différentes tailles de catalogue, voici le graphe illustrant cela :

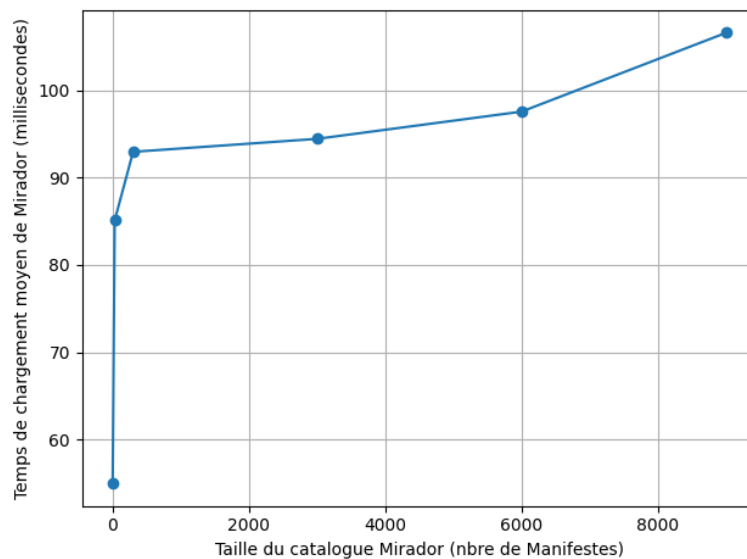


Figure 6.5: Impact de la taille du catalogue sur le temps de chargement de Mirador

Dans un premier temps, on peut ici constater que le temps de chargement de Mirador augmente de manière significative avec l'ajout du catalogue. Cependant, cette croissance se stabilise rapidement et ne varie pas énormément que l'on soit confronté à un catalogue avec cent, mille ou dix-mille Manifestes. Cela fait sens, car lors du chargement initial de la ressource liée à la description, seuls les URI des Manifestes sont chargés initialement, ce qui n'est pas très gourmand en terme de ressources. En effet, le chargement des vignettes et des informations de base des Manifestes du catalogue (cf. Figure 6.3) ne survient que lorsque le bouton pour accéder au catalogue est activé. C'est ici que la prise en compte d'un nombre élevé de Manifestes devient problématique. Lors de la tentative de chargement de la liste, l'interface peut commencer à laguer et les ressources système peuvent rapidement être dépassées, entraînant des erreurs en console. Un cas de figure comme celui-ci n'est absolument pas souhaitable pour garantir une bonne expérience utilisateur. C'est après ce constat que la décision de limiter le nombre d'éléments

dans le catalogue Mirador a été prise. Il a été fixé à un maximum de cinquante Manifestes, ce qui permet de garantir un temps de chargement raisonnable ainsi qu'une navigation aisée dans la liste, tout en évitant à l'utilisateur de se sentir submergé par une quantité excessive de Manifestes. De plus, il est important de noter que les Manifestes sélectionnés correspondent aux descriptions archivistiques les plus proches de la ressource actuelle au niveau de l'arborescence d'AtoM, assurant ainsi une pertinence et une facilité d'accès pour l'utilisateur.

# Chapitre 7

## Conclusion

À travers les différents chapitres de ce mémoire, nous avons examiné les divers aspects relatifs à la gestion d'archives et de documents numériques au sein du logiciel AtoM ainsi que les standards et fonctionnalités proposés par le framework IIIF. L'analyse approfondie de ces technologies, à la fois du point de vue technique et fonctionnel a permis d'aborder efficacement la phase de développement découlant des besoins liés à notre problématique. Au cours de cette étude, nous avons également pu constater les avantages significatifs relatifs aux innovations IIIF ainsi que les intérêts liés à l'intégration de ces normes dans l'environnement d'AtoM.

Le résultat concret obtenu suite à la phase d'implémentation de la nouvelle visionneuse correspond aux exigences définies initialement dans l'objectif et les besoins du projet. Cependant, l'intégration actuelle ne représente pas pour autant une version figée. Il est essentiel de considérer ce travail comme une base solide et évolutive, ouverte à d'éventuelles perspectives d'améliorations futures. En effet, l'implémentation open source de Mirador dans AtoM a été structurée selon des principes de modularité et de flexibilité, facilitant ainsi les possibles futures personnalisations et extensions.

J'espère que ce mémoire aura été éclairant et utile, car il aura été, pour moi, très enrichissant. Les normes IIIF, souvent méconnues du grand public, proposent des innovations qui améliorent grandement l'expérience utilisateur ainsi que la manipulation et le partage d'images en ligne. Travailler sur ce cas d'étude m'a permis de mieux comprendre l'importance et l'enjeu de ces technologies dans la préservation et l'accès aux ressources numériques. Je suis convaincu que la démocratisation et l'adoption croissantes de celles-ci renforceront positivement le paysage archivistique.

# Bibliographie

- [1] Archives de l'État en Belgique - Numériser de l'information, <https://www.arch.be/index.php?l=fr&m=fonctionnaire&r=numeriser-de-l-information>
- [2] Archives de l'UCLouvain - Catalogue AtoM, <https://archives.uclouvain.be/atom/index.php/>
- [3] Archives de l'OTAN - Catalogue AtoM, <https://archives.nato.int/>
- [4] AtoM, <https://www.accesstomemory.org/fr/>
- [5] AtoM - Technical overview, <https://www.accesstomemory.org/en/docs/2.4/user-manual/overview/intro/#technical-overview>
- [6] AtoM - Default user permissions settings in AtoM, <https://www.accesstomemory.org/en/docs/2.4/user-manual/administer/edit-permissions/#default-user-permissions-settings-in-atom>
- [7] AtoM Online Demo, <https://demo.accesstomemory.org/>
- [8] Conseil international des archives - ISAD(G): Norme générale et internationale de description archivistique (deuxième édition), 2000, <https://www.ica.org/resource/isadg-general-international-standard-archival-description-second-edition/>
- [9] Régis Robineau - Les protocoles IIIF, 23/01/2019, <https://projet.bibliissima.fr/fr/introduction-protocoles-iiif-formation-enssib-2019>
- [10] Julien Antoine Raemy, René Schneider - Suggested measures for deploying IIIF in Swiss cultural heritage institutions, 29/05/2019, [https://www.researchgate.net/publication/333566762\\_Suggested\\_measures\\_for\\_deploying\\_IIIF\\_in\\_Swiss\\_cultural\\_heritage\\_institutions](https://www.researchgate.net/publication/333566762_Suggested_measures_for_deploying_IIIF_in_Swiss_cultural_heritage_institutions)
- [11] Youtube - IIIF, <https://www.youtube.com/@IIIF-Consortium/videos>
- [12] IIIF | International Image Interoperability Framework, <https://iiif.io/>

- [13] Stuart Snyderman, Robert Sanderson, Tom Cramer - The International Image Interoperability Framework (IIIF): A community & technology approach for web-based images, mai 2015, [https://stacks.stanford.edu/file/druid:df650pk4327/2015ARCHIVING\\_IIIF.pdf](https://stacks.stanford.edu/file/druid:df650pk4327/2015ARCHIVING_IIIF.pdf)
- [14] UCL Archives - Fonds : Archives de Georges Lemaître, <https://archives.uclouvain.be/items/browse?collection=43>
- [15] UCL Archives - Fonds : Archives de l'expédition Philby-Ryckmans-Lippens en Arabie (1951-1952), <https://archives.uclouvain.be/items/browse?collection=63>
- [16] Julien Antoine Raemy - IIIF Pre-conference - Usability testing conducted on the UV and Mirador, 05/06/2017, <https://fr.slideshare.net/slideshow/iiif-preconference-usability-testing-conducted-on-the-uv-and-mirador/76798452>
- [17] FranceArchives - IIIF, un outil pour visualiser les archives numérisées sur FranceArchives, 02/04/2024, <https://francearchives.gouv.fr/fr/article/714850036>
- [18] Harvard Library - IIIF Manifests & Digital Objects, <https://library.harvard.edu/services-tools/iiif-manifests-digital-objects>
- [19] Forum de discussion des utilisateurs d'AtoM - Request to develop IIIF viewer embed in archival descriptions, mars 2023, <https://groups.google.com/g/ica-atom-users/c/5cT1tM6LnDw/m/4NZU-fySEwAJ>
- [20] AtoM - Installation, <https://www.accesstomemory.org/fr/docs/2.4/admin-manual/installation/>
- [21] GitHub - ProjectMirador/mirador, Paramètres de configuration de Mirador, <https://github.com/ProjectMirador/mirador/blob/master/src/config/settings.js>
- [22] AtoM - List of plugins, <https://www.accesstomemory.org/en/docs/2.8/user-manual/administer/manage-plugins/#list-of-plugins>

# Annexes

## A) Code source

Dépôt Github du projet: <https://github.com/CarlierLouis/atom-2.4.0-mirador3>

## B) Fichier de métadonnées d'image "info.json"

```
{
  "@context": "http://iiif.io/api/image/2/context.json",
  "@id": "https://archives.uclouvain.be/iiif-img/575093",
  "protocol": "http://iiif.io/api/image",
  "width": 987,
  "height": 660,
  "sizes": [
    {
      "width": 200,
      "height": 134
    },
    {
      "width": 800,
      "height": 535
    },
    {
      "width": 987,
      "height": 660
    }
  ],
  "profile": [
    "http://iiif.io/api/image/2/level2.json",
    []
  ]
}
```

## C.1) Exemple de Manifeste IIIF

Lien vers le Manifeste : <https://archives.uclouvain.be/iiif/147253/manifest?manifest=https://archives.uclouvain.be/iiif/147253/manifest>

## C.2) Utilisation des APIs Image et Presentation

**Séquence**

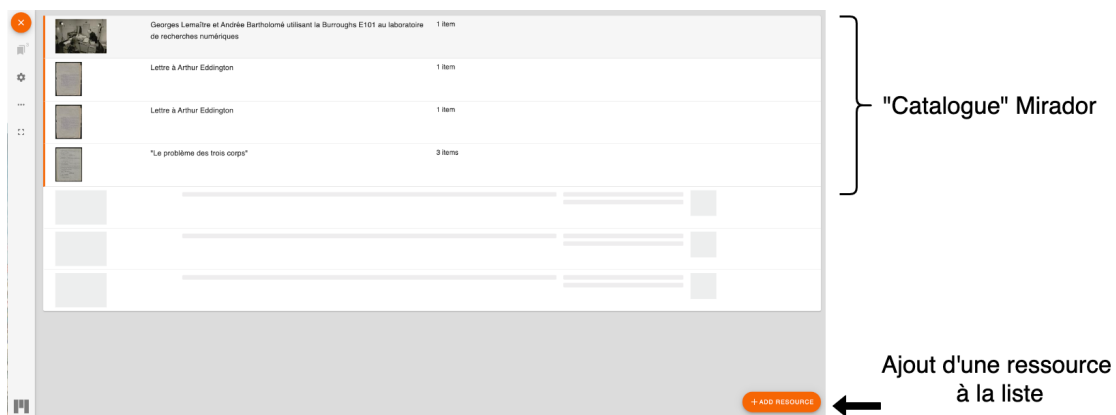
**Métadonnées**

**Image API**  
(Requêtes d'image, Zoom, etc.)

**Presentation API**

L'interface utilisée ici pour disposer la ressource IIIF est la visionneuse Universal Viewer. En fonction de l'outil utilisé, plus ou moins d'informations et de données sur le document numérique peuvent être représentées sur l'interface.

## D) Représentations de la visionneuse Mirador 3



## Comparaison de documents numériques



The screenshot displays a digital interface for document comparison, organized into three main panels:

- Top Left Panel:** A photograph showing two individuals, Georges Lemaître and André Bartholomé, working in a laboratory setting with scientific equipment, including a large machine and a desk.
- Bottom Left Panel:** A document titled "Le problème des trois corps" (The three-body problem), showing two pages of dense, handwritten mathematical text.
- Right Panel:** A scanned document titled "Lettre à Arthur Eddington" (Letter to Arthur Eddington). The text is typed and includes the following content:

Journal, Ch. Rev. 1923  
40, rue de Valenciennes

Dear Mr. Eddington,

I am glad to tell you that I have been appointed lecturer at the University of Louvain. The sector assigned to your intervention through M. de Doncker in his introduction address was I am sure to me very much appreciated. I had no opportunity to be made more sure.

I have left a sign and work I had begun on the occasion as I cannot find any answer to the objection you had made when in his way about the sign of the Poynting vector, but I have achieved the relation of the equation of the field within a square of uniform EVOLUTION reality. I got it done I could say at the Massachusetts Institute of Technology and I hope it will be printed in the Journal of Mathematics next October. I am sure you have a copy of the introduction where I try to explain your position in regard to this problem and a summary where I have outlined the final results.

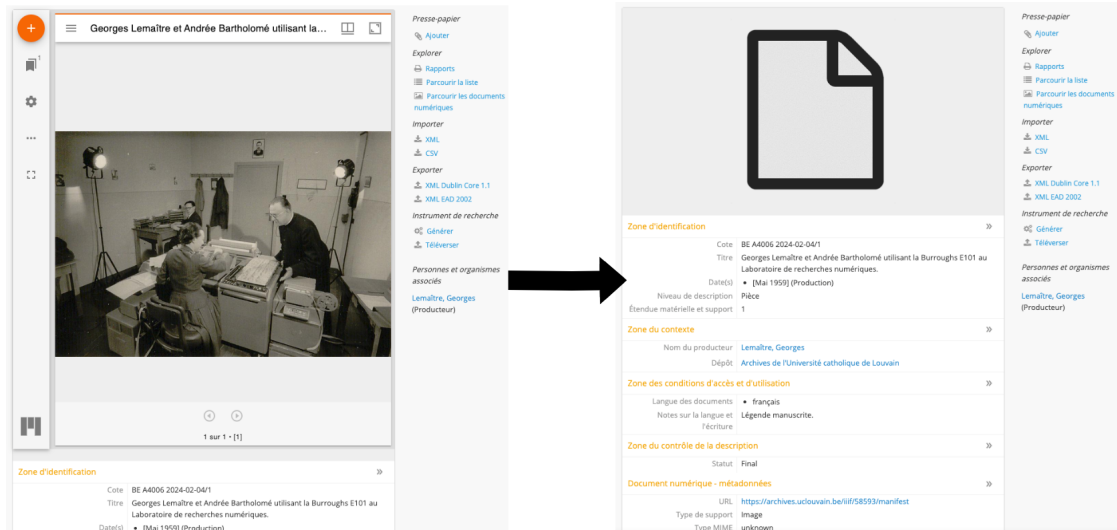
I would be very glad to know if I have correctly understood the statements you made in your book and what might be its value into the physical interpretation of the medium where the finite control pressure. I remember you were much puzzled with this result when I talk you a version.

I am sorry I could not get back to you a copy of the book on the other's universe but I am sure you in July. I am not really sorry.

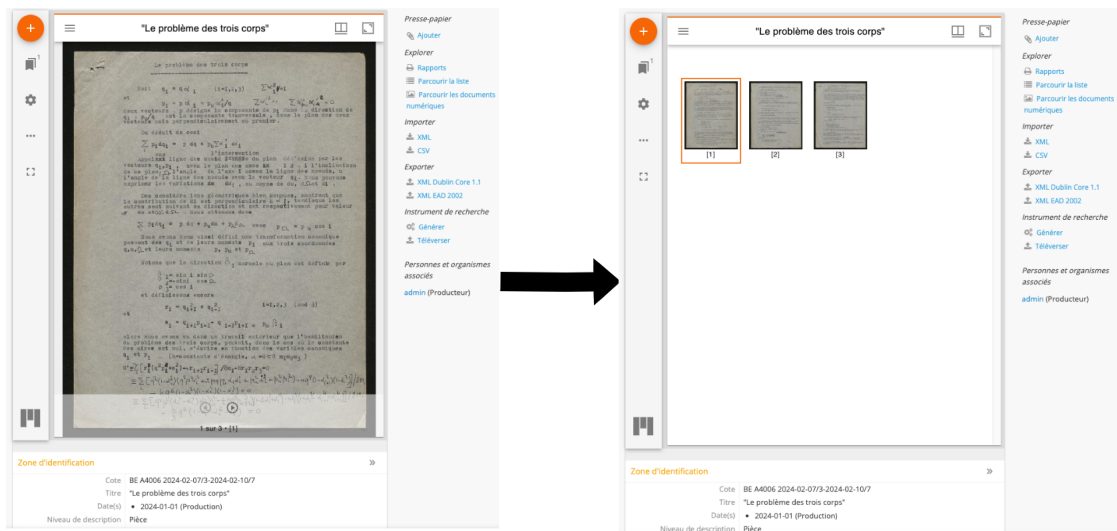
With kind regards I am

Yours sincerely

## E.1) Désactivation de Mirador au sein d'AtoM



## E.2) Comparaison entre la vue "simple" et la vue "galerie" de Mirador au sein d'AtoM



UNIVERSITÉ CATHOLIQUE DE LOUVAIN  
École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | [www.uclouvain.be/epl](http://www.uclouvain.be/epl)