



FAKULTA APLIKOVANÝCH VĚD  
ZÁPADOČESKÉ UNIVERZITY  
V PLZNI

KATEDRA INFORMATIKY  
A VÝPOČETNÍ TECHNIKY



**Diplomová práce**

# Interoperabilita obrazových dat pomocí frameworku IIIF

Viktorie Pavlíčková





FAKULTA APLIKOVANÝCH VĚD  
ZÁPADOČESKÉ UNIVERZITY  
V PLZNI

KATEDRA INFORMATIKY  
A VÝPOČETNÍ TECHNIKY

## **Diplomová práce**

# **Interoperabilita obrazových dat pomocí frameworku IIF**

Bc. Viktorie Pavlíčková

### **Vedoucí práce**

Doc. Dr. Ing. Jana Klečková

© Viktorie Pavlíčková, 2024.

Všechna práva vyhrazena. Žádná část tohoto dokumentu nesmí být reprodukována ani rozšiřována jakoukoli formou, elektronicky či mechanicky, fotokopírováním, nahráváním nebo jiným způsobem, nebo uložena v systému pro ukládání a vyhledávání informací bez písemného souhlasu držitelů autorských práv.

**Citace v seznamu literatury:**

PAVLÍČKOVÁ, Viktorie. *Interoperabilita obrazových dat pomocí frameworku IIIF*. Plzeň, 2024. Diplomová práce. Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Katedra informatiky a výpočetní techniky. Vedoucí práce Doc. Dr. Ing. Jana Klečková.

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd  
Akademický rok: 2023/2024

# ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Viktorie PAVLÍČKOVÁ**  
Osobní číslo: **A22N0001P**  
Studijní program: **N0613A140040 Softwarové a informační systémy**  
Téma práce: **Interoperabilita obrazových dat pomocí frameworku IIIF**  
Zadávací katedra: **Katedra informatiky a výpočetní techniky**

## Zásady pro vypracování

1. Seznamte se s frameworkem IIIF.
2. Zjistěte aktuální stav využití v praxi.
3. Proveďte analýzu možností využití pro Inventarium.
4. Vytvořte návrh pro vybrané případy.
5. Proveďte praktické ověření návrhu.
6. Zhodnoďte výsledky.

Rozsah diplomové práce: **doporuč. 50 s. původního textu**  
Rozsah grafických prací: **dle potřeby**  
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

dodá vedoucí diplomové práce

Vedoucí diplomové práce: **Doc. Dr. Ing. Jana Klečková**  
Katedra informatiky a výpočetní techniky

Datum zadání diplomové práce: **8. září 2023**  
Termín odevzdání diplomové práce: **16. května 2024**

L.S.

---

**Doc. Ing. Miloš Železný, Ph.D.**  
děkan

---

**Doc. Ing. Přemysl Brada, MSc., Ph.D.**  
vedoucí katedry

V Plzni dne 11. října 2023

# Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného akademického titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Západočeská univerzita v Plzni má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Plzni dne 14. června 2024

.....  
Viktorie Pavlíčková

V textu jsou použity názvy produktů, technologií, služeb, aplikací, společností apod., které mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

## Abstrakt

Tato diplomová práce prozkoumává International Image Interoperability Framework (IIIF), který slouží ke zlepšení správy a přístupnosti digitálních obrazových dat. Hlavním cílem práce je zmapování stávajícího využití IIIF v akademickém prostředí a analýza jeho potenciálního přínosu pro projekt Inventaria Rudolphina, který se zabývá digitalizací a správou obrazových inventářů Rudolfa II. Práce zahrnuje detailní analýzu možností využití tohoto frameworku, návrh konkrétních řešení pro zlepšení interoperability a dostupnosti digitálních obrazových sbírek, implementaci vybraných návrhů a jejich testování. Výsledky práce potvrzují, že IIIF může významně přispět k rozšíření možností výzkumu a vzdělávání v oblasti kulturního dědictví.

## Abstract

This thesis explores the International Image Interoperability Framework (IIIF), which is used to improve the management and accessibility of digital image data. The main objective of the thesis is to map the current use of the IIIF in an academic setting and to analyse its potential contribution to the Inventaria Rudolphina project, which deals with the digitisation and management of the image inventories of Rudolph II. The work includes a detailed analysis of the possibilities of using this framework, the proposal of specific solutions to improve the interoperability and accessibility of digital image collections, the implementation of selected proposals and their testing. The results of the work confirm that the IIIF can make a significant contribution to expanding the possibilities for research and education in the field of cultural heritage.

## Klíčová slova

IIIF • Interoperabilita • digitální obrazová data • Inventaria Rudolphina • manifest

## Poděkování

Ráda bych poděkovala Doc. Dr. Ing. Janě Klečkové za její podporu, rady a trpělivost, se kterou se mnou vedla konzultace. Dále bych chtěla vyjádřit vděčnost své rodině, přátelům a partnerovi za jejich neustálou podporu během mého studia a při psaní této práce.



# Obsah

<b>1</b>	<b>Úvod</b>	<b>5</b>
1.1	Struktura práce . . . . .	5
<b>2</b>	<b>Inventaria Rudolphina</b>	<b>7</b>
2.1	Rudolf II. . . . .	7
2.2	Struktura Inventaria Rudolphina . . . . .	9
2.3	Detail položky . . . . .	9
2.4	Technologické řešení Inventaria Rudolphina . . . . .	13
<b>3</b>	<b>IIIF framework</b>	<b>16</b>
3.1	Počátky a vize IIIF . . . . .	17
3.1.1	Konsorcium (IIIF-C) . . . . .	17
3.1.2	Motivace . . . . .	18
3.2	API . . . . .	18
3.3	Image API . . . . .	20
3.3.1	Image Requests . . . . .	21
3.3.1.1	Region . . . . .	21
3.3.1.2	Size . . . . .	21
3.3.1.3	Rotation . . . . .	23
3.3.1.4	Quality . . . . .	24
3.3.1.5	Format . . . . .	25
3.3.1.6	Dodatečné informace k implementaci . . . . .	26
3.3.2	Image Information . . . . .	26
3.3.2.1	Technické vlastnosti . . . . .	27
3.3.2.2	Sizes . . . . .	28
3.3.2.3	Tiles . . . . .	28
3.3.2.4	Preferred Formats . . . . .	28
3.3.2.5	Práva . . . . .	29
3.3.2.6	Linking Properties . . . . .	29
3.3.3	Server responses . . . . .	29

3.3.4	Zabezpečení . . . . .	29
3.4	Presentation API . . . . .	30
3.4.1	Resource Types . . . . .	30
3.4.1.1	Collection . . . . .	31
3.4.1.2	Manifest . . . . .	32
3.4.1.3	Canvas . . . . .	32
3.4.1.4	Range . . . . .	33
3.4.1.5	Annotation Page . . . . .	34
3.4.1.6	Annotation . . . . .	34
3.4.1.7	Content Resources . . . . .	34
3.4.1.8	Annotation Collection . . . . .	35
3.4.2	Resource Properties . . . . .	35
3.4.2.1	Descriptive Properties . . . . .	35
3.4.2.2	Technical Properties . . . . .	37
3.4.2.3	Linking Properties . . . . .	38
3.4.2.4	Structural Properties . . . . .	40
3.5	IIIF image viewers . . . . .	40
3.5.1	Mirador . . . . .	41
3.5.1.1	Výhody . . . . .	41
3.5.1.2	Nevýhody . . . . .	41
3.5.2	Universal Viewer . . . . .	41
3.5.2.1	Výhody . . . . .	42
3.5.2.2	Nevýhody . . . . .	42
3.5.3	OpenSeadragon . . . . .	42
3.5.3.1	Výhody . . . . .	42
3.5.3.2	Nevýhody . . . . .	42
3.5.4	Tify . . . . .	43
3.5.4.1	Výhody . . . . .	43
3.5.4.2	Nevýhody . . . . .	43
3.5.5	IIIF Curation Viewer . . . . .	43
3.5.5.1	Výhody . . . . .	43
3.5.5.2	Nevýhody . . . . .	43
3.5.6	Srovnání . . . . .	44
<b>4</b>	<b>Použití IIIF Frameworku</b>	<b>45</b>
4.1	Aktuální stav využití v praxi . . . . .	45
4.1.1	Aplikace technologie a její cíle . . . . .	46
4.1.2	Spolupráce a otevřený přístup . . . . .	46
4.1.3	Geografie . . . . .	47
4.2	Výzvy a omezení v aplikaci frameworku . . . . .	47

4.3	Analýza oblastí využití IIIF u obrazů Rudolfa II. . . . .	48
<b>5</b>	<b>Analýza možností využití pro Inventarii</b>	<b>50</b>
5.1	Využití IIIF manifestů . . . . .	50
5.2	Poskytování obrazů na IIIF kompatibilním serveru . . . . .	51
5.3	Integrace Image vieweru do Inventarie . . . . .	52
<b>6</b>	<b>Návrh pro vybrané případy</b>	<b>53</b>
6.1	Vytvoření IIIF manifestů pro díla . . . . .	53
6.1.1	Motivace návrhu . . . . .	53
6.1.2	Omezení stávajícího řešení . . . . .	54
6.1.3	Implementace . . . . .	55
6.1.3.1	Výběr technologie pro extrakci dat z databáze . . . . .	55
6.1.3.2	Teoretický postup extrakce dat . . . . .	56
6.1.3.3	Návrh transformace dat do IIIF manifestů . . . . .	56
6.1.3.4	Návrh validace vytvořených IIIF manifestů . . . . .	57
6.2	Integrace IIIF do aplikačního serveru . . . . .	58
6.2.1	Motivace návrhu . . . . .	58
6.2.2	Omezení stávajícího řešení . . . . .	59
6.2.3	Update aktuálního serveru o podporu IIIF . . . . .	59
6.2.3.1	Návrh implementace Image API . . . . .	59
6.2.3.2	Návrh testování . . . . .	60
6.3	Návrh integrace Image Vieweru do Inventaria . . . . .	61
6.3.1	Motivace návrhu . . . . .	61
6.3.2	Omezení stávajícího řešení . . . . .	62
6.3.3	Výběr image vieweru . . . . .	62
6.3.4	Stažení balíčku Mirador . . . . .	63
6.3.5	Vytvoření komponenty . . . . .	63
6.3.6	Dynamické zpracování a aktualizace manifestu . . . . .	63
6.3.7	Návrh testování . . . . .	64
<b>7</b>	<b>Praktické ověření návrhu</b>	<b>65</b>
7.1	Implementace . . . . .	65
7.1.1	Komunikace s databází a uložení . . . . .	65
7.1.2	Transformace JSON souborů do IIIF manifestu . . . . .	66
7.2	Testování . . . . .	67
7.2.1	Validace manifestu . . . . .	67
7.2.2	Testovací scénáře . . . . .	68
7.2.2.1	Příprava prostředí . . . . .	68
7.2.2.2	Scénář pro získání vstupu . . . . .	68

7.2.2.3	Scénář pro provedení transformace . . . . .	69
7.2.2.4	Scénář pro ověření validace výstupu . . . . .	69
7.3	Evaluaace výsledku implementace . . . . .	70
7.3.1	Ukázka vytvořeného manifestu . . . . .	70
7.3.1.1	Ukázka výsledného obrázku v prohlížeči . . . . .	72
<b>8</b>	<b>Náměty na budoucí rozšíření</b>	<b>73</b>
8.1	Rozšíření manifestu . . . . .	73
8.2	Integrace poznámek . . . . .	73
8.3	IIIF kolekce jako inventáře . . . . .	75
<b>9</b>	<b>Zhodnocení výsledků</b>	<b>77</b>
9.1	Vylepšení zavedením IIIF manifestů . . . . .	77
9.1.1	Nedostatky stávajícího řešení . . . . .	77
9.1.2	Výhody nového řešení . . . . .	78
9.1.3	Možná omezení nového řešení . . . . .	78
9.2	Celkové zhodnocení . . . . .	78
<b>10</b>	<b>Závěr</b>	<b>80</b>
<b>A</b>	<b>Uživatelská dokumentace</b>	<b>81</b>
A.1	Obsah balíčku . . . . .	81
A.2	Příprava prostředí . . . . .	81
A.2.1	Instalace Pythonu . . . . .	81
A.2.2	Instalace potřebných knihoven . . . . .	81
A.2.3	Spuštění skriptů . . . . .	82
A.2.3.1	Spuštění skriptu extract.py . . . . .	82
A.2.3.2	Spuštění skriptu transform.py . . . . .	82
<b>B</b>	<b>Ukázka struktury položky Inventarie</b>	<b>83</b>
<b>C</b>	<b>Příklad vytvořeného manifestu</b>	<b>87</b>
	<b>Seznam zkratk</b>	<b>89</b>
	<b>Bibliografie</b>	<b>90</b>
	<b>Seznam obrázků</b>	<b>93</b>
	<b>Seznam výpisů</b>	<b>94</b>

Tato diplomová práce se věnuje prozkoumání a aplikaci frameworku IIIF (International Image Interoperability Framework), který se zaměřuje na zlepšení dostupnosti a interoperability digitálních obrazových dat. Cílem je adaptovat tento framework pro potřeby projektu *Inventaria Rudolphina*, spravovaného Akademií věd, jenž se soustředí na digitalizaci a správu obrazových inventářů Rudolfa II.

Práce nejprve představí základní principy a technologie frameworku IIIF. Následně se zaměří na důkladnou analýzu jeho využití v rámci zmíněného projektu a na základě této analýzy navrhne specifická zlepšení v systému správy obrazových dat. Tento návrh bude nejen teoreticky rozpracován, ale také prakticky implementován a otestován, aby bylo možné přímo ověřit jeho přínosy a funkcionalitu.

Cílem práce je provést důkladnou analýzu možností, které IIIF poskytuje, a také ověřit, jak mohou být využity v akademickém prostředí.

## 1.1 Struktura práce

První část práce se seznámí s historickou postavou Rudolfa II., který je známý svou vášní pro umění a rozsáhlými sbírkami uměleckých děl. Toto historické pozadí poskytuje základ pro projekt *Inventaria Rudolphina*, který se věnuje digitalizaci a správě obrazů právě z období Rudolfa II. a jeho dvora.

Následně práce představí framework IIIF (International Image Interoperability Framework), který umožňuje institucím efektivněji spravovat a sdílet digitální obrazová data. Vysvětlí motivaci za vytvořením tohoto frameworku, jeho hlavní cíle a přednosti. Detailně popíše hlavní API, které IIIF nabízí, včetně jejich funkcí. Také se zaměří na několik specifických image viewerů (prohlížečů obrázků), které jsou součástí ekosystému IIIF.

Další část práce se věnuje mapování oblastí, kde je framework IIIF využíván. Prozkoumá různé sektory od akademických institucí přes veřejné knihovny, kde IIIF pomáhá zpřístupňovat a propojovat obrazové zdroje. Analyzuje, jak se IIIF uplatňuje v praxi a jaké přínosy přináší konkrétním uživatelům.

Následuje analytická část, kde práce provede analýzu možností, jak implementovat IIF do projektu *Inventaria Rudolphina*. Zhodnotí technické, organizační a uživatelské aspekty takové integrace a navrhne optimální postup pro využití IIF v rámci tohoto projektu.

V poslední části se práce zaměří na praktickou aplikaci vybraného návrhu. Popíše proces návrhu, implementaci a následné ověření vybraného řešení. Toto ověření umožní posoudit úspěšnost implementace a její dopady na správu obrazových dat v Inventarii.

# Inventaria Rudolphina

## 2

Centrum Studia Rudolphina, které je součástí Ústavu dějin umění v rámci Akademie Věd v Praze, se specializuje na vědecký výzkum spojený s osobností Rudolfa II. jako vášnivého sběratele. Jeho hlavním zaměřením je analýza umění a kultury spojených s císařskými sbírkami a dvorními umělci z období Rudolfovy vlády. Centrum má za úkol podporovat výzkumné aktivity v této oblasti mezi odborníky a zároveň poskytovat podporu českým i zahraničním studentům [1].

Tým centra Studia Rudolphina v současné době pracuje na virtuální databázi, která uchovává informace o umělecké sbírce Rudolfa II. kolem roku 1600. Tento projekt nese název *Art for display: The Painting Collection of Emperor Rudolf II within the Context of Collecting Practices circa 1600* [1].

## 2.1 Rudolf II.

Rudolf II. Habsburský, narozen 18. července 1552 ve vídeňském paláci Hofburg, byl prvorozeným synem Maxmiliána II. Habsburského a Marie Španělské. Jako nejstarší z 16 dětí byl roku 1563 na příkaz své matky poslán spolu se svým mladším bratrem Arnoštem do Španělska za výchovou a vzděláním. Poté se jako císař usadil trvale v Praze na Pražském Hradě v roce 1583, doprovázen dvorskými umělci, kteří již pracovali pro jeho otce Maxmiliána II. ve Vídni. Jeho záliba v umění byla zděděna po předcích, zejména po otci a strýci, španělském králi Filipu II. Rudolf II. byl patronem umění a sběratelem, ovlivněný italskými mistry, a inspiroval se také vídeňskou větví Habsburků [2] [3].

Umělci na dvoře Rudolfa II. byli zkušení mistři z různých uměleckých center Evropy, což na začátku vytvářelo stylistickou různorodost. Termín "rudolfínský styl" se začal používat až na přelomu 16. a 17. století, kdy umělci dlouhodobě spolupracovali a docházelo k vzájemnému ovlivňování [4] [3].

Mezi umělce patřili Giulio Licinio, Giuseppe Arcimboldo, Martino Rota, a Bartholomeus Spranger. Licinio se věnoval návrhům dekorací a portrétní malbě. Arcimboldo zhotovoval portréty a byl uměleckým agentem. Rota se specializoval na por-

tréty a pracoval do své smrti v roce 1583. Spranger, původně krajinář, se později zaměřil na mytologické obrazy a alegorie a pracoval pro Rudolfa až do své smrti v roce 1611. Dalšími umělci byli Hans Mont, sochař a architekt, a Hans Hoffmann, specialista na malbu přírodnin a kopista Dürerových děl. Jejich práce přispěly k vývoji uměleckého stylu na dvoře Rudolfa II. Větší proměny stylu došlo v 90. letech, kdy se Sprangerův styl začal ovlivňovat dalšími umělci na dvoře [4].

V roce 1590 došlo na Rudolfově dvoře k významnému přelomu, kdy starší generace umělců částečně vystřídala novou. Následující dekáda byla obdobím intenzivní umělecké aktivity, přičemž došlo k diferenciaci uměleckých námětů, zejména v oblasti malířství. Císař Rudolf II. upřednostňoval alegorická témata s erotickým nádechem a vykazoval zájem o žánrové malby. Vynikajícím příkladem byly zátiší s květinami od Roelanda Saveryho, který na dvoře působil od roku 1603. Savery, podobně jako další umělci, napomáhal rozvoji naturalismu v rudolfínském umění, zejména ve zobrazování přírody. Jeho kresby krajin kolem Prahy jsou cennými svědectvími dobového vzhledu města. Kromě Saveryho působili jako krajináři například Paulus van Vianen a Pieter Stevens, kteří se věnovali i žánrovým obrazům s vesničany při slavnostech, interpretovaným nyní spíše s moralistickým podtextem [4].

V oblasti malby portrétů vynikl Josef Heintz st., který do Prahy přišel v roce 1601. Po studijních letech v Itálii přinesl styl z Říma, kde byl ovlivněn pracemi Correggia a Federica Barroccioho. Jeho Faetonův pád a Leda s labutí patřily k významným obrazům, které namaloval pro císaře Rudolfa II. Hans von Aachen, třetí významný malíř na dvoře, proslul jako vynikající portrétista a alegorický malíř. v jeho díle vyniká kolorit silně ovlivněný benátskou malbou, a v pozdním období, inspirovaném italskými umělci jako Carracci, dosáhl vrcholu [4].

Rudolfova hluboká vzdělanost a významná úroveň jeho dvora, kde působili přední vědci a myslitelé, ovlivnily i uměleckou tvorbu. Typickými rysy jsou mytologické obrazy, které skrývají hluboké alegorické významy. Častými motivy jsou obnažené milenecké postavy s jasným erotickým podtextem, i když primární význam má často morální a filozofický charakter. Umělci využívali antické bohy a hrdiny k vyjádření abstraktních idejí a interpretaci historických událostí. Tento přístup, podporovaný renesančními příručkami a komentáři k Ovidiovi, byl v souladu s hermetismem a novoplatonismem, které byly v 16. století oblíbené. Tato filozofická a mystická témata byla v Praze za Rudolfovy vlády výrazně populární [4].

Rudolfínské umění též inovovalo malbu naturálií. v Praze se objevily počátky samostatného zátiší a krajinomalby, inspirované studii v přírodě. Nizozemské malířství mělo na rudolfínské umění podobný vliv s poklesem zájmu o náboženskou malbu a s pronikáním morálního prvku do jiných žánrů. Často se objevovaly náměty nerovných párů a galantní scény s naturalistickými prvky. v tomto období nebyly antické nebo renesanční předlohy pouze inspirací, ale spíše výzvou, s níž malíři



soutěžili a transformovali je směrem k modernímu pojetí [4].

Po Rudolfově smrti byly sbírky rozptýleny, některé darovány, některé ztraceny. Snahy o obnovení sbírek v následujících letech se střetly s nejistým osudem původních exponátů. Rudolfova sbírka, zahrnující tisíce uměleckých děl, soch a předmětů, zůstává záhadou v historii Pražského hradu [4].

## 2.2 Struktura Inventaria Rudolphina

Projekt *Inventaria Rudolphina* reprezentuje virtuální sbírku, která poskytuje textové i vizuální informace o uměleckých dílech, která byla součástí sbírky Rudolfa II. Výčet sběratelských pramenů jako jsou inventáře, dobová svědectví a identifikovaná díla, společně s heuristickým výzkumem a vytvořením specializované databáze, umožňuje systémovou analýzu rozsáhlé sbírky Rudolfa II. Zaměřuje se na obrazové žánry, malířské školy, vkus sběratele a též na prostorové uspořádání a správu sbírky. Studium sbírky se začlenilo do kontextu sběratelských aktivit Habsburků v Madridu a Bruselu, knížecích dvorů v Itálii (zejména ve Florencii a Mantově) a Německu (Drážďany, Mnichov). Rovněž zkoumá proměny statutu malby jako "galerijního exponátu" a současně analyzuje malířské inovace z období kolem roku 1600, zejména v Římě. Důraz je kladen i na specifické charakteristiky Rudolfovy galerie s ohledem na její roli v rezidenčním komplexu Pražského hradu.

*Inventaria Rudolphina* obsahuje soubor inventářů datovaných včetně roku a místa jejich sestavení (např. *1595 Brussels*). Každý inventář dokumentuje specifický seznam uměleckých děl, které tvořily součást tehdejší císařské sbírky. Každý artefakt v inventáři je dále podrobně prozkoumán (viz sekce 2.3). Dále pak *Inventaria* mimo jiné zahrnuje vizualizaci plánu Pražského hradu, konkrétně widget, který ilustruje stav hradu kolem roku 1621.

Závěrečným významným prvkem webového rozhraní *Inventaria* je sekce určená pro poznámky, sloužící k diskuzi a poskytující možnost komentářů ke specifickým artefaktům.

## 2.3 Detail položky

Charakteristiky individuálních artefaktů se různí v závislosti na dostupných informacích. Konkrétní detail položky může zahrnovat záložky, které nesou identifikátory inventáře a roku. Tato úroveň detailu je doplněna o prvek barvy, sloužící k označení propojení mezi jednotlivými artefakty, které jsou buďto identické nebo podobné. Tento prvek je označován termínem *certainty of concordance* a reflektuje úroveň, na níž autoři dosahují jistoty ohledně příslušnosti názvu a díla k sobě.

Modrá barva je využívána k označení shody mezi položkami, což značí identičnost děl. Dále je možné použít různé barvy pro vyjádření míry jistoty, viz obrázek 2.1.

Zelená barva reprezentuje vysokou pravděpodobnost propojení, oranžová signalizuje střední pravděpodobnost a červená indikuje nízkou pravděpodobnost.



Obrázek 2.1: Zobrazení záložek nad kartou detailu

Dále se v obsahu položky nachází karta s informacemi o artefaktu, viz obrázek 2.2. Mezi klíčové atributy obsažené v těchto podrobnostech patří identifikace autora, kde jméno autora slouží jako hypertextový odkaz na seznam všech jeho prací. v případě neznámého autora je uvedeno označení *Unknown*. Následující údaje se liší v závislosti na dostupnosti informací o daném díle.

### Pieter Bruegel I Christ Carrying the Cross

Obrázek 2.2: Jméno autora a název díla

Prvním informačním prvkem je *Inventory Item*, viz obrázek 2.3. Tento atribut slouží k poskytnutí základního informačního popisu daného artefaktu. Většinou zahrnuje identifikační číslo položky, název díla a jméno autora v jednotném celku.

### Inventory item: 16 Christi Creutzigung von Bruegel

Obrázek 2.3: Příklad zobrazení *Inventory item*

Následně může položka obsahovat informace o názvu pokoje (*room*) nebo konkrétním místě (*place*), kde se dané umělecké dílo nacházelo v období působení Rudolfa II., viz obrázek 2.4. Termín *room* typicky označuje název místnosti, kde se artefakt nacházel. *Place* poté více specifikuje polohu artefaktu nebo jeho blízkost k určitému prvku v dané místnosti, v překladu například "poblíž oken" či "na římse".

Room: *Folgen ferner die Gemähl, welche in den fördern gang sein.*  
Place: *oben auf dem gesims*

Obrázek 2.4: Příklad označení pokoje a místa

Dále se může volitelně nacházet poznámka k dílu, viz obrázek 2.5.

Dalším atributem může být hypertextový odkaz *IconClass*, především využívaný v galeriích či muzeích pro detailní popis a klasifikaci obrazů, viz obrázek 2.6.

Note:  
Fehlte schon bei der Revision von 1635 (KK 166, LL 166, A 166: unter 317 stucken ersetzt),  
ersetzt aus dem Zugängen von 1635 (AA)

### Obrázek 2.5: Příklad poznámky

Jedná se o klasifikační systém, který umožňuje systematické uspořádání objektů na díle, čímž vytváří možnost propojení jednotlivých obrazů a navazování vztahů mezi nimi. Tím je dosaženo možnosti odkazovat se na konkrétní díla či objekty na díle. v důsledku použití tohoto klasifikačního systému je každý obraz přiřazen identifikátor (ID), který reprezentuje jeho zařazení do různých tříd.

Iconclass 73D41

carrying of the cross: Christ bearing the cross, alone or with the help of others (e.g. Simon the Cyrenian)



### Obrázek 2.6: Ukázka IconClass

Následujícím atributem je *Subject*, který též souvisí s klíčovými slovy ohledně díla, viz obrázek 2.7. v tomto případě se jedná výhradně o klíčová slova, jež slouží k popisu charakteristik díla.

*Apuleius, Metamorphoses, Cupid, Golden Ass (Apuleius), Love, Metamorphoses (Apuleius), Psyche, ancient history, classical antiquity, goddess, gods, heaven, history, love-affair, lover, mythology, serving*

erotic, Amor

### Obrázek 2.7: Klíčová slova *Subject*

Dále může dílo obsahovat tzv. *Additional information*, neboli dodatečné informace, viz obrázek 2.8. Tato část může zahrnovat poznámky o původu díla, tj. zda se jedná o originál či kopii (s případným doplněním pravděpodobnosti), fyzické rozměry obrazu a odhadovanou cenu daného artefaktu.

Additional information:

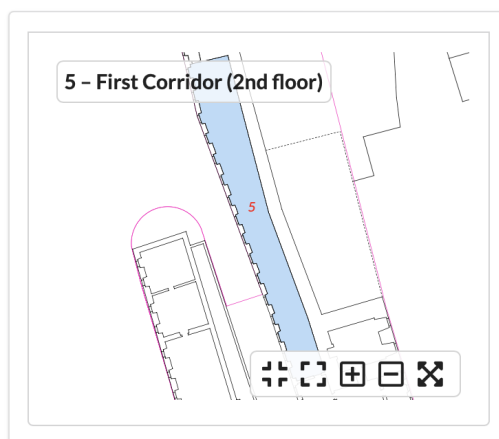
Source: original  
(certainty:high)

Size: large                      groß

### Obrázek 2.8: Dodatečné informace k dílu

V podrobnostech jsou dále zobrazeny další dvě karty. První z nich reprezentuje mapu hradu, na níž je znázorněno místo, kde bylo dílo vystaveno, viz obrázek 2.9.

Tato vizualizace prezentuje dispoziční uspořádání podlaží a místností, sloužících k výstavě Rudolfových uměleckých děl. Plán zahrnuje celkem tři podlaží, přičemž každé podlaží může být dále rozděleno na úrovně pokojů. Jednotlivé pokoje mohou obsahovat specifická místa, jako například "na římsě". Plán hradu poskytuje srozumitelný kontext pro lokalizaci jednotlivých exponátů.



Obrázek 2.9: Vizualizace plánu Pražského hradu

Druhá ze zmiňovaných karet obsahuje informace o autorovi daného díla, viz obrázek 2.10. Konkrétně se jedná o jméno autora, jeho místo původu, povolání a přesné nebo přibližné datum narození a úmrtí. Následně poskytuje seznam jmen, pseudonymů nebo přezdivek, pod kterými mohl autor vystupovat.



Obrázek 2.10: Příklad karty s informacemi o autorovi

Nicméně klíčovým aspektem této práce je vizuální dokumentace jednotlivých uměleckých děl. Každý detail díla může obsahovat jednu nebo více vizuálních reprezentací, případně žádnou, pokud není známa vizuální podoba. Detail díla může také zahrnovat obrazy podobných děl, jejich kopií nebo obrázky, které jsou jakýmkoli

způsobem spojeny s daným uměleckým dílem. Každý vizuální záznam je označen klíčovým termínem pro toto označení.

Mezi první klíčové označení patří *Identified artwork*, viz obrázek 2.11. Tato kategorie zahrnuje obrazy, které jsou přiřazeny k danému dílu. Přiřazení má také určitou míru *certainty of concordance*, obdobně jako podobnost mezi díly.



Obrázek 2.11: Ukázka obrazu s označením *Identified* a se střední mírou jistoty

Dále může obsahovat také kategorie *Resembling objects*, viz obrázek 2.12. To jsou obrazy, které pouze připomínají dané dílo, nikoli ho ovšem reprodukují. Například známe pozdější kopii původního uměleckého díla nebo máme kresbu, která vznikla před finální malbou. Potom se jedná právě o *Resembling object*.

Následně se můžeme setkat i s označením *Related Item*, viz obrázek 2.13. Jedná se rovněž o obraz, který je s daným dílem propojen pouze nějakým způsobem.

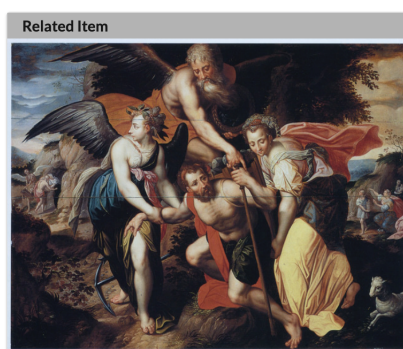
## 2.4 Technologické řešení Inventaria Rudolphina

Technologické řešení aplikace *Inventaria Rudolphina* zahrnuje MongoDB pro databázi, Node.js pro backend a Semantic UI společně s Vue.js pro frontend. Databáze je rozdělena do několika kolekcí. Jedná se o *artists*, *concordances*, *iconclass*, *inventory*, *notes*, *related*, *rooms*, *ulan* a *users*. Tyto kolekce představují relační tabulku a sdružují jednotlivé dokumenty, které jsou ve formátu JSON. Tato sekce se bude nadále věnovat JSON struktuře v detailu položky a jejímu rozboru. Ukázka této struktury položky je v příloze B.

Jednotlivé atributy, které JSON dokument může nabývat, jsou následující:



Obrázek 2.12: Ukázka obrazu s označením *Resembling*



Obrázek 2.13: Ukázka obrazu s označením *Related*

- **order**: pořadí v inventáři
- **xml\_id**: identifikace díla
- **subject**: klíčová slova
- **search\_subject**: klíčová slova pro využití ve vyhledávání
- **iconclass\_external\_id**: externí id pro IconClass
- **concordances**: seznam odkazů na související díla z jiných inventářů včetně míry jistoty
- **text**: popis díla
- **title**: název díla

- **object:**
  - **caption:** název díla
  - **type:** typ díla
  - **text** text
  - **digitalisation\_layer:** TRUE/FALSE hodnota; označení, zda bylo dílo digitalizováno
  - **name:** informace o autorovi; obsahuje jméno, roli a informace z Getty
  - **xml\_id:** id XML souboru
  - **relationship\_type:** vztah mezi dílem a jeho metadaty
  - **images:** odkaz na konkrétní obrázek ze serveru s popisem
  - **cert:** míra jistoty propojení obrázku s metadaty
  - **caption:** popis díla
  - **object\_name:** název díla
  - **id\_number:** id díla v odkazující instituci
  - **institution:** název instituce, ve které se fyzické dílo nachází
  - **country:** země instituce
  - **city:** město instituce
  - **latitude:** a **longitude:** zeměpisná šířka a délka instituce
  - **physical\_description:** další informace týkající se díla v instituci
  - **material:** informace o technice malby
  - **dimensions:** rozměry fyzického díla
  - **origDate:** datum vytvoření díla
  - **provenance:** provenience díla
- **inventory:** název a pořadí inventáře obsahující dílo
- **room:** textový popis pokoje, kde se dílo nacházelo na Pražském hradě za doby Rudolfa II.
- **place:** textový popis konkrétního místa (zeď, roh,...), kde se dílo nacházelo
- **prevItem:** odkaz na předchozí položku v inventáři
- **nextItem:** odkaz na následující položku v inventáři

# IIIF framework

## 3

Zkratka IIIF (obecně vyslovována jako *tripl áj ef*) znamená *International Image Interoperability Framework*, což je v překladu Mezinárodní framework pro obrázkovou interoperabilitu [5].

Následuje detailní vysvětlení těchto pojmů:

- Termín *mezinárodní* znamená, že tento framework není omezen implementací jedné země, nýbrž má mezinárodní dosah a zabírá celý svět, zahrnující kupříkladu Evropu, Ameriku a Asii [6].
- na počátku své existence se skupina IIIF věnovala převážně zpřístupňování velkých obrázků na internetu, což vysvětluje použití slova *Image* v názvu.
- Dále termín *interoperabilita* vyjadřuje schopnost systémů vzájemně komunikovat, spolupracovat a poskytovat si služby. Cílem interoperability je zajištění snadné a bezproblémové integrace mezi různými technologiemi a systémy. IIIF zajišťuje interoperabilitu tím, že umožňuje opětovné spojení vzdáleně distribuovaných obrázků v nových digitálních edicích. Dále poskytuje standardní způsob vystavení obrázků a metadat, což umožňuje přepínat prohlížeče (označované také jako *viewers*) pro zobrazení totožného obsahu [7].
- *Framework* představuje komplexní soubor nástrojů, jako jsou knihovny, šablony nebo moduly s předpřipraveným kódem. Tyto nástroje usnadňují vývoj aplikací a softwaru, poskytují vývojářům strukturovaný základ, který umožňuje efektivnější práci bez opakovaného programování a implementace běžných prvků. IIIF zajišťuje, že *viewers* a *publishers* dosáhnou shody na sdílení kompatibilních dat. Tyto standardy jsou implementovány prostřednictvím aplikačních rozhraní (API), z nichž dva hlavními jsou Image API a Presentation API [5].



## 3.1 Počátky a vize IIF

Přístup k obrazovým zdrojům představuje základní pilíř pro výzkum, vědeckou práci a přenos kulturních poznatků. Dosud byly mnohé obrazové zdroje na internetu izolovány v uzavřených úložištích, k nimž byl přístup pouze přes specifické aplikace. IIF vytváří jednotný standard pro zobrazení knih, map, svitků, rukopisů, hudebních partitur a archivního materiálu ze zapojených institucí. Tento framework umožňuje badatelům a studentům z celého světa individuálně i ve skupinách zobrazovat, manipulovat, měřit a anotovat tyto obrazy [8].

### 3.1.1 Konsorcium (IIF-C)

Zrod IIF iniciativy, vzniklé při večeri technologů z knihoven Bodleian na Oxfordské univerzitě, Britské knihovny a knihoven Stanfordovy univerzity v minulé dekádě, oficiálně vyústil ve vytvoření Mezinárodního konsorcia pro obrázkovou interoperabilitu (IIF-C). Cílem konsorcia je eliminovat neefektivnost a redundanci plynoucí z nekompatibility v současném systému doručování obrazů. Paralelně s tím se snaží definovat společná aplikační programová rozhraní (API) a aktivně vyvíjet a dokumentovat sdílené technologie, jako jsou obrazové servery a webovní klienti. Tyto technologické aspekty jsou koncipovány tak, aby poskytovaly uživatelsky významný zážitek při prohlížení, porovnávání, manipulaci a anotaci obrazů [8].

Vzniklé Konsorcium IIF (IIF-C) představuje spojení vědeckých knihoven, národních knihoven, archivů, softwarových společností, digitálních agentur a agregátorů s hlavním záměrem poskytovat bezprecedentní jednotný a bohatý přístup k digitalizovaným obrazovým, zvukovým a video zdrojům na celosvětové úrovni. Založeno bylo v roce 2015 iniciátory, mezi nimiž byly:

- University of Oxford
- British Library
- Stanford University
- Artstor (nyní ITHAKA)
- Die Bayerische Staatsbibliothek
- Cornell University
- La Bibliothèque Nationale de France
- Nasjonalbiblioteket
- Princeton University

- Wellcome Trust
- Yale University

S aktuální účastí 67 členů z celého světa, kteří podpořili *IIIF-C Memorandum of Understanding*, se konsorcium zavázalo aktivně podporovat růst a implementaci IIIF. Tím přispívá k rozvoji a šíření tohoto frameworku pro obrázkovou interoperabilitu [9].

## 3.1.2 Motivace

V rámci výzkumu na této práci byla kontaktována IIIF skupina za účelem zjistit další cíle. Bylo zjištěno, že další motivací za vytvoření IIIF frameworku bylo několik klíčových faktorů. Za prvé, v reakci na rostoucí význam digitálních obrazů na internetu jakožto hlavního prostředku pro sdílení informací v oblasti kulturního dědictví, bylo třeba vyvinout standardizovaný přístup, který by podpořil jejich efektivní využívání a opakované použití.

Dále zde byla potřeba zefektivnit proces dodávání obrázků, aby byl rychlejší, jednodušší a levnější, což by umožnilo institucím snadněji sdílet své sbírky a široké veřejnosti poskytnout přístup k bohatství kulturního dědictví.

Třetí motivací byla snaha odstranit překážky mezi institucemi a jejich servery a systémy pro doručování obrazů, a tím usnadnit interoperabilitu a sdílení digitálních zdrojů.

A v neposlední řadě, vytvoření IIIF frameworku mělo za cíl umožnit implementaci nových funkcí pro uživatele, jako je hloubkové zvětšení (také označováno jako *deep zoom*), sdílení částí obrázků a ořezávání, čímž by se zlepšila uživatelská přívětivost (UX) a otevřeli nové možnosti pro zkoumání a využití digitálních sbírek v kontextu kulturního dědictví. Tyto motivace společně vedly k vytvoření IIIF frameworku, který reagoval na potřeby institucí pracujících s digitalizovaným kulturním dědictvím a přinesl standardizovaný a efektivní přístup k distribuci digitálních obrazů na internetu.

## 3.2 API

API (Application Programming Interface) je soubor komunikačních protokolů a podprogramů, které programy používají ke vzájemné komunikaci. Programátor může využívat různé nástroje API, které mu usnadní a zjednoduší práci s programem. API také usnadňuje programátorům efektivní způsob vývoje jejich softwarových programů. Přijme požadavek od uživatele a odešle jej poskytovateli služby a poté opět odešle výsledek vygenerovaný poskytovatelem služby požadovanému uživateli [10].

Aktuální specifikace IIIF obsahuje celkem 6 API. Jedná se o *Image API* [11], *Presentation API* [12], *Authorization Flow API* [13], *Change Discovery API* [14], *Content Search API* [15] a *Content State API* [16]. První dvě zmíněné jsou klíčové, neboť se starají o vykreslení obrázku se zadáním příhodného URI z hlediska *Image API* a popis struktury díla z hlediska *Presentation API*. Jejich detailnímu popisu se budou věnovat sekce 3.3 a 3.4 [17].

*Authorization Flow API* se stará primárně o ověřování přístupu ke sdílenému zdroji. Proces ověřování může sahát od jednoduchého omezení podle IP adresy nebo dohody o prokliknutí až po vícefaktorové ověřování se zabezpečeným poskytovatelem identit. Specifikace popisuje, jak z klientské aplikace (například prohlížeče) zahájit interakci se systémem řízení přístupu, aby uživatel mohl získat všechna pověření, která potřebuje k zobrazení omezeného obsahu. Zároveň umožňuje klientovi zjistit, zda má uživatel aktuálně přístup ke zdroji, aniž by musel vědět cokoli o tom, jak a k čemu je oprávněn [13].

Dalším API je *Change Discovery API*, které poskytuje informace potřebné k zjišťování změn obsahu daného zdroje. Může se jednat o jakékoliv změny v manifestu, například jeho update. Vyhledávání zdrojů IIIF vyžaduje konzistentní a dobře srozumitelný vzor pro poskytovatele obsahu, kteří zveřejňují seznamy odkazů na svůj dostupný obsah. Tento proces lze optimalizovat tím, že poskytovatelé obsahu zveřejní popisy toho, jak se jejich obsah změnil, což umožní vyhledávacím systémům získat pouze zdroje, které byly od posledního vyhledávání změněny. Tyto změny mohou zahrnovat případy, kdy je obsah odstraněn nebo se jinak stane nedostupným [14].

*Content Search API* se stará o vyhledávání anotačního obsahu v rámci jednoho zdroje IIIF, jako je manifest, plátno, rozsah nebo kolekce. Vyhledávání metadat nebo jiných popisných vlastností není předmětem této specifikace. Mezi typické případy užití se může řadit například vyhledávání v textu generovaném OCR za účelem nalezení slov nebo frází v knize či v novinách [15].

Posledním zmíněným API je *Content State API*. Tato specifikace poskytuje způsob odkazování na prostředek rozhraní IIIF *Presentation API*, který lze použít k inicializaci zobrazení tohoto prostředku v libovolném klientovi. Tento popis se nazývá stav obsahu (content state). Specifikace poskytuje formát stavu obsahu a mechanismy pro jeho předávání mezi aplikacemi bez ohledu na jejich různá uživatelská rozhraní a možnosti. Cílem rozhraní je poskytnout standardizovaný formát pro sdílení určitého pohledu na jeden nebo více zdrojů *Presentation API*, jako je například kolekce nebo manifest (všechny typy zdrojů budou probrány v sekci 3.4.1). Případ užití může být zaměření prohlížeče na příslušnou část objektu, například na konkrétní řádek textu, který obsahuje hledaný výraz [16].

## 3.3 Image API

Rozhraní IIIF Image API [11], definované konsorciem IIIF, představuje webovou službu, která na základě standardního HTTP nebo HTTPS požadavku vrátí obrázek. Uživatel může prostřednictvím URI specifikovat oblast, velikost, rotaci, kvalitativní charakteristiky a formát žádaného obrázku. Je také možné sestavit URI tak, aby požadovalo základní technické informace o obrázku, což podporuje klientské aplikace. Toto API je navrženo primárně pro usnadnění systematického znovupoužití obrazových zdrojů uložených v digitálních obrazových archivech spravovaných organizacemi kulturního dědictví, ale může být přijato jakýmkoli obrazovým archivem nebo službou.

IIIF Image API [11] je zaměřeno na různé typy aplikací, včetně digitálních obrazových úložišť a distribuovaných obsahových sítí, webových aplikací zaměřených na obrázky, jako jsou prohlížeče s možností posunu a zvětšení, čtečky knih a klientů využívajících obrazový obsah k analýze nebo porovnání. Specifikace se primárně týká způsobu, jakým klienti mohou požadovat obrázky, nezabývá se však správou obrazů serverem. Podrobně specifikuje, jak server má reagovat na požadavky, které odpovídají určité syntaxi URI, ale nezahrnuje metody implementace, jako jsou algoritmy rotace, překódování, správa barev, komprese, ani reakce na URI, které nesplňují specifikovanou syntaxi. Toto usnadňuje flexibilitu implementace v oblastech s konkrétními omezeními nebo zvyklostmi komunity, zároveň podporuje interoperabilitu v běžných případech.

IIIF Image API [11] lze volat dvěma způsoby: požadavek na obrázek odvozený ze základního obrazového obsahu nebo požadavek na informace o obrazové službě, včetně jejích charakteristik a dostupných funkcí. Informace o požadavku jsou předány v segmentech cesty URI, nikoli jako parametry dotazu, což usnadňuje ukládání odpovědí do mezipaměti na serveru nebo v rámci standardní infrastruktury pro ukládání do mezipaměti webu.

Základní URI syntaxe zahrnuje čtyři sdílené parametry mezi různými typy požadavků: `scheme`, `server`, `prefix`, a `identifier`. Každý z těchto parametrů pomáhá definovat základní URI služby obrazu a určit zdrojový obrazový obsah. Když je základní URI dereferencováno, měla by být odpověď přesměrována na dokument s informacemi o obrázku. Detailní požadavky na obrázek nebo informace o obrázku pak mají své specifické URI šablony, které zahrnují dodatečné parametry, jako jsou `region`, `size`, `rotation`, `quality` a `format` pro obrázky nebo `cesta k info.json` pro informace o obrázcích.

## 3.3.1 Image Requests

Pro správnou konstrukci URI v rozhraní IIIF Image API je nezbytné dodržet všechny následující parametry v uvedeném pořadí. Obrázek je nejdříve extrahován z celé jeho plochy, následně se přizpůsobí požadovaným rozměrům, může být zrcadlen nebo otočen a poté je upraven do finální kvality a barevného formátu. Takto upravený obrázek je pak vrácen jako reprezentace pro dané URI.

### 3.3.1.1 Region

Parametr `region` definuje obdélníkovou část podkladového obsahu obrázku, která má být vrácena. Příklad se nachází na obrázku 3.1. Může nabývat hodnot:

- `Full`

Vrátí se celý obrázek bez toho, aby byl jakýmkoliv způsobem oříznut.

- `Square`

Vrátí se čtvercová oblast, jehož šířka i výška se rovnají délce kratšího rozměru celého obrázku. Oblast může být umístěna kdekoli v delším rozměru celého obrázku podle uvážení serveru, přičemž je často rozumným výchozím nastavením střed.

- `x,y,w,h`

Oblast vráceného obrázku je určena hodnotami pixelů. Hodnota `x` představuje počet pixelů od pozice 0 na vodorovné ose a hodnota `y` představuje počet pixelů od pozice 0 na svislé ose. Od této pozice `w` představuje šířku oblasti a `h` výšku oblasti v pixelech.

- `pct:x,y,w,h`

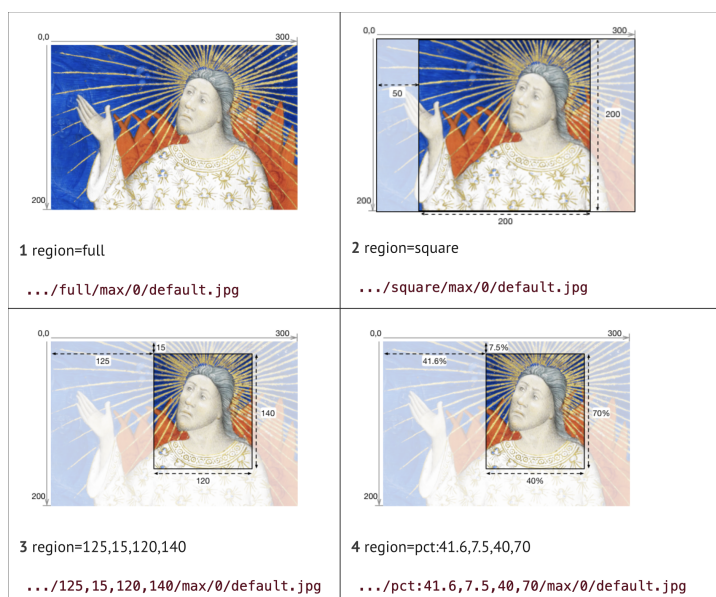
Vrácená oblast je představována stejným způsobem jako předchozí řešení, pouze se jedná o procentuální hodnoty.

### 3.3.1.2 Size

Parametr `size` určuje rozměry, na které má být extrahovaná oblast škálovaná, viz obrázek 3.2.

- `max`

Extrahovaná oblast se vrátí v maximální dostupné velikosti, ale nebude zvětšena.



Obrázek 3.1: Příklady správného použití parametru region

- $\wedge$ max

Extrahovaná oblast je transformována do maximálních rozměrů povolených pomocí parametrů `maxWidth`, `maxHeight` nebo `maxArea`, jak je uvedeno v technických specifikacích. Pokud jsou výsledné rozměry přesahující šířku a výšku původní extrahované oblasti v pixelech, dojde k přeškálování této oblasti.

- w,

Extrahovaná oblast by měla být škálována tak, aby šířka vráceného obrázku byla přesně rovna `w`. Hodnota `w` nesmí být větší než šířka obrázku.

- $\wedge$ w,

Extrahovaná oblast by měla být škálována tak, aby šířka vráceného obrázku byla přesně rovna `w`. Pokud je ale hodnota `w` větší než šířka pixelu extrahované oblasti, extrahovaná oblast se zvětší.

- ,h

Extrahovaná oblast by měla být škálována tak, aby výška vráceného obrázku byla přesně rovna `h`. Hodnota `h` nesmí být větší než výška obrázku.

- $\wedge$ ,h

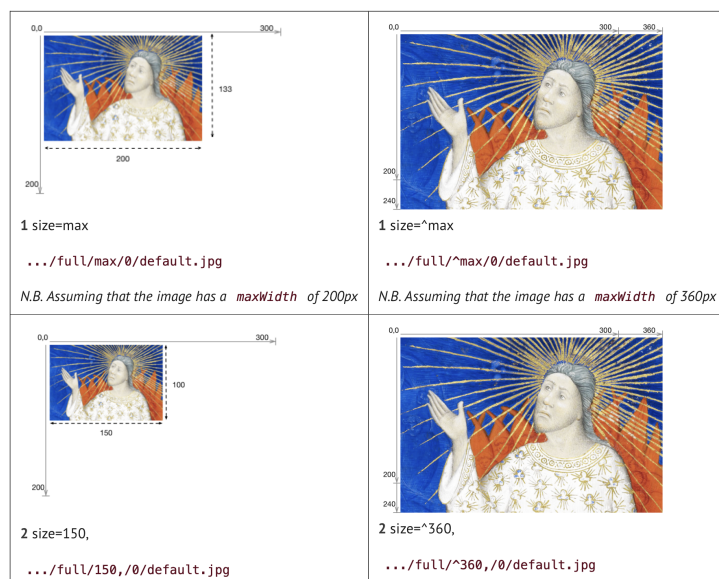
Extrahovaná oblast by měla být škálována tak, aby výška vráceného obrázku byla přesně rovna `h`. Pokud je ale hodnota `h` větší než výška pixelu extrahované oblasti, extrahovaná oblast se zvětší.

- `pct:n`  
Šířka a výška vráceného obrázku se škáluje na  $n$  procent šířky a výšky extrahované oblasti. Hodnota  $n$  nesmí být větší než 100.
- `^pct:n`  
Šířka a výška vráceného obrázku se škáluje na  $n$  procent šířky a výšky extrahované oblasti. Při hodnotách  $n$  větších než 100 je extrahovaná oblast zvětšena.
- `w,h`  
Šířka a výška vráceného obrazu jsou přesně  $w$  a  $h$ . Poměr stran vráceného obrazu se může výrazně lišit od extrahované oblasti, což by mohlo vést ke zkreslení obrazu. Hodnoty  $w$  a  $h$  nesmí být větší než odpovídající rozměry pixelů extrahované oblasti.
- `^w,h`  
Šířka a výška vráceného obrazu jsou přesně  $w$  a  $h$ . Poměr stran vráceného obrazu se může výrazně lišit od extrahované oblasti, což by mohlo vést ke zkreslení obrazu. Pokud jsou hodnoty  $w$  a  $h$  větší než odpovídající rozměry pixelů extrahované oblasti, extrahovaná oblast se zvětší.
- `!w,h`  
Extrahovaná oblast se škáluje tak, aby šířka a výška vráceného obrazu nebyly větší než  $w$  a  $h$  a zároveň byl zachován poměr stran. Vrácený obrázek musí být co největší, ale nesmí být větší než extrahovaná oblast,  $w$  nebo  $h$ , případně než limity stanovené serverem.
- `^!w,h`  
Extrahovaná oblast se škáluje tak, aby šířka a výška vráceného obrazu nebyly větší než  $w$  a  $h$  a zároveň byl zachován poměr stran. Vrácený obrázek musí být co největší, ale nesmí být větší než  $w$ ,  $h$  nebo limity stanovené serverem.

### 3.3.1.3 Rotation

Parametr rotace určuje zrcadlení a rotaci obrazu, viz obrázek 3.3. Číselná hodnota představuje počet stupňů otočení ve směru hodinových ručiček a může být libovolné číslo s pohyblivou řádovou čárkou od 0 do 360. Může obsahovat vykřičník (!), který označuje, že obraz by měl být zrcadlen podle  $y$  osy před použitím rotace.

- `n`  
Stupňů otáčení podle směru hodinových ručiček od 0 do 360.



Obrázek 3.2: Příklady správného použití parametru size

- `!n`

Obrázek by měl být zrcadlově obrácen a poté otočen stejným způsobem, jak je uvedeno výše.

### 3.3.1.4 Quality

Parametr kvality určuje, zda bude obrázek vrácen barevný, ve stupních šedi nebo černobílý. Příklad je na obrázku 3.4.

- `Color`

Obrázek je vrácen s celou barevnou informací.

- `Gray`

Obrázek je vrácen ve stupních šedi, kde každý pixel je černý, bílý nebo jakýkoli odstín šedé mezi nimi.

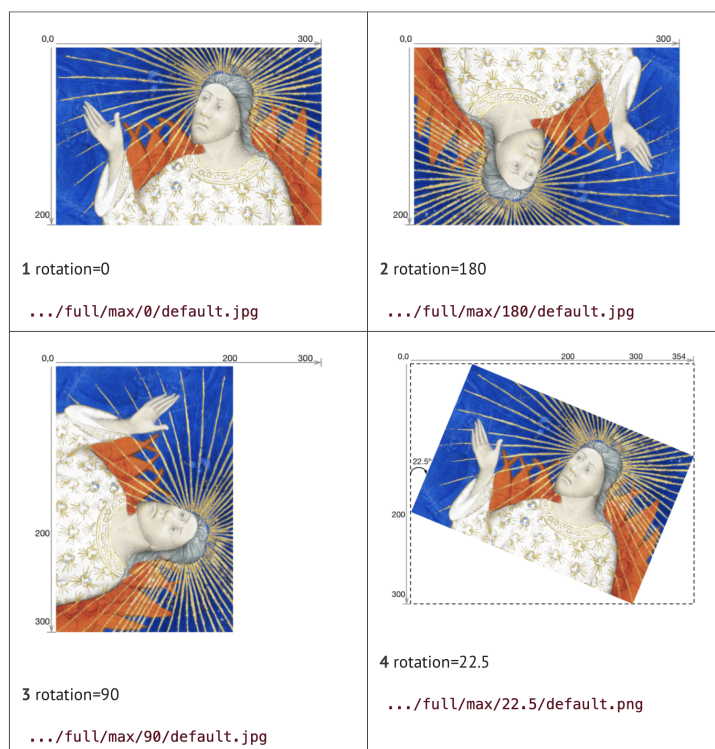
- `Bitonal`

Vrácený obrázek je bitonální, přičemž každý pixel je buď černý nebo bílý.

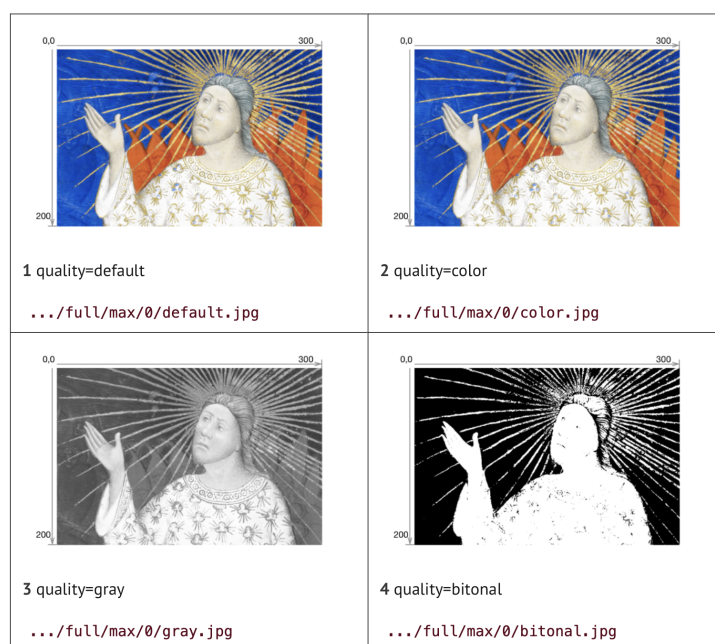
- `Default`

Obrázek je vrácen s použitím výchozí kvality serveru.





Obrázek 3.3: Příklady správného použití parametru Rotation



Obrázek 3.4: Příklady správného použití parametru Quality

### 3.3.1.5 Format

Formát vráceného obrázku je vyjádřen příponou na konci URI. Může se jednat o:

- jpg
- tif
- png
- gif
- jp2
- pdf
- webp

### 3.3.1.6 **Dodatečné informace k implementaci**

Parametry by měly být interpretovány následujícím způsobem:

```
1 Region THEN Velikost THEN Rotace THEN Kvalita THEN Formát
2
```

Zdrojový kód 3.1: Ukázka použití parametrů v URI

Parametry velikost a oblast jsou udávány v procentech a parametr rotace umožňuje kladné hodnoty čísel s pohyblivou řádovou čárkou. Pokud je to možné, měly by se používat celočíselné hodnoty.

Stejný obrázek je možné vyžádat pomocí různých kombinací parametrů. Kanonická syntaxe URI umožňuje statické implementace založené na souborovém systému, které budou mít pouze jeden URI, na kterém je obsah dostupný. Ukládání do mezipaměti je výrazně efektivnější, a to jak na straně klienta, tak na straně serveru, pokud jsou používány URI stejné pro různé systémy a relace. Časy odezvy lze zlepšit tím, že se zamezí přesměrování z požadované nekanonické syntaxe URI na kanonickou syntaxi přímým použitím kanonické formy. Aby byly podpořeny výše uvedené požadavky, měli by klienti konstruovat URI požadavky na obrázky pokud možno s použitím následujících kanonických hodnot parametrů

Rozhraní IIIF Image API je rozšiřitelné v rámci syntaxe URI požadavku na obrázek přidáním nových vzorů parametrů pro parametry oblast, velikost a rotace nebo nových hodnot pro parametry kvalita a formát. Informace o požadavku nad rámec stávajících parametrů lze předat image serveru jako parametry dotazu.

## 3.3.2 **Image Information**

Servery musí podporovat požadavky na *Image information*. Odpovědí je JSON dokument, který obsahuje technické vlastnosti celého obrazu. Může také obsahovat informace o právech a službách souvisejících s obrazem.

### 3.3.2.1 Technické vlastnosti

JSON response má několik technických vlastností, které popisují dostupné funkce pro obsah obrázku. Většina těchto vlastností jsou požadované a patří mezi ně:

- **@context**  
Hodnotou musí být buď URI či JSON pole s více URI. Tato vlastnost říká Linked data processorům, jak *Image information* interpretovat.
- **id**  
Id je základní URI obrázku podle definice v syntaxi URI, včetně schématu, serveru, předpony a identifikátoru bez koncového lomítka.
- **type**  
Jedná se o typ pro rozhraní API Image. Hodnota musí být řetězec `ImageService3`.
- **protocol**  
Obsahuje URI `http://iiif.io/api/image`, podle kterého lze určit, že dokument popisuje obrazovou službu, která je verzí rozhraní IIIF Image API.
- **profile**  
Řetězec označující nejvyšší tzv. úroveň shody, kterou služba plně podporuje. Definuje sadu podporovaných parametrů, které odpovídají různým úrovním shody s rozhraním API. Jsou definovány tři úrovně shody. `Level0` je minimální soubor parametrů a funkcí, které musí být implementovány, aby byla služba kvalifikována jako vyhovující specifikaci IIIF Image API. Dále jsou pak dvě další vyšší úrovně, tedy `Level1` a `Level2`.
- **width**  
Šířka celého obrázku v pixelech zadaná jako celé číslo.
- **height**  
Výška celého obrázku v pixelech zadaná jako celé číslo.

JSON odpověď poté může obsahovat také volitelné vlastnosti.

- **maxWidth**  
Maximální šířka v pixelech, která je pro tento obrázek podporována. Šířka nesmí být větší než tato hodnota. Hodnota `MaxWidth` musí být zadána, pokud je zadána i hodnota `maxHeight`.

- `maxHeight`

Maximální výška v pixelech, která je pro tento obrázek podporována. Výška nesmí být větší než tato hodnota. Pokud je zadána hodnota `maxWidth` a `maxHeight` nikoliv, pak `maxHeight = maxWidth`.

- `maxArea`

Maximální podporovaná plocha v pixelech pro tento obrázek. `Width * height` nesmí být větší než je tato hodnota.

### 3.3.2.2 Sizes

JSON může obsahovat vlastnost `sizes`, která se používá k popisu preferovaných kombinací výšky a šířky pro reprezentaci celého obrázku. Tato vlastnost je také volitelná. Jedná se o pole JSON objektů s vlastnostmi `width` a `height`. Tyto velikosti určují preferované hodnoty, které mají být uvedeny v syntaxi `w,h` parametru požadavku na velikost pro zmenšené verze celého obrázku. v případě serverů, které nepodporují požadavky na libovolné velikosti, mohou být tyto velikosti jediné dostupné.

Objekty JSON v poli `sizes` mají vlastnosti `type`, `width` a `height`. Požadavky na obrázky těchto velikostí by měly mít parametr `region=full`, parametr `size` v kanonickém tvaru `w,h` a rotaci 0. Úplná adresa URL pro obrázek s výchozí kvalitou ve formátu `jpg` by tedy byla:

```
scheme://server/prefix/identifier/full/width,height/0/def.jpg
```

### 3.3.2.3 Tiles

JSON response může obsahovat vlastnost `tiles`, která popisuje sadu oblastí obrazu s konzistentní výškou a šířkou, které lze vizuálně spojit. JSON objekty v poli `tiles` mají vlastnosti `type`, `width`, `height` a tzv. `scaleFactors`. Jedná se o sadu faktorů měřítka rozlišení pro předdefinované dlaždice obrázku, vyjádřená jako kladná celá čísla, kterými se dělí celá velikost obrázku. Například faktor měřítka 4 znamená, že služba může efektivně poskytovat obrázky o velikosti 1/4 nebo 25% výšky a šířky celého obrázku.

### 3.3.2.4 Preferred Formats

Vlastnost `preferredFormat` umožňuje vydavateli vyjádřit preferenci formátu, který klient požaduje. Například podpořit použití efektivnějšího formátu nebo navrhnout formát, který poskytne lepší výsledky pro obsah obrázku, jako je bezztrátový `webp` nebo `png` pro kresby nebo grafiku.

### 3.3.2.5 Práva

Jedná se o řetězec, který identifikuje licenci nebo prohlášení o právech vztahující se na obsah prostředku, například pixely obrázku. Hodnota musí být vybrána ze sady *Creative Commons license URIs*, URI prohlášení o právech *RightsStatements.org* nebo přidáných prostřednictvím rozšíření. Zahrnutí této vlastnosti je informativní a může být například použito k zobrazení ikony reprezentující prohlášení o právech.

### 3.3.2.6 Linking Properties

JSON response může obsahovat linking vlastnosti, které odkazují na externí zdroje, včetně služeb, které prohlížeči zpřístupňují další funkce.

- `partOf`

Link na jiný prostředek, který odkazuje na tuto obrazovou službu, například link na canvas nebo manifest. Hodnota musí být pole objektů JSON.

- `seeAlso`

Link na externí zdroj, který souvisí s tímto zdrojem, například popis XML nebo RDF. Vlastnosti externího zdroje by měly klientovi pomoci vybrat mezi více popisy (pokud jsou poskytnuty) a aby mohl dokument vhodně použít. URI dokumentu musí identifikovat jedinou reprezentaci dat v určitém formátu. Hodnotou musí být pole objektů JSON.

### 3.3.3 Server responses

V rámci IIIF Image API je zásadní, aby servery podporovaly metodu HTTP `GET` pro získání obrázků a HTTP `OPTIONS` pro CORS (Cross-origin resource sharing) preflight požadavky, což umožňuje ověření oprávnění klientů před samotným požadavkem. Je doporučeno také implementovat metodu HTTP `HEAD` pro získání metadat bez stažení obsahu. Servery by měly zpracovávat úspěšné odpovědi pomocí stavových kódů 200 pro přímé doručení obsahu nebo 3xx pro přesměrování. v případě chyb by měly servery vrátit příslušný HTTP stavový kód s popisem problému pro uživatele.

### 3.3.4 Zabezpečení

Složení URI má jen málo bezpečnostních aspektů kromě možného odhalení citlivých informací v URI nebo odhalení chování uživatelů při prohlížení.

Serverové aplikace implementující Image API by měly zvážit možné útoky typu odepření služby a zranitelnosti při ověřování založené na podvržení DNS (Domain

Name System). Aplikace musí dbát na to, aby analyzovaly a upravovaly příchozí požadavky (URI) způsobem, který zabrání útokům typu overflow, injection a directory traversal.

## 3.4 Presentation API

Sbírkám digitalizovaných fyzických objektů i velké části rodícího se digitálního obsahu prospívá standardizovaný popis jejich struktury, uspořádání a způsobu prezentace. Tento dokument specifikuje tento standardizovaný popis sbírky nebo složeného objektu pomocí formátu JSON.

Složený objekt může zahrnovat řadu stránek, ploch nebo časových úseků; například jeden pohled na obraz, dvě strany fotografie, čtyři světové strany sochy, mnoho stránek vydání novin nebo knihy nebo trvání jednoho jednání opery. Tato specifikace se zabývá tím, jak zajistit pořadí těchto pohledů nebo rozsahů, odkazy na zdroje potřebné k jejich prezentaci a popisné informace potřebné k tomu, aby uživatel pochopil, co je vidět nebo slyšet.

Cílem Presentation API [12] je poskytnout informace nezbytné k tomu, aby bylo možné prezentovat složené digitální objekty uživateli v online prostředí, často ve spojení s Image API.

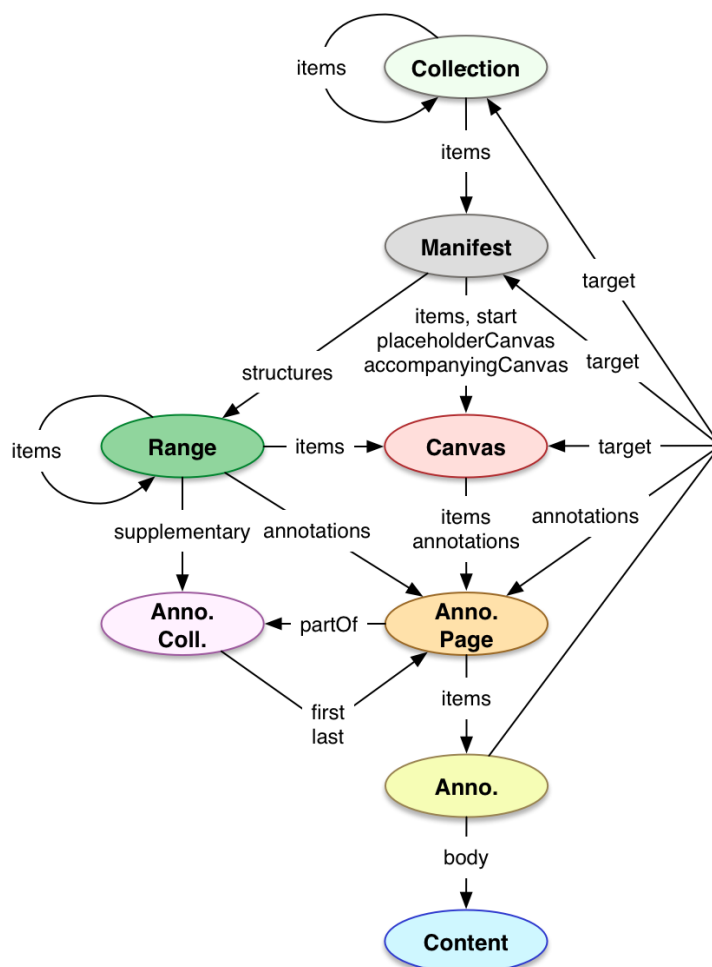
Implementace této specifikace budou schopny:

- Zobrazit uživateli digitalizované obrázky, video, audio a další typy obsahu spojené s konkrétním fyzickým nebo digitálním objektem.
- Umožnit uživateli procházet mezi více zobrazeními nebo časovými rozsahy objektu, a to buď postupně, nebo hierarchicky.
- Zobrazit popisné informace o objektu, zobrazení nebo navigační strukturu, aby uživatel poskytl kontext.
- Poskytovat sdílené prostředí, v němž mohou vydavatelé i uživatelé doplnit objekt a jeho obsah o další informace.

### 3.4.1 Resource Types

Výše popsané cíle vyžadují model, ve kterém lze charakterizovat složený objekt (prostřednictvím zdroje (nebo také *resource*) manifest) a jednotlivé pohledy na objekt (zdroje canvas). Každý pohled může odkazovat na obrázky, zvuk, video a další zdroje obsahu, aby bylo možné jej vhodně vykreslit. Složený objekt může mít také části; například kniha může mít kapitoly o několika stránkách nebo divadelní hra může být rozdělena na jednání a scény (zdroje range) a mohou existovat skupiny takových objektů (zdroje collection). Tyto typy zdrojů spolu s jejich vlastnostmi tvoří IIIF

Presentation API. Následující obrázek 3.5 zobrazuje propojení jednotlivých typů zdrojů.



Obrázek 3.5: Data model (čerpáno z [12])

Specifikace definuje následující typy zdrojů:

### 3.4.1.1 Collection

*Collection* (také jako *kolekce*) slouží k zobrazení seznamu manifestů, které jsou k dispozici. Kolekce mohou zahrnovat jak jiné kolekce, tak i manifesty, a vytvářet tak stromovou hierarchii.

Hlavním cílem použití kolekce je například:

- Načíst předem definovanou sadu manifestů v době inicializace.
- Přijmout sadu Manifestů, například výsledky vyhledávání, pro vykreslení.
- Vizualizovat seznamy nebo hierarchie souvisejících manifestů.

- Poskytovat navigaci v seznamu nebo hierarchii dostupných manifestů.

Kolekce mohou být také vloženy do jiných kolekcí, například když se kolekce používá především k rozdělení větší kolekce na lépe zvládnutelné části. Nicméně manifesty nesmí být vloženy do kolekcí. Vložená kolekce by také měla mít svůj vlastní URI, ze kterého je dostupný popis JSON.

Na manifesty nebo kolekce lze odkazovat z více než jedné kolekce. Například instituce může definovat čtyři sbírky: jednu pro moderní díla, jednu pro historická díla, jednu pro noviny a jednu pro knihy. Manifest pro moderní noviny by se pak objevil jak ve sbírce pro moderní díla, tak ve sbírce pro noviny.

### 3.4.1.2 Manifest

Typ zdroje *manifest* obvykle představuje jeden objekt a jakékoli duševní dílo nebo díla, která jsou v tomto objektu obsažena. Obsahuje zejména popisné informace, informace o právech a propojení objektu. Manifest obsahuje plátna, která by měla být vykreslena jako zobrazení objektu a obsahuje dostatek informací, aby se klient mohl inicializovat a začít uživateli zobrazovat obsah.

Identifikátor v `id` musí být možné dereferencovat pro načtení popisu JSON `manifestu`, a proto musí používat schéma HTTP(S) URI.

Manifest musí mít property `items` (všechny vlastnosti budou detailněji popsány v sekci 3.4.2), což je pole objektů JSON-LD (JSON formát pro *linked data*). Každý objekt je `canvas` s požadavky popsány v následující sekci 3.4.1.3. Manifest může mít také property `structures`, ve které je uveden jeden nebo více rozsahů, které popisují další strukturu obsahu, například takovou, která může být vykreslena jako obsah. Manifest může mít property `annotations`, která obsahuje zdroje `annotation page`, kde `annotations` mají manifest jako svůj cíl. Obvykle se bude jednat o anotace ve stylu komentářů.

### 3.4.1.3 Canvas

*Canvas* (také jako *plátno*) představuje jednotlivou stránku nebo zobrazení a slouží jako centrální bod pro sestavení různých zdrojů obsahu, které tvoří zobrazení. Plátna musí být identifikována pomocí URI HTTP(S). Na plátna lze odkazovat odděleně od manifestu prostřednictvím jejich URI a také je lze vložit.

Zdroje obsahu jsou s plátnem spojeny prostřednictvím webových anotací. Obsah, který má být vykreslen jako součást plátna, musí být přidružen pomocí anotace, která má motivační hodnotu *painting*. Tyto anotace jsou zaznamenány v položkách jedné nebo více `annotation pages`, na které se odkazuje v poli položek plátna.



Obsah, který je odvozen z plátna, jako je OCR nebo slova vyslovená ve zvukové reprezentaci, musí být spojen s anotací, která má motivační hodnotu *supplementing*. Anotace mohou mít i jakékoli jiné motivační hodnoty.

Plátno musí mít obdélníkový poměr stran (popsaný pomocí vlastností *height* a *width*) nebo dobu trvání, která udává časový rozsah. Tyto rozměry umožňují přiřazení zdrojů ke konkrétním oblastem plátna v rámci poskytnutého prostorového nebo časového rozsahu. Obsah nesmí být přiřazen k prostoru nebo času mimo rozměry plátna, například na souřadnicích pod 0, 0, větších než výška nebo šířka, před 0 sekundami nebo po uplynutí doby trvání. Zdroje obsahu, které mají rozměry, jež nejsou pro plátno definovány, nesmí být s tímto plátnem spojeny pomocí anotace, která má motivační hodnotu *painting*. Například je platné použít anotaci, která má motivační hodnotu malování, k přiřazení obrázku (který má pouze výšku a šířku) k plátnu, které má všechny tři rozměry, ale je chybou přiřadit zdroj Video (který má výšku, šířku a trvání) k plátnu, které nemá všechny tři rozměry. Takový prostředek by měl být namísto toho odkazován pomocí vlastnosti *rendering* nebo pomocí anotací, které mají ve vlastnosti *annotations* jinou motivační hodnotu než *painting*.

### 3.4.1.4 Range

*Ranges* (jinak také jako *rozsahy*) se používají k reprezentaci struktury v rámci objektu mimo výchozí pořadí plátna ve vlastnosti *items* v manifestu, například novinové sekce nebo články, kapitoly v knize nebo části v hudební skladbě. Rozsahy mohou zahrnovat plátna, části plátna nebo jiné rozsahy a vytvářet stromovou strukturu podobnou obsahu.

Záměrem přidání rozsahu do manifestu je umožnit klientovi zobrazit lineární nebo hierarchické navigační rozhraní, které uživateli umožní rychlý pohyb po obsahu objektu.

Rozsahy musí mít URI `HTTP(S)`. Rozsahy nejvyšší úrovně jsou vloženy nebo externě odkazovány v rámci manifestu ve vlastnosti *structures*. Tyto rozsahy nejvyšší úrovně pak vkládají nebo odkazují na jiné rozsahy, plátna nebo části plátna ve vlastnosti *items*. Každá položka ve vlastnosti *items* musí být objekt JSON a musí mít vlastnosti *id* a *type*.

Všechna plátna nebo části, které by měly být považovány za součást rozsahu, musí být zahrnuty v property *items* tohoto rozsahu nebo v položkách potomka rozsahu.

Plátna a části plátna nemusí na sebe navazovat ani nemusí být ve stejném pořadí jako ve vlastnosti položek manifestu nebo jiného *range*. Příkladem mohou být novinové články, které pokračují v různých částech, kapitola, která začíná v polovině

stránky, nebo časové úseky jednoho plátna, které představují různé části hudební skladby.

Rozsahy mohou odkazovat na annotation collection, která má obsah rozsahu, pomocí supplementary property. Odkazovaná annotation collection bude obsahovat anotace, které se zaměřují na oblasti pláten v rámci rozsahu a propojují zdroje obsahu s těmito plátny.

### 3.4.1.5 Annotation Page

Spojení obrázků a dalšího obsahu s příslušnými plátny se provádí pomocí annotations (nebo-li anotace, poznámky). Tradičně se anotace používají pro přiřazení komentáře ke zdroji, jehož se text nebo tělo anotace týká. Model webových anotací ovšem umožňuje přiřazení jakéhokoli zdroje k jakémukoli jinému zdroji nebo jeho části a je opakovaně používán jak pro komentáře, tak pro zdroje obrazu na plátně. Mezi další zdroje kromě obrázků může patřit úplný text objektu, notové záznamy, hudební představení, prepisy diagramů, komentáře, značky, video, data a další.

Tyto anotace jsou shromážděny ve zdroji *annotation page* (jinak také jako *anotační stránka*), které jsou zahrnuty v property `items` plátna. Každá annotation page může být vložena celá, pokud mají být anotace zpracovány co nejdříve, když uživatel přejde na dané plátno, nebo jako odkaz na externí stránku. Všechny anotace na annotation page by měly mít jako target plátno.

### 3.4.1.6 Annotation

*Anotace* spojují zdroje obsahu s plátny. Pro viditelné nebo slyšitelné zdroje se používá stejný mechanismus jako pro prepisy, komentáře, značky a další obsah. To poskytuje jednotnou metodu pro přiřazování informací a poskytuje rámec založený na standardech pro rozlišování částí zdrojů a částí plátna. Vzhledem k tomu, že anotace lze přidávat později, podporuje distribuovaný systém, v němž mohou vydavatelé sladit svůj obsah s popisy vytvořenými jinými.

Anotace se řídí datovým modelem webových anotací (Web Annotation data model). Musí mít své vlastní URI `HTTP(S)`, které jsou uvedeny ve vlastnosti `id`. Pokud je URI dereferencováno, měl by být vrácen popis anotace ve formátu JSON-LD v souladu s protokolem *Web Annotation Protocol*.

### 3.4.1.7 Content Resources

*Content resources* (taky jako *zdroje obsahu*) jsou externí webové zdroje, na které se odkazuje v rámci manifestu nebo kolekce. Patří sem obrázky, video, audio, data, webové stránky nebo jakýkoli jiný formát.

Obsah spojený s plátnem (a tedy i obsah manifestu) je poskytován vlastností body anotací s motivem `painting`. na zdroje obsahu lze také odkazovat z vlastností `thumbnail`, `homepage`, `logo`, `rendering` a `seeAlso`.

S plátnem lze zacházet jako se zdrojem obsahu pro účely jeho anotace na jiných plátnech. v této situaci může být plátno vloženo do anotace nebo může vyžadovat dereferenci pro získání svého popisu.

### 3.4.1.8 Annotation Collection

to, na které plátno nebo prostředek se zaměřují. To umožňuje například shromáždit všechny anotace, které tvoří určitý překlad textu knihy. Klient by pak mohl prezentovat uživatelské rozhraní, které umožňuje zobrazit nebo skrýt všechny anotace v kolekci anotací podle preferencí uživatele.

## 3.4.2 Resource Properties

Většina vlastností definovaných v této specifikaci může být spojena s kterýmkoli z výše popsaných typů prostředků a může mít více než jednu hodnotu.

### 3.4.2.1 Descriptive Properties

Tyto properties popisují nebo reprezentují zdroj, ke kterému jsou přiřazeny, a obvykle se vykreslují uživateli.

#### **Label**

Jedná se o čitelný štítek, jméno nebo název. Může být plně internacionalizovaný a každý jazyk může mít více hodnot.

#### **Metadata**

Metadata jsou uspořádaný seznam popisů, které se zobrazí uživateli při interakci se zdrojem, zadaný jako dvojice položek `label` a `value`. Obsah těchto položek je určen pouze pro prezentaci.

#### **Summary**

`Summary` je krátké textové shrnutí, které má být uživateli sděleno v případě, že se nezobrazují metadata. Řídí se stejným vzorem jako výše popsaná vlastnost `label`.

#### **RequiredStatement**

Text, který se objeví při zobrazení nebo použití resource typu. Property lze například použít k zobrazení prohlášení o autorských právech nebo jakéhokoli jiného textu, který vydavatelská organizace považuje za důležitý pro zobrazení uživateli.

### **Rights**

Řetězec, který identifikuje licenci nebo prohlášení o právech vztahující se na obsah prostředku, například JSON manifestu nebo pixely obrázku. Hodnota musí být vybrána ze sady *Creative Commons license URIs*, URI prohlášení o právech *RightsStatements.org* nebo přidáných prostřednictvím rozšíření. Zahrnutí této vlastnosti je informativní a může být například použito k zobrazení ikony reprezentující prohlášení o právech.

Pokud chceme zobrazit informace o právech přímo uživateli nebo je zapotřebí label definovaný vydavatelem, pak se doporučuje zahrnout tyto informace pomocí property `requiredStatement` nebo do property `metadata`.

### **Provider**

Provider je organizace nebo osoba (reprezentovaná jako agent), která se podílela na poskytnutí obsahu zdroje. Od vlastnosti `requiredStatement` se liší tím, že údaje jsou strukturované, což klientovi umožňuje více než jen prezentovat text, ale místo toho má k dispozici bohatší informace o osobách a organizacích, které může použít v různých rozhraních.

### **Thumbnail**

Zdroj obsahu, například malý obrázek nebo krátký zvukový klip, který představuje zdroj s vlastností miniatury. Zdroj může mít více zdrojů miniatur, které mají stejný nebo různý typ a formát.

### **NavDate**

Datum, které mohou klienti použít pro účely navigace při zobrazení zdroje uživateli v uživatelském rozhraní založeném na datu, například v kalendáři nebo na časové ose. v property `metadata` by měly být zahrnuty popisnější rozsahy dat, určené pro zobrazení přímo uživateli.

### **PlaceholderCanvas**

Plátno, které poskytuje dodatečný obsah pro použití před vykreslením hlavního obsahu prostředku nebo jako reklamu či náhradu tohoto obsahu. Příkladem mohou být obrázky, text a zvuk nahrazující obsah videa předtím, než uživatel zahájí přehrávání. Obsah poskytovaný vlastností `placeholderCanvas` se liší od `thumbnailu` tím, že klient může použít `thumbnail` pro shrnutí a navigaci více zdrojů a poté zobrazit obsah z vlastnosti `placeholderCanvas` jako součást úvodní prezentace jednoho

zdroje.

### 3.4.2.2 Technical Properties

Tyto vlastnosti popisují technické vlastnosti zdrojů a klient je obvykle zpracovává, aby pochopil, jak prostředek vykreslit.

#### **Id**

Jedná se o URI, které identifikuje zdroj. Všechny resource typy musí obsahovat property `id`.

#### **Type**

Typ nebo třída resource typu. Může nabývat následujících hodnot:

- Dataset
- Image
- Model
- Sound
- Text
- Video

#### **Format**

Konkrétní typ média, například *image/jpeg*. To je důležité pro rozlišení různých formátů stejného celkového typu zdroje, například pro rozlišení textu v XML od prostého textu.

Tento formát se liší od property `formats` v rozhraní *API Image*, která udává rozšíření. v tomto případě by bylo nevhodné ji použít, protože `format` lze použít u jakéhokoli zdroje obsahu, nejen u obrázků.

#### **Language**

Jedná se o jazyk nebo jazyky použité v obsahu tohoto zdroje.

#### **Profile**

Schéma nebo pojmenovaná sada funkcí dostupných z daného prostředku. Profil může dále upřesnit typ nebo formát externího prostředku nebo služby a umožnit klientům přizpůsobit zacházení se zdrojem.

#### **Height, width a duration**

Výška a šířka resource typu canvas nebo externího obsahu. U `resources content` se hodnota udává v pixelech. U `resources canvas` nemá tato hodnota jednotku. Kombinace `height` a `width` vyjadřuje poměr stran prostoru, ve kterém jsou zdroje obsahu umístěny. Hodnota obou properties musí být kladné celé číslo.

Property `duration` se používá u canvas resource typu nebo externího obsahu a udává se v sekundách.

#### **ViewingDirection**

Směr, ve kterém se má sada canvases zobrazit uživateli. Tato specifikace definuje čtyři hodnoty směru.

- `left-to-right`
- `right-to-left`
- `top-to-bottom`
- `bottom-to-top`

#### **Behavior**

Soubor funkcí (chování), které by vydavatel obsahu chtěl, aby klient při prezentaci zdroje používal. U více spisů se může jednat například o správné stránkování atd.

### 3.4.2.3 Linking Properties

Tyto vlastnosti jsou odkazy nebo vazby mezi zdroji a dělí se na externí odkazy, kdy je odkazovaný objekt mimo prostor IIIF, a interní odkazy, kdy je odkazovaný objekt resource IIIF.

#### **Homepage**

Webová stránka o objektu reprezentovaném prostředkem. Webovou stránku obvykle publikuje organizace odpovědná za objekt a může být generována systémem pro správu obsahu nebo jiným katalogizačním systémem. Zdroj musí být možné zobrazit přímo uživateli. Zdroje, které spolu souvisejí, ale nejsou domovskými stránkami, musí být místo toho přidány do property metadata s vhodným označením `label` nebo `value` popisující vztah.

#### **Logo**

Malý obrazový prostředek, který představuje agenta, k němuž je přiřazen. Logo musí být při zobrazení nebo použití zřetelně zobrazeno, aniž by byl obrázek oříznut,

otočen nebo jinak zkreslen. Doporučuje se, aby pro tento obrázek byla k dispozici služba IIF Image API pro další manipulace.

#### **Rendering**

Zdroj, který je alternativní non-IIF reprezentací zdroje, který má vlastnost vykreslování. Takové reprezentace obvykle nelze vykreslit na jediné plátno, protože buď obsahují příliš mnoho pohledů, mají nekompatibilní rozměry, nebo se jedná o složené prostředky vyžadující další vykreslovací funkce. Vykreslovací prostředek musí být možné zobrazit přímo lidskému uživateli, ačkoli prezentace může být mimo klienta IIF. Příkladem může být vykreslení knihy ve formátu PDF, prezentace s obrázky budovy nebo 3D model sochy.

#### **Service**

Služba, se kterou může klient přímo komunikovat a získat další informace nebo funkce pro použití prostředku, který má vlastnost služby, například z obrázku na základní URI přidružené služby IIF Image API. Zdroj služby by měl mít přiřazený další informace, aby klient mohl určit, jak jej vhodně využít.

#### **SeeAlso**

Strojově čitelný zdroj, například popis XML nebo RDF. Vlastnosti by měli být uvedeny tak, aby klientovi pomohly vybrat mezi více popisy (pokud jsou poskytnuty). Pokud je třeba vztah mezi zdrojem a dokumentem upřesnit, pak by měl dokument obsahovat tento vztah. URI dokumentu musí identifikovat jedinou reprezentaci dat v určitém formátu.

#### **PartOf**

Když klient narazí na property `partOf`, může načíst odkazovaný obsahující resource, pokud není vložen do aktuální reprezentace, aby přispěl ke zpracování obsaženého zdroje. Například vlastnost `partOf` na plátně lze použít k odkazu na externí manifest, aby bylo možné zjistit další relevantní informace. Podobně může manifest odkazovat na obsaženou kolekci, aby pomohl při navigaci.

#### **Start**

Plátno nebo část plátna, které má klient zobrazit při inicializaci zdroje s vlastností `start`. Odkaz na část plátna se zpracovává stejným způsobem, jakým se ranges odkazují na části pláten. Tato vlastnost umožňuje klientovi začít s prvním plátnem, které obsahuje zajímavý obsah, místo toho, aby uživatel musel ručně přecházet k jeho nalezení.

### Supplementary

Klient supplementary property může použít k tomu, aby při interakci s rozsahem prezentovali uživateli další obsah z jiného plátna nebo aby přešli na další část rozsahu v rámci téhož plátna. Například rozsah může představovat novinový článek, který se rozprostírá na nesouvislých stránkách, a pak použije supplementary property k odkazu na kolekci anotací, která se skládá z anotací zaznamenávajících text, rozdělených na stránky anotací pro každou novinovou stránku.

### 3.4.2.4 Structural Properties

Tyto properties definují strukturu objektu reprezentovaného ve formátu IIIF tím, že umožňují začlenění podřízených prostředků do rodičovských, například canvas do manifestu. ve většině případů se používají `items`, existují však dva zvláštní případy pro různé druhy struktur.

#### Items

Velká část funkcí rozhraní IIIF Presentation API jednoduše zaznamenává pořadí, v jakém se podřízené zdroje vyskytují v rámci nadřazeného zdroje, například kolekce nebo manifesty v rámci nadřazené kolekce nebo plátna v rámci manifestu. Všechny tyto situace jsou pokryty touto vlastností.

#### Structures

Strukturu objektu reprezentovaného jako manifest lze popsat pomocí hierarchie rozsahů. Rozsahy lze použít k popisu "obsahu" objektu nebo jiných struktur, s nimiž může uživatel pracovat mimo pořadí dané vlastností položek manifestu. Hierarchie je vytvořena vnořením podřízených rozsahů do pole `items` vyšší úrovně rozsahu. Nejvyšší úroveň rozsahu těchto hierarchií je uvedena ve vlastnosti `structures`.

#### Annotations

Uspořádaný seznam annotation pages, které obsahují komentáře nebo jiné anotace, oddělené od annotations, které se používají k malování obsahu na canvas.

## 3.5 IIIF image viewers

Aby mohl uživatel media (v rámci této práce obrázky) přenášené pomocí IIIF prohlížet, existují tzv. *Image viewers* (také jako *prohlížeče obrázků*). Jedná se o software (obvykle webový), který umožňuje prohlížení a manipulaci s těmito obrázky. Někdy se také nazývá jako *klient*. Prostřednictvím tohoto nástroje mohou koncoví uživatelé interagovat s materiály IIIF, a mnoho image viewers přidává další funkce, jako je možnost přidávat anotace nebo vyhledávat v prepisech či překladech [5].



Existuje mnoho těchto prohlížečů IIIF. Některé jsou nástroji pro všeobecné použití, jiné se specializují na konkrétní druhy obsahu nebo funkcí. Prohlížeče kompatibilní s IIIF obecně umožňují uživatelům posouvat, zvětšovat, otáčet a měnit velikost obrazů a přehrávat zvukové či vizuální soubory. Některé umožňují přidávat poznámky pomocí textu, zvuku a dalších prvků. Jiné umožňují porovnávat objekty z jedné sbírky vedle sebe (nebo dokonce objekty z více sbírek, pokud je uživatelům zpřístupněn manifest objektu) [18].

Následující sekce se bude věnovat srovnáním několika vybraných image viewers.

### 3.5.1 Mirador

Mirador je konfigurovatelný, rozšiřitelný a snadno integrovatelný prohlížeč obrázků, který umožňuje anotovat a porovnávat obrázky z úložišť nacházející se po celém světě. Byl optimalizován pro zobrazování zdrojů z úložišť, která podporují IIIF. Poskytuje dlaždicové prostředí pro porovnávání více obrazových zdrojů, synchronizovanou strukturální a vizuální navigaci obsahu, vytváření a prohlížení anotací na plátnech s možností hlubokého zvětšení, zobrazení metadat, čtení knih, záložek a další funkce. Poskytuje několik pracovních prostorů pro porovnávání obrazových zdrojů, které jsou vhodné pro použití v oblasti kulturního dědictví i výzkumu. Dále umožňuje uživatelům vytvářet a sdílet anotace, což je užitečné pro výzkumné projekty nebo online kurzy, kde je potřeba sdílet poznámky či interpretace obrázků [19].

Mirador získal popularitu také díky svému otevřenému kódu a aktivní komunitě vývojářů. Jeho modulární architektura umožňuje snadnou integraci s různými digitálními repozitáři a knihovními systémy, což umožňuje široké použití tohoto vieweru ve vědeckých institucích, muzeích a knihovnách po celém světě [20].

#### 3.5.1.1 Výhody

- Flexibilní a interaktivní uživatelské rozhraní.
- Možnost zobrazit dva obrázky v jednom vieweru.
- Podpora vytváření anotací.

#### 3.5.1.2 Nevýhody

- Pro začátečníky může být příliš složité z pohledu množství funkcionalit.

### 3.5.2 Universal Viewer

Tento viewer využívá široce dostupné technologie s open source kódem, které podporují deep zoom pro obrázky. Také umožňuje sdílet a vkládat prohlížeč přímo

na konkrétní stránky v rámci rukopisu. Kromě toho má Universal Viewer search funkci pro vyhledávání konkrétních stránek v rukopisu a také možnost sdílet a zobrazovat audio či video, 3d a PDF obsah [21].

### 3.5.2.1 Výhody

- Snadné použití.
- Široká škála funkcí.
- Přizpůsobitelnost.
- Aktivní komunita.

### 3.5.2.2 Nevýhody

- Omezenější rozšiřitelnost.

## 3.5.3 OpenSeadragon

OpenSeadragon je vysoce výkonný image viewer, který se vyznačuje schopností zobrazovat velké obrázky s vysokým rozlišením bez ztráty detailů. Jedná se o viewer s open source kódem pro obrázky, implementovaný v čistém JavaScriptu pro počítače a mobilní zařízení. OpenSeadragon podporuje několik protokolů pro zobrazování obrázků a aktivně přidává podporu pro další. Poskytuje mnoho volitelných prvků rozhraní, včetně možnosti zoomu, otáčení, přiblížení a pohybu po obrázku, což umožňuje uživatelům detailně prozkoumávat a analyzovat zobrazené obrazy [22].

### 3.5.3.1 Výhody

- Vysoký výkon.
- Podpora velkých obrázků.
- Škálovatelnost.
- Flexibilita.

### 3.5.3.2 Nevýhody

- Méně intuitivní uživatelské rozhraní.
- Menší množství integrovaných funkcí.

## 3.5.4 Tify

Tify je *mobile-friendly* viewer dokumentů IIIF vytvořený pomocí Vue.js. Podporuje verzi 2 rozhraní IIIF Presentation API a Image API. Podpora aktuálního rozhraní API verze 3 se pro Tify chystá v blízké budoucnosti. Vyznačuje se například tím, že dokáže zobrazit libovolný počet stránek vedle sebe [23].

### 3.5.4.1 Výhody

- Jednoduché a intuitivní uživatelské rozhraní.
- Podpora IIIF a možnost integrace s dalšími aplikacemi.

### 3.5.4.2 Nevýhody

- Méně funkcí ve srovnání s některými jinými viewery.
- Chybí podpora nejnovější verze API.

## 3.5.5 IIIF Curation Viewer

IIIF Curation Viewer je image viewer kompatibilní s IIIF. Nabízí nejen základní funkce, jako je navigace po stránce a zvětšení stránky, ale také funkce kurátorství pro shromažďování částí obrázků ze světových webových stránek kompatibilních s IIIF. Software je navržen jako jednoduchý s minimem funkcí, ale je zde také možnost rozšíření pomocí plugin frameworků [24].

Kurátorství původně spočívalo ve shromažďování obsahu souvisejícího s tématem, jeho uspořádání ve vhodném pořadí a vystavení jako nového obsahu. Kurátorství je odvozené dílo, které vytváří hodnoty z nové kombinace existujících prvků, a je to také obsah řízený uživateli, který vytváří nové interpretace obsahu sdílením různých pohledů lidí. Účelem platformy IIIF Curation Platform je vnést mechanismus vytváření hodnot řízený uživateli do světa IIIF. Zásadní roli zde hraje funkce kurátorství neboli "scissors and paste"(vystříhni a vlož) [24].

### 3.5.5.1 Výhody

- Specializovaný pro zobrazování kurátorovaných sbírek obrázků.
- Podpora pro prohlížení digitálních sbírek vytvořených pomocí IIIF.

### 3.5.5.2 Nevýhody

- Možné omezení v obecné flexibilitě pro zobrazování obrázků mimo kontext kurátorovaných kolekcí.

- Omezená možnost získávání informací (původně se nejedná o anglické prostředí).

### 3.5.6 Srovnání

Mirador, Universal Viewer, OpenSeadragon, Tify a IIIF Curation Viewer jsou populární image viewery, které podporují standard IIIF, ale každý má své vlastní charakteristiky a výhody. Mirador se vyznačuje robustním rozhraním a schopností synchronizace mezi různými obrázky, což umožňuje detailní srovnání a analýzu. Universal Viewer je známý svou snadnou použitelností a širokou škálou funkcí, což ho činí vhodným pro uživatele všech úrovní stí. OpenSeadragon vyniká vysokým výkonem a schopností efektivně zobrazovat velké obrázky s vysokým rozlišením bez ztráty detailů. Tify je moderní image viewer s intuitivním rozhraním a podporou pro mobilní prohlížení. IIIF Curation Viewer je zaměřen na kurátorské práce a umožňuje uživatelům vytvářet a sdílet kurátorované kolekce obrázků. Každý z těchto viewrů má své místo a uplatnění v digitálním prostředí a volba závisí na konkrétních potřebách uživatele a charakteru projektu.

Pokud jde o univerzální použití s širokým spektrem funkcí, Mirador, Universal Viewer a OpenSeadragon zůstávají silnými volbami, zatímco Tify a IIIF Curation Viewer jsou specializovanější a mohou být vhodné pro určité typy projektů.

# Použití IIIF Frameworku

## 4

Tato kapitola se zaměřuje na praktické aplikace IIIF frameworku ve světovém měřítku. Po předchozím seznámení s technickými aspekty a historickým vývojem IIIF se nyní práce zaměří na jeho specifické využití v různých institucionálních a akademických kontextech. Důraz bude kladen na to, jak instituce přistupují k integraci IIIF do svých procesů, a na přínosy, které tato integrace přináší jak ve smyslu zvýšení kvality digitálního obsahu, tak ve zlepšení jeho přístupnosti a interaktivního využití.

Z pohledu praktické implementace se instituce rozhodují pro IIIF primárně kvůli možnosti standardizovaného sdílení a prezentace obrazových materiálů na mezinárodní úrovni. IIIF poskytuje jednotné standardy pro popis a doručování obrázků přes web, což usnadňuje interoperabilitu mezi obrazovými repozitáři a zlepšuje uživatelskou přívětivost v přístupu k digitálním sbírkám. Instituce jako Harvardovy umělecká muzea a Národní galerie umění, které již IIIF implementovaly, využívají tuto technologii pro zvýšení viditelnosti a dostupnosti svých archivů a sbírek na globální úrovni [25].

Mezi nejčastěji využívané funkce IIIF patří hluboké přiblížení, které umožňuje detailní prohlížení obrazů bez ztráty kvality a možnost porovnávání obrazů vedle sebe, což je ideální pro studijní a výzkumné účely. Tyto funkce poskytují nejen kvalitnější zobrazení detailů v uměleckých a historických artefaktech, ale také podporují interaktivitu a zapojení uživatelů při prohlížení digitálních sbírek. Instituce využívají IIIF také pro manipulaci a anotaci obrazů, což umožňuje uživatelům přidávat poznámky a další kontextové informace přímo do digitálních obrazů [25].

## 4.1 Aktuální stav využití v praxi

V této sekci je prezentována analýza, která je založena na informacích získaných přímo z komunikace se skupinou IIIF. Tyto údaje nejsou čerpány z publikovaných zdrojů, ale jsou výsledkem aktivního dialogu s expertním týmem IIIF, což umožnilo získat aktuální a praktické informace přímo od těch, kdo se podílejí na vývoji

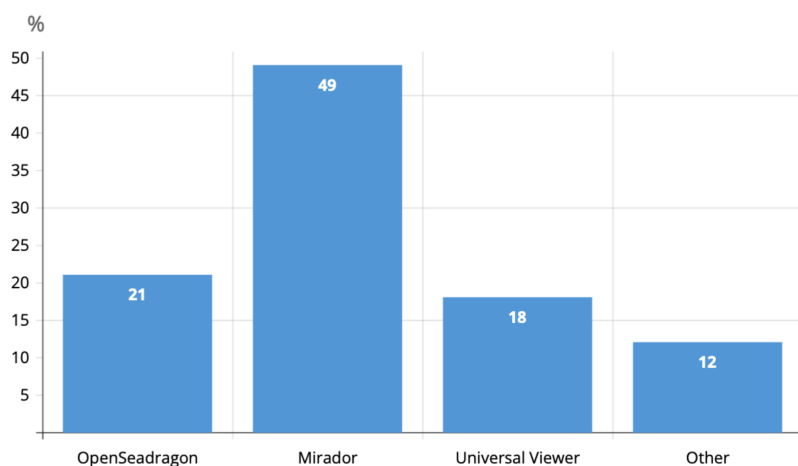
a správě IIIF standardů.

Na základě provedené analýzy nám využití IIIF v různých institucích ukazuje jeho širokou použitelnost v oblasti kulturního dědictví. Mezi tyto instituce patří muzea, galerie, univerzity a knihovny. Momentálně je na celém světě je kolem stovky uživatelů IIIF v těchto institucích [6].

### 4.1.1 Aplikace technologie a její cíle

Mnoho institucí implementuje jak Image API, tak Presentation API od IIIF, což umožňuje snadný online přístup a manipulaci s digitálními obrazy vysoké kvality a souvisejícími metadaty. k dosažení pokročilých možností prohlížení, jako je detailní přiblížení nebo srovnávání obrazů, jsou často integrovány technologie jako OpenSeadragon, Universal Viewer a Mirador. Některé instituce navíc vyvíjejí vlastní aplikace a nástroje (například ArtGIF, Art Tab, react-iiif-viewer) ke zlepšení uživatelské interakce a zapojení do jejich digitálních kolekcí.

Primární využití IIIF v různých institucích spočívá ve zpřístupňování digitálních sbírek online, což vede k lepší dostupnosti uměleckých děl, historických artefaktů a vědeckých vzorků pro širokou veřejnost a akademické výzkumníky. Několik institucí také využívá IIIF pro podporu specifických vzdělávacích a výzkumných programů, včetně použití ve výuce, výzkumu a veřejného zapojení pomocí interaktivních nástrojů.



Obrázek 4.1: Grafová ukázka procentuálního užití Image Viewers z vybraného vzorku dat (vlastní zpracování)

### 4.1.2 Spolupráce a otevřený přístup

Otevřený přístup se stává zřetelným trendem mezi kulturními institucemi. Příkladem jsou Yale University Art Gallery a Dallas Museum of Art, které aktivně sdílejí

své IIIF manifesty s externími subjekty. Toto umožňuje širší přístup k digitálním sbírkám a jejich další využití. Je kladen důraz na kolaborativní prvky, jako je možnost uživatelů vytvářet osobní sbírky nebo se podílet na výzkumech.

Instituce rovněž inovují s využitím IIIF k prozkoumání nových druhů digitálních interakcí, včetně funkce anotovaných obrazových prohlídek, začleňování do vzdělávacích platform jako je EdX či pomalého prohlížení (relaxační zážitek, který zobrazí na celé obrazovce pomalu a detailně vybraný obrázek, který si uživatel může bez rozptylování prohlédnout podobně jako v galerii).

Přijímání nejnovějších verzí IIIF API, jako je Image API verze 3, ze strany institucí, například Přírodovědného muzea, demonstruje trend využívání nejnovějších technologických inovací pro zlepšení výkonu a funkčnosti.

Dle dostupných informací je možné konstatovat, že instituce jako umělecká muzea Harvard a Národní galerie umění demonstrují aktivní využívání technologií IIIF pro rozšiřování a inovaci svých digitálních sbírek. Jejich aktualizace a účast na komunitních diskuzích, jako je například diskusní skupina Museums Community call, kde se účastnilo Dallas Museum of Art, svědčí o aktivním zapojení a závazku k rozvoji standardů IIIF.

### 4.1.3 Geografie

Rozšíření technologie IIIF v Evropě a Severní Americe je výrazné, s množstvím přijímajících institucí v hlavních městech. Tento fakt podtrhuje silné angažmá v oblasti digitálních archivačních technologií v uvedených regionech. Zdá se, že tyto geografické oblasti představují špičku v používání IIIF, což může být podpořeno jak dostupností technologických zdrojů, tak dlouhodobým závazkem k digitalizaci kulturního dědictví.

Naopak v Asii, Oceánii a dalších částech světa je zapojení institucí méně rozšířené, ale i zde je patrný rostoucí zájem a postupná adopce IIIF. Tento trend může odrážet různé faktory, včetně dostupnosti technologií a iniciativ na digitalizaci kulturního dědictví. Narůstající zájem o IIIF v těchto regionech naznačuje, že i zde se postupně uznává důležitost a přínosy IIIF pro zpřístupnění a mezinárodní sdílení kulturních aktiv.

## 4.2 Výzvy a omezení v aplikaci frameworku

Implementace IIIF vyžaduje technologické znalosti, které nemusí být vždy snadno dostupné všem institucím. Například při projektu GallicaPix, realizovaném Bibliothèque Nationale de France, bylo nutné vyvinout a implementovat sofistikované techniky strojového učení pro kategorizaci a vyhledávání obrazů. Tento projekt využíval technologie jako IBM Watson a Google Cloud Vision pro analýzu a klasifikaci



Obrázek 4.2: Zobrazení využití IIF ve světě [6]

obrazového obsahu, což představuje vysoké nároky na technologickou infrastrukturu a odborné znalosti [26].

Další výzvou pro instituce je zajištění interoperability dat mezi různými platformami a systémy. IIF sice usnadňuje sdílení a přístup k digitálním materiálům, ale často se setkává s problémy v harmonizaci metadat. Například v rámci OCLC bylo potřebné provést rozsáhlou práci na konverzi dat a workflow, aby bylo možné plně využít výhod IIF pro sdílení a vyhledávání v digitálních sbírkách [27].

Implementace a udržování IIF může být mimo jiné i finančně náročné. Příkladem může být nutnost investovat do vývoje vlastních aplikací a nástrojů pro práci s IIF, jak to činí mnoho větších institucí. Kromě počátečních investic jsou tu také náklady spojené s pravidelnými aktualizacemi a údržbou systémů, aby byly kompatibilní s nejnovějšími standardy IIF.

## 4.3 Analýza oblastí využití IIF u obrazů Rudolfa II.

Na základě provedené analýzy lze pozorovat, že řada významných světových institucí implementuje IIF také pro obrazy Rudolfa II. v kontextu děl Rudolfa II. se dokázalo identifikovat několik příkladů institucí, které tuto technologii využívají. Patří mezi ně například Art Institute Chicago, Cambridge či Nationalmuseum Stockholm. Tyto instituce nejen že poskytují přístup k IIF manifestům, ale také nabízejí možnosti prohlížení, porovnávání a detailní studium uměleckých děl prostřednictvím moderních webových viewerů, jako je Mirador nebo OpenSeadragon.



Některé instituce mají však IIIF manifesty skryté nebo nejsou přímo dostupné bez pokročilé znalosti API, což může být limitující pro běžné uživatele. To naznačuje potřebu lepší integrace a transparentnosti v poskytování přístupu k digitálním sbírkám.

Celkově lze říci, že implementace IIIF poskytuje robustní základ pro digitalizaci a sdílení kulturního dědictví Rudolfa II. na mezinárodní úrovni, což umožňuje vědcům, studentům a veřejnosti ponořit se do detailů uměleckých děl bez nutnosti fyzické přítomnosti v galerii. Tento přístup je zvláště cenný pro studium a ochranu uměleckých děl.

# Analýza možností využití pro Inventarii

## 5

Cílem této kapitoly je analýza možností, jak by se IIF framework mohl využít v rámci projektu *Inventaria Rudolphina*. Práce provede analýzu různých přístupů a strategií, které by mohly optimalizovat správu, přístupnost a interaktivitu digitálních sbírek. Tato úvaha bude sloužit jako příprava pro další kapitolu, kde budou tyto možnosti dále rozvedeny a budou navržena konkrétní řešení pro efektivní využití IIF v praxi Inventaria.

## 5.1 Využití IIF manifestů

První zvažovanou strategií je identifikace a distribuce již existujících IIF manifestů obrazů Rudolfa II. v rámci Inventaria. Tato metoda však přináší dva zásadní problémy. Prvním je závislost na externích serverech, které hostují tyto manifesty. Vzhledem k tomu, že tyto servery jsou nezávislé a mohou kdykoliv rozhodnout přestat manifesty sdílet, vystavuje to Inventarii riziku ztráty přístupu k digitálním dílům. Druhým problémem je možná ztráta výzkumné práce a vkladů výzkumníků Akademie věd, které byly investovány do analýzy a anotace specifických děl. Tento přístup by mohl vést ke ztrátě unikátních poznámek a výzkumných dat spojených s jednotlivými obrazy.

Alternativním řešením, které by mohlo zabránit těmto rizikům, je vytvoření a hostování vlastních IIF manifestů pro všechna díla spravovaná Inventarií. Tím by se Inventaria stala nezávislým poskytovatelem manifestů, které by mohly být následně sdíleny dalšími výzkumníky. Tento přístup by nejen zajišťoval stabilitu a dostupnost digitálních děl, ale také umožnil zachování a integraci výzkumné práce do samotných manifestů. Tato strategie a její implementace bude dále rozebírána v sekci 6.1.

## 5.2 Poskytování obrazů na IIIF kompatibilním serveru

Dále by se dalo uvažovat o poskytování obrazů s využitím IIIF serverů, což by Inventariu umožnilo efektivněji sdílet a přistupovat k digitálním sbírkám. Současná infrastruktura Inventaria zahrnuje vlastní server postavený na frameworku Node.js, kde jsou obrázky přístupné prostřednictvím následujícího URI:

```
https://www.inventariarudolphina.com/static/images/obr.jpg
```

Toto řešení však neumožňuje využití pokročilých funkcí IIIF Image API, jako je hluboké přiblížení a další interaktivní vlastnosti. Přestože je možné implementovat IIIF Presentation API, které by umožnilo efektivní správu metadat a struktury prezentace, stávající URI by bylo použito beze změny, což by nevedlo ke zlepšení funkcionalit spojených s obrazovým materiálem.

Nový přístup by zahrnoval buď využití externího hostingu, nebo nastavení vlastního serveru, který by byl plně kompatibilní s IIIF. To znamená, že by podporoval IIIF Image API. Toto řešení by umožnilo plné využití vytvořených manifestů, protože každý obrázek by měl své vlastní URL, což by umožnilo jeho zobrazování v různých IIIF image viewerech. Externí IIIF servery nabízí pokročilé funkce jako hluboké zoomy a porovnávání obrazů, což zvyšuje interaktivitu a uživatelskou přívětivost digitálních sbírek. Nicméně, závislost na externím poskytovateli může přinést rizika týkající se dostupnosti a kontroly nad daty, což vyžaduje pečlivé zvážení bezpečnostních opatření a výběr spolehlivého poskytovatele.

Pro implementaci IIIF se nabízí tři strategické možnosti. První možností je aktualizace stávajícího serveru tak, aby byl kompatibilní s IIIF. Alternativou je vytvoření nového serveru již s předinstalovanou podporou pro IIIF. Třetí možností je využití externího hostingu na infrastruktuře, která podporuje IIIF.

Zvažování externího hostingu na infrastruktuře podporující IIIF nabízí několik potenciálních výhod, jako je snížení zátěže na interní zdroje a možnost využití specializovaných služeb, které mohou nabídnout lepší výkon a škálovatelnost. Toto řešení však přináší i značné nevýhody, především závislost na třetí straně, což může ovlivnit kontrolu a dostupnost dat, a potenciální bezpečnostní rizika spojená s hostováním dat mimo interní bezpečné prostředí. Vzhledem k těmto faktorům a omezením se zdá, že tento přístup momentálně pro projekt Inventaria není vhodný.

Vytvoření nového serveru speciálně navrženého pro podporu IIIF, jako je například Cantaloupe [28], by přineslo výhody jako optimalizace pro manipulaci s obrazovými daty a integraci pokročilých IIIF funkcí. Nicméně, zavedení nové technologie jako je Java, kterou Cantaloupe vyžaduje, by mohlo znamenat významné technologické a organizační změny. To by mohlo zkomplikovat integraci s existujícími

základy. Přestože je toto řešení zajímavé pro budoucí rozšíření a možné technologické inovace, v současné době může představovat příliš velkou změnu.

Na základě analýzy možností a současných technologických zdrojů se jeví jako optimální volba integrace IIIF přímo do stávajícího serveru postaveného na Node . js. Tento přístup minimalizuje potřebu zásadních změn v infrastruktuře a využívá stávající technologické zdroje. Integrace IIIF do existujícího systému umožní efektivně zavést pokročilé funkce pro práci s obrazovými sbírkami.

Vzhledem k tomu, že pro výzkumníky Inventaria je prioritou spíše přehledné ukládání a zobrazování metadat než samotné obrázkové funkce, bude toto řešení v sekci 6.2 navrženo, avšak v rámci této diplomové práce nebude implementováno.

## 5.3 Integrace Image vieweru do Inventarie

Další možností, jak rozšířit funkčnost a dostupnost digitálních sbírek Inventaria, je integrace vybraného IIIF image vieweru přímo do webových stránek. Tento krok by zahrnoval výběr a implementaci jednoho z dostupných IIIF viewerů, jako je například Mirador nebo Universal Viewer, které jsou plně kompatibilní s IIIF specifikacemi a nabízejí uživatelům rozsáhlé možnosti pro interaktivní prohlížení obrazů. Implementace by zahrnovala specifické technické změny na webu. v závislosti na stávajícím technologickém stacku webu může být také potřeba provést úpravy backendu, například pro správu manifestů nebo optimalizaci ukládání a načítání obrazových dat.

Integrace IIIF vieweru do webu by s sebou přinesla významné výhody. Zvýšila by user experience (uživatelskou zkušenost) tím, že by poskytla návštěvníkům pokročilé nástroje pro vizuální exploraci a analýzu obrazů, což je zvláště cenné v akademickém a výzkumném kontextu. Dále by to umožnilo Inventariu lépe prezentovat své digitální sbírky a zvýšit jejich viditelnost a dostupnost na mezinárodní úrovni. na druhou stranu, integrace IIIF vieweru vyžaduje počáteční technické úsilí pro implementaci a testování. Dále je zde riziko technických problémů, které mohou ovlivnit výkon webu, jako jsou zvýšené nároky na server při načítání velkých obrazových souborů nebo potenciální bezpečnostní rizika. v rámci této práce bude tato možnost v sekci 6.3 navržena, ale nebude implementována.

Prozkoumané možnosti využití IIIF frameworku pro Inventarii budou podrobněji rozpracovány v následující kapitole. Zde se práce zaměří na formulaci konkrétních návrhů a strategií, které by měly umožnit efektivní implementaci IIIF, a tím zlepšit správu, dostupnost a interaktivitu digitálních sbírek Inventaria.

# Návrh pro vybrané případy

## 6

Tato kapitola se bude věnovat návrhům vybraným případům na základě předchozí analýzy. Cílem bude navrhnout praktická řešení třech témat. První z nich se bude věnovat návrhu skriptu pro transformaci stávající JSON struktury díla do struktury IIIF manifestu.

Druhá sekce se bude věnovat návrhu poskytování obrázků na IIIF kompatibilním serveru. Budou rozebírány možnosti a navrhnut bude případ, který nejlépe sedí do momentálního řešení Inventaria.

Poslední návrh se bude věnovat výběru a integraci image vieweru do stávajícího řešení Inventaria.

## 6.1 Vytvoření IIIF manifestů pro díla

Tato sekce se bude věnovat návrhu skriptu na transformaci struktury stávající položky do struktury manifestu vyhovující standardům IIIF. Návrh bude obsahovat motivaci, omezení stávajícího řešení a návrh implementace společně s validací a otestováním návrhu.

### 6.1.1 Motivace návrhu

V následujících bodech je shrnuta klíčová motivace a cíle pro vytvoření vlastních IIIF manifestů, které budou sloužit nejen k zajištění lepší kontroly nad daty, ale také k jejich efektivnějšímu využití a sdílení v akademické komunitě i širší veřejnosti:

- **Zlepšení kontroly nad daty:** Vlastní IIIF manifesty umožní Inventariu plnou kontrolu nad tím, jaká metadata a obrazový obsah budou zveřejněny. Tímto přístupem se eliminuje riziko ztráty dat v důsledku nespolehlivosti externích poskytovatelů a zajišťuje se, že veškerý obsah je prezentován v souladu s interními směrnici a politikami.
- **Přizpůsobení systému potřebám uživatelů:** Implementací funkcí, jako jsou anotace a odkazy na další relevantní zdroje, se systém stane flexibilnějším

a přívětivějším pro uživatele. Tyto funkce umožní uživatelům lépe interagovat s digitálním obsahem, což zvýší hodnotu a užitečnost sbírek.

- **Integrace s MongoDB pro dynamické generování manifestů:** Využití MongoDB databáze pro dynamické generování IIIF manifestů zajišťuje, že všechny informace jsou aktuální a správné. Tento proces nejenže zefektivňuje správu dat, ale také usnadňuje jejich aktualizaci a rozšíření.
- **Zachování vlastního výzkumu a dat:** Vytvoření vlastních manifestů umožní Inventariu zachovat a prezentovat vlastní poznámky a výzkumné práce spojené s jednotlivými díly. Tímto způsobem lze uchovat originální interpretace a hlubší kontextové informace, které jsou nepostradatelné pro akademické a vzdělávací účely.
- **Zlepšení uživatelského zážitku při prohlížení:** s IIIF kompatibilními prohlížeči se výraznělepší vizuální prezentace a interaktivita s digitálními sbírkami. Uživatelé budou mít přístup k pokročilým funkcím prohlížení, což výrazně obohacuje jejich zkušenost s obsahem.

Tento přístup nejen že podporuje dlouhodobou udržitelnost a rozvoj Inventaria jako nezávislého poskytovatele kulturního obsahu, ale také napomáhá k posílení jeho role v mezinárodní akademické a vědecké komunitě.

## 6.1.2 Omezení stávajícího řešení

V současném technologickém nastavení platformy Inventaria se objevují několik významných omezení, která komplikují efektivní využívání a sdílení digitálních sbírek. Tyto nedostatky značně snižují možnosti vzdělávacího, výzkumného a veřejného přínosu digitálního obsahu. Detailní rozbor současných omezení je následující:

- **Omezený přístup a izolace dat:** Ačkoli jsou data přístupná ve formátu xml přímo v rámci aplikace, je jejich struktura specificky navržena pro potřeby projektu *Inventaria Rudolphina* a není bez dalších úprav kompatibilní s jinými systémy nebo platformami. Tato unikátnost způsobuje, že i přestože jsou data technicky dostupná, jsou ve skutečnosti izolovaná a nemohou být snadno zobrazena ani využita mimo primární aplikaci. Tento nedostatek značně omezuje možnosti jejich sdílení a integrace s digitálním obsahem v širším kontextu.
- **Omezené možnosti interakce s obrazy:** v současném systému nelze s digitálními obrazy efektivně manipulovat. Chybí možnosti jako hluboký zoom, rotace nebo porovnání různých obrazů vedle sebe. Tyto funkce jsou klíčové pro podrobnou vědeckou analýzu a vzdělávací účely, jejich absence tedy snižuje uživatelský zážitek a funkční hodnotu digitálních sbírek.

Tyto nedostatky společně vyvolávají potřebu přechodu na robustnější a flexibilnější technologickou infrastrukturu, která umožní Inventariu plně využívat potenciál svých digitálních sbírek. Implementace IIIF a vytvoření vlastních manifestů představují klíčový krok k překonání těchto omezení a zvýšení dostupnosti, interaktivity a sdílenosti digitálních obrazových dat.

## 6.1.3 Implementace

Následující sekce bude popisovat návrh implementace transformace stávající struktury *Inventaria Rudolphina* do struktury manifestů kompatibilní s IIIF. Návrh bude zahrnovat výběr technologie pro extrakci dat z databáze, postup této extrakce, návrh pro transformaci dat ze stávající JSON struktury do struktury manifestu a v neposlední řadě návrh řádného otestování.

### 6.1.3.1 Výběr technologie pro extrakci dat z databáze

Extrakce dat je prvotním krokem v procesu transformace dat uložených v MongoDB pro projekt *Inventaria* do IIIF manifestů. Pro správu a manipulaci s daty v MongoDB byly zvažovány různé nástroje, včetně MongoDB Shell, Studio 3T, a MongoDB Compass.

MongoDB Shell, přestože je výkonným nástrojem umožňujícím provádět širokou škálu databázových operací z příkazové řádky, byl pro tento projekt považován za méně vhodný kvůli jeho vyšší složitosti a požadavku na hluboké znalosti MongoDB Query Language. Tento nástroj také neposkytuje žádnou vizuální podporu, což může ztížit proces analýzy a selekce dat [29].

Studio 3T, které je dalším grafickým nástrojem pro správu MongoDB, nabízí uživatelsky přívětivé rozhraní a základní funkce pro procházení a editaci dat. Avšak oproti MongoDB Compass, Studio 3T nabízí méně pokročilé analytické nástroje a může být omezen ve funkcích potřebných pro komplexní datovou analýzu a optimalizaci výkonu [30] [31].

MongoDB Compass byl nakonec vybrán jako nejvhodnější nástroj. Jeho silnou stránkou je intuitivní grafické uživatelské rozhraní, které nejen zjednodušuje procházení a správu dat, ale také poskytuje vizualizace, které pomáhají lépe pochopit strukturu a vztahy dat. Compass nabízí také rozsáhlé možnosti pro analýzu výkonu a monitorování databáze, což je nezbytné pro udržení vysokého výkonu a bezpečnosti při manipulaci s daty. Tyto vlastnosti činí MongoDB Compass ideálním nástrojem pro efektivní a bezpečnou extrakci dat, která je nezbytná pro další kroky v procesu vytváření IIIF manifestů.

### 6.1.3.2 Teoretický postup extrakce dat

V procesu transformace dat z MongoDB do IIIF manifestů je potřeba efektivní extrakce dat z kolekce `inventory`. Tato kolekce obsahuje komplexní informace o uměleckých dílech, včetně obrázků a doplňujících metadat.

Po připojení k MongoDB pomocí Compassu se data z vybrané kolekce extrahují jako kompletní JSON dokumenty. Tento přístup umožňuje získat plný přehled o struktuře a obsahu dat, což je zásadní pro další zpracování. Následně se využívá programovací jazyk Python společně s knihovnou `pymongo`, která slouží k manipulaci s načtenými daty. Python byl zvolen kvůli jeho flexibilitě, rozsáhlé podpoře knihoven a snadné integraci s různými datovými formáty a webovými technologiemi, což je ideální pro následnou práci s IIIF.

Detailní selekce konkrétních atributů z dokumentů pro transformaci do IIIF manifestů a specifikace, jak budou tyto informace v manifestech reprezentovány, bude probíhat v následující sekci práce. Tento postup zajišťuje systematické a metodické zpracování dat, od extrakce po transformaci, což umožňuje zachování kontextu a maximální využití dostupných informací pro generování kvalitních digitálních manifestů.

### 6.1.3.3 Návrh transformace dat do IIIF manifestů

Proces transformace dat z MongoDB do IIIF manifestů začíná extrakcí potřebných dat z kolekce `inventory`. Tato kolekce obsahuje klíčová data, která je třeba zpracovat pro vytvoření strukturovaných a informativních digitálních manifestů. Pomocí MongoDB Compass jsou z databáze extrahovány kompletní JSON dokumenty. Tyto dokumenty obsahují široký rozsah atributů, z nichž některé jsou pro tento projekt nezbytné, zatímco jiné mohou být redundantní.

V rámci návrhu je prioritou selektivní extrakce atributů, které jsou klíčové pro reprezentaci uměleckých děl. z kolekce `inventory` se vybírají atributy jako `xml_id`, `inventory`, `author_name`, `title` a další, které poskytují základní informace o každém díle. Následně extrahuje informace o obrázku, pokud dokument nějaký obsahuje.

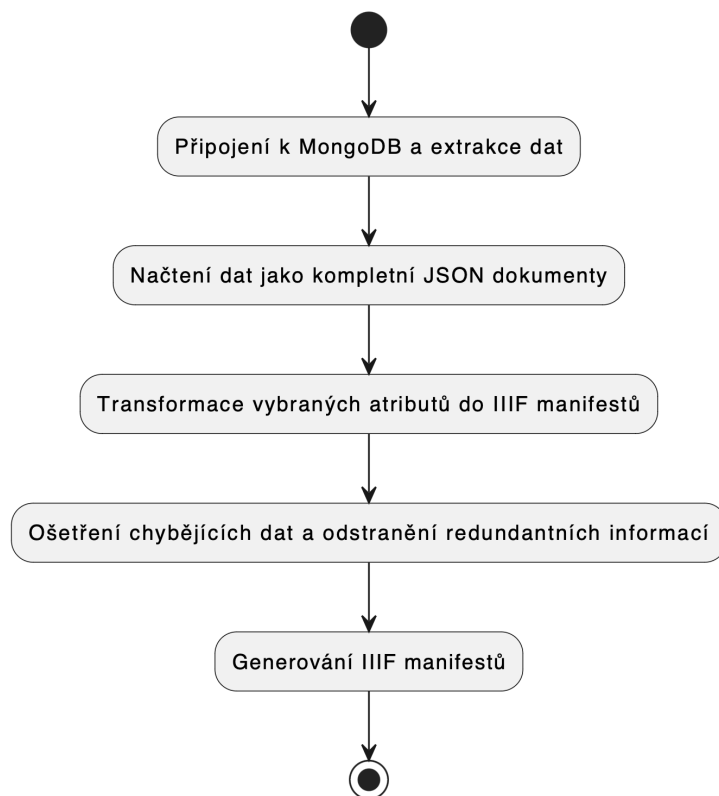
Pro zpracování extrahovaných dat je použit Python, což umožňuje efektivní manipulaci s JSON strukturami a automatizaci procesu vytváření manifestů. Skript zajišťuje, že všechny vybrané atributy jsou transformovány do formátu IIIF manifestů, přičemž se dbá na správné zacházení s chybějícími daty. v případech, kdy některé atributy nejsou přítomny, jsou tyto informace přeskočeny, což zajišťuje integritu a konzistenci výsledných manifestů. Pro tento účel bude využita python knihovna `json`.

Každý IIIF manifest je pak strukturován tak, aby obsahoval metadata o díle a jednotlivé obrázky reprezentované jako plátna v manifestu.



Celkově tento návrh plánu představuje detailní přístup k transformaci dat do IIIF manifestů, který zohledňuje jak technické aspekty tak potřeby uživatelů, a zajišťuje vytvoření kvalitních digitálních reprezentací uměleckých děl.

#### Procesní Diagram: Transformace Dat z MongoDB do IIIF Manifestů



Obrázek 6.1: Procesní diagram transformace

### 6.1.3.4 Návrh validace vytvořených IIIF manifestů

Validace výsledných IIIF manifestů je nezbytným krokem k zajištění jejich správnosti a kompatibility s IIIF Presentation API. Pro tento účel bude využit validátor dostupný na oficiálních stránkách IIIF, konkrétně na adrese IIIF validátoru. Tento nástroj poskytuje prostředí, ve kterém lze jednoduše ověřit, zda jsou manifesty vytvořené v souladu s definovanými standardy IIIF.

Proces validace zahrnuje nahrání manifestu do validátoru, který analyzuje jeho strukturu a obsah. Validátor následně buď potvrdí, že manifest je bez chyb a plně odpovídá specifikacím IIIF, nebo identifikuje a reportuje případné chyby nebo nedostatky v manifestu. Tyto informace jsou klíčové pro rychlou identifikaci problémových oblastí a umožňují cílenou optimalizaci skriptů, aby výstupem byly validní manifesty.

Dalším krokem v procesu ověřování funkčnosti a správnosti manifestů je jejich testování ve vizualizačních nástrojích. Výsledné manifesty budou v rámci testování publikovány na veřejně přístupném serveru, jako je GitHub, což umožní jejich snadné sdílení a přístup. Následně bude pro zobrazení a interaktivní prohlížení děl využít Universal Viewer nebo jiné IIIF kompatibilní prohlížeče. Tento přístup umožní nejen vizuálně ověřit, že manifesty fungují jak bylo zamýšleno, ale také zjistit, jak jsou data prezentována koncovým uživatelům.

Tento návrh na validaci a testování IIIF manifestů zajišťuje, že všechny vytvořené manifesty budou nejen technicky správné, ale také uživatelsky funkční. Uvedený přístup je nezbytný pro úspěšné nasazení digitálních sbírek v souladu s mezinárodními standardy, což zvyšuje jejich dostupnost a využitelnost pro širokou veřejnost a odbornou komunitu.

## 6.2 Integrace IIIF do aplikačního serveru

V této sekci se práce zaměří na návrh integrace IIIF do stávajícího serverového řešení projektu Inventaria. Aktualizace serveru o podporu IIIF představuje důležitý krok k rozšíření schopností v oblasti sdílení a interakce s digitálními obrazovými sbírkami. Tento návrh nejenže umožní efektivnější využití existujících obrazových dat, ale také otevře nové možnosti pro jejich prezentaci a analýzu v souladu s IIIF standardem.

### 6.2.1 Motivace návrhu

Klíčová motivace a cíle pro integraci IIIF do stávajícího serveru projektu Inventaria jsou následující:

- **Zlepšení přístupnosti a interaktivnosti:** Implementace IIIF umožní Inventariu poskytnout pokročilé obrazové funkce, jako jsou hluboké zoomy a dynamické manipulace s obrázky, což zvýší interaktivitu a uživatelskou přívětivost digitálních sbírek.
- **Integrace a kompatibilita s globálními standardy:** Použitím IIIF se Inventaria stane součástí globální komunity, která sdílí a využívá standardizované obrazové protokoly, což usnadní výměnu dat a integraci s jinými institucemi a platformami.
- **Flexibilita a rozšiřitelnost:** Díky modularitě IIIF je možné snadno přidávat nové knihovny a rozšiřovat systém podle potřeb uživatelů bez zásadních změn v základní infrastruktuře.

- **Efektivnější správa digitálních sbírek:** IIIF umožní efektivnější správu obrazových a metadata sbírek, což zlepší možnosti archivace, vyhledávání a sdílení informací.
- **Zvýšení kvality digitálního obsahu:** s IIIF lze dosáhnout vyšší kvality obrazové prezentace, což je klíčové pro akademické a výzkumné účely, kde je detail a vizuální přesnost nezbytná.

## 6.2.2 Omezení stávajícího řešení

Současná infrastruktura Inventaria se setkává s několika technologickými omezeními, která komplikují efektivní využití a sdílení digitálních sbírek. Detailní rozbor těchto omezení zahrnuje:

- **Omezené funkce pro manipulaci s obrazy:** Stávající systém nepodporuje pokročilé obrazové funkce, což omezuje možnosti využití digitálních sbírek pro vzdělávací a výzkumné účely.
- **Nekompatibilita s mezinárodními standardy:** Absence podpory pro IIIF ztěžuje integraci s jinými digitálními knihovnami a archivy, což izoluje Inventarii od mezinárodní akademické a vědecké komunity v rámci IIIF.
- **Omezená škálovatelnost a rozšiřitelnost:** Bez podpory pro modulární a rozšiřitelné standardy jako IIIF je obtížné adaptovat systém na měnící se technologické a uživatelské požadavky.

## 6.2.3 Update aktuálního serveru o podporu IIIF

IIIF poskytuje různé možnosti integrace do existujících systémů, přičemž klíčovým rozhodnutím je, zda implementovat podporu pouze pro Image API, nebo zahrnout i Presentation API. Image API umožňuje klientům požadovat obrázky v určitých rozměrech a formátech a provádět další specifické transformace, což vyžaduje, aby byly manifesty vytvářeny vlastním procesem. Vzhledem k tomu, že již bude vyvinut program pro transformaci manifestů ze stávajících JSON reprezentací položek, okamžitá potřeba podpory Presentation API není zásadní. Nicméně, s přihlédnutím k budoucímu rozšiřování sbírky a přidávání nových položek, se zdá být implementace Presentation API jako strategická výhoda, například pomocí knihovny Shimmy.

### 6.2.3.1 Návrh implementace Image API

Pro implementaci Image API byla vybrána knihovna `iiif-processor` [32]. Tato volba byla učiněna z několika klíčových důvodů. Primárně, `iiif-processor` je specificky navržen pro Node.js, což je technologické prostředí, ve kterém je již stávající

server Inventaria postaven. Tato kompatibilita zjednodušuje integraci a minimalizuje potřebu zásadních změn v architektuře nebo přidání dalších technologií, což by mohlo znamenat vyšší náklady a složitější údržbu. Knihovna poskytuje kompletní sadu funkcí potřebných pro zpracování a servírování obrazů podle specifikací IIIF, což je klíčové pro zlepšení přístupnosti a interaktivity digitálních sbírek v rámci Inventaria.

#### Instalace:

Tato knihovna se instaluje prostřednictvím npm (Node Package Manager) [33], což je efektivní a rychlý způsob, jak přidat nové funkce do Node.js aplikací. `iiif processor` poskytuje robustní API pro zpracování obrazových požadavků dle specifikací IIIF. Příkazem `npm install iiif-processor` je knihovna přidána do projektu a připravena k další konfiguraci a integraci.

#### Konfigurace:

Dále je potřeba konfigurace `stream resolver` [34]. Jedná se o funkci zpětného volání, která umožňuje knihovně načítat obrazová data. `Stream resolver` by měl vracet čitelný stream obrazu z úložiště obrazů na základě identifikátoru obrazu a základní URL. Další užitečnou funkcí, kterou lze `iiif-processoru` poskytnout, je `dimensionFunction` [35]. Tato funkce vrací rozměry zdrojového obrazu, což umožňuje procesoru optimalizovat zpracování požadavků. Příklad správné implementace je na stránce `npm.js`.

Po nastavení `streamResolver` a `dimensionFunction` lze vytvořit instanci `iiif processor` a použít ji k zpracování IIIF požadavků na obrazy.

Více detailních informací ke konfiguraci knihovny je na stránce `npm.js`.

Když je `iiif-processor` plně nakonfigurován, URL pro přístup k obrazům se generují na základě specifikace IIIF Image API. Každé URL je strukturováno tak, aby odráželo žádanou operaci, například ořez, změnu velikosti, rotaci, kvalitu a formát obrázku.

```
1 https://{ baseUrl }/ iiif /{ version }/{ identifier }/{ region }
2 /{ size }/{ rotation }/{ quality }.{ format }
```

Zdrojový kód 6.1: Struktura výsledného URI

### 6.2.3.2 Návrh testování

V rámci integrace řešení je nezbytné provést důkladné testování jak serverové, tak klientské části systému tak, aby řešení bude stabilní a funkční.

Jako první jsou nezbytné funkční testy pro ověření, že všechny klíčové aspekty aplikace pracují podle specifikací. Tyto testy se zaměřují primárně na serverovou část, kde se nachází implementace IIIF Image API. Cílem je ověřit, že server správně zpracovává a odpovídá na požadavky dle IIIF specifikací. Testování by mělo zahrnovat vyvolání různých HTTP požadavků na API a ověření odpovědí, jak z hlediska

správnosti dat, tak i struktury a formátu odpovědí. Funkční testy lze automatizovat pomocí nástrojů jako Postman nebo skriptů napsaných v jazyce Python nebo JavaScript, které systematicky testují různé endpointy a parametry.

Další klíčovou oblastí testování je parametrizace, která ověřuje funkčnost IIIF Image API. IIIF API umožňuje uživatelům manipulovat s obrázky prostřednictvím různých parametrů, viz kapitola 3.3. Testování parametrizace se zaměřuje na to, zda server korektně interpretuje a aplikuje tyto parametry na požadované obrázky. Testování může například zahrnovat generování URL s různými kombinacemi parametrů a ověření, že obrázky jsou vráceny ve správné velikosti, výřezu, rotaci a kvalitě. Automatizované testy pomocí skriptů nebo nástrojů jako Selenium mohou simulovat tyto požadavky a porovnávat výsledky s očekávanými výstupy.

V neposlední řadě je na místě provést uživatelské testování pro měření uživatelské spokojenosti a intuitivnosti aplikace. Testy by měly zahrnovat realistické scénáře používání, které odrážejí běžné úkony, jako třeba přibližování. Cílem je identifikovat, jak snadno a efektivně mohou uživatelé interagovat s digitálními obrazy, zahrnující akce jako je zvětšování, posouvání a prohlížení různých oblastí obrázků. Důležité je sledovat, jak uživatelé navigují v systému, jaké problémy při používání nastávají a jak rychle a přesně mohou dosáhnout požadovaných výsledků.

Pro toto testování lze využít například online verzi Universal Vieweru.

## 6.3 Návrh integrace Image Vieweru do Inventaria

Tato sekce se bude věnovat návrhu integrace IIIF image vieweru do webové platformy Inventaria. Cílem je zvolit vhodný image viewer a popsat jeho implementaci do webové aplikace, která je vyvíjena s využitím technologie Vue.js. Integrace image vieweru přinese významná vylepšení v interaktivitě a přístupnosti digitálních sbírek, což umožní uživatelům hlubší a smysluplnější interakci s obrazovým materiálem.

### 6.3.1 Motivace návrhu

V následujících bodech je shrnuta klíčová motivace a cíle pro integraci IIIF image vieweru do webové platformy Inventaria, které zvýší interaktivitu a přístupnost digitálních sbírek:

- **Zvýšení uživatelské přívětivosti:** Integrace IIIF image vieweru umožní uživatelům využívat pokročilé funkce pro interaktivní prohlížení, jako jsou hluboké zoomy a detailní vizuální analýzy, což zlepší celkovou uživatelskou přívětivost.

- **Podpora akademického výzkumu:** Viewer poskytne nástroje pro efektivní vizuální exploraci a analýzu a umožní lepší interpretaci a porozumění obrazovému materiálu.
- **Mezinárodní kompatibilita:** Použitím IIIF image viewerem se Inventaria zařadí mezi globální digitální knihovny, což zvýší jeho viditelnost a dostupnost na mezinárodní úrovni.

### 6.3.2 Omezení stávajícího řešení

Současná infrastruktura Inventaria se potýká s několika omezeními, která brání v plném využívání digitálních obrazových sbírek. Tyto nedostatky jsou:

- **Omezené funkce prohlížení:** v současném systému nejsou dostupné pokročilé nástroje pro manipulaci s obrazy, což znamená, že uživatelé nemohou provádět operace.
- **Chybějící interaktivita:** Nedostatek interaktivního image vieweru znamená, že uživatelé nemohou efektivně porovnávat obrazy.
- **Snížená uživatelská zkušenost:** Absence specializovaného image vieweru vede k omezenému vizuálnímu zážitku, což může odradit uživatele od hlubšího prozkoumání a studia digitálních sbírek.

### 6.3.3 Výběr image vieweru

Tato sekce se zaměřuje na návrh integrace Image Vieweru do webové platformy Inventaria. Cílem je zvolit vhodný image viewer a popsat návrh jeho implementace do webové aplikace, která je vyvíjena s využitím technologie Vue.js. Proces výběru správného vieweru je klíčový pro zajištění, že aplikace bude schopna efektivně zpracovávat a prezentovat digitální obrazový materiál. Následně bude rozebrán návrh implementace, který zajistí hladkou integraci vybraného řešení do stávající infrastruktury Inventarie.

Při výběru vhodného image vieweru pro Inventaria byla zvažena řada možností, včetně vieweru Tify, který je napsán ve Vue.js a na první pohled se jevil jako ideální kandidát pro snadnou integraci. Nicméně, Tify v současné době nepodporuje Presentation API 3.0, což je standard, který bude v této práci použit pro vývoj skriptu na výrobu manifestů. Vzhledem k tomu, že je cílem vyvinout řešení kompatibilní s nejnovějšími specifikacemi IIIF, nedostatek podpory pro Presentation API 3.0 vede k vyřazení Tify z dalšího zvážení.

Po důkladném posouzení různých možností byl jako nejvhodnější viewer pro Inventaria vybrán Mirador. Mirador je široce uznávaný pro svou výkonnost a flexibilitu v akademickém a výzkumném prostředí. Nabízí pokročilé funkce pro správu

více canvasů, což umožňuje efektivní zobrazení a porovnávání více různých obrazových materiálů v rámci jednoho manifestu. Mirador kromě toho exceluje také v podpoře pro anotace a metadata, což umožňuje zobrazení rozsáhlých informací, jako jsou poznámky a další kontextové údaje. Mirador také umožňuje ukládat anotace nad obrázky.

### 6.3.4 Stažení balíčku Mirador

Jako první bude potřeba stáhnout Mirador, který je dostupný jako npm balíček, což usnadňuje jeho instalaci a správu. Instalace knihovny se provádí pomocí následujícího příkazu v terminálu v projektovém adresáři:

```
npm install mirador --save
```

### 6.3.5 Vytvoření komponenty

Základem integrace prohlížeče do Inventarie je vytvoření dedikované Vue . js komponenty `MiradorViewer`, která bude sloužit jako kontejner pro Mirador prohlížeč. Komponenta bude definována jednoduchým `<template>` tagem obsahujícím `<div>` element, který funguje jako cílový DOM (Document Object Model) prvek pro inicializaci Miradoru. Tato modularita umožňuje flexibilní použití komponenty v různých částech aplikace bez nutnosti duplikace kódu.

Ve skriptové části komponenty bude importována knihovna Mirador a využívá se životního cyklu `mounted` [36] pro inicializaci prohlížeče s konfigurací, která určuje, jaký manifest má být načten. Tento životní cyklus se spustí poté, co je komponenta připojena k DOM webové stránky, což znamená, že všechny elementy komponenty jsou již plně vykreslené a dostupné pro manipulaci pomocí JavaScriptu. Komponenta přijme `manifestUrl` jako *prop* [37], což umožňuje dynamické načítání různých manifestů v závislosti na uživatelských interakcích nebo jiných aplikacích.

*Props* (vlastnosti) ve Vue . js jsou klíčové pro předávání dat mezi komponentami. v případě `MiradorViewer`, `manifestUrl` jako *prop* umožňuje komponentě dynamicky reagovat na změny a načítat různé IIF manifesty podle potřeb uživatele. Tento mechanismus je základem pro reaktivní a uživatelsky přívětivé aplikace, které potřebují flexibilně reagovat na změny v uživatelském vstupu nebo externích datech.

### 6.3.6 Dynamické zpracování a aktualizace manifestu

Dynamika komponenty `MiradorViewer` je dále umocněna použitím sledování změn vlastnosti `manifestUrl`. Při změně této vlastnosti je pomocí *watcheru* [38] automaticky spuštěna aktualizace Miradoru s novým manifestem. Tento přístup bude

zajišťovat, že komponenta reaguje na změny a udržuje aktualitu zobrazovaného obsahu.

*Watcher* je ve `Vue.js` nástroj, který sleduje změny ve vlastnostech dat a umožňuje vykonat akce, když dojde k těmto změnám. v `MiradorViewer` je `watcher` na `manifestUrl` využit k okamžité aktualizaci obsahu prohlížeče `Mirador`, což je zásadní pro udržení aktuálnosti zobrazovaných informací.

Z tohoto `watcher`u bude volána metoda `initializeMirador`, která inicializuje `Mirador` s novým manifestem, který je předán jako argument.

### 6.3.7 Návrh testování

Testování komponenty `MiradorViewer` by mělo zahrnovat jednotkové testy pro ověření funkcionality metod, integrační testy pro testování interakcí mezi komponentami a uživatelské testy pro zajištění, že komponenta splňuje očekávání koncových uživatelů. Ladění a kontinuální monitoring během vývoje a po nasazení aplikace zajistí, že komponenta bude stabilní a spolehlivá.



# Praktické ověření návrhu

## 7

Tato kapitola se bude věnovat popisu praktické implementace návrhu vytvoření manifestů pro díla a jeho ověření včetně validace a porovnání výsledků. Za účelem umožnění vlastní implementace a testování byla v rámci infrastruktury Západočeské univerzity vytvořena demonstrační verze projektu *Inventaria Rudolphina*. Tento krok usnadnil vývoj této práce a poskytl nezbytné prostředí pro její realizaci.

Tento postup zahrnoval následující kroky:

- Inicializace spojení s databázovým systémem za účelem extrakce dat.
- Selektivní získávání dokumentů, které jsou nezbytné pro další zpracování a transformaci. Každý dokument reprezentuje specifické obrazová nebo textová data.
- Konstrukce struktury manifestu podle IIIF specifikací.
- Transformace extrahovaných JSON dokumentů do struktury manifestu.
- Testování a validace vytvořené struktury.

## 7.1 Implementace

Pro vytvoření skriptu byl použit programovací jazyk Python verze 3.12.3 a knihovny `pymongo` a `json`.

### 7.1.1 Komunikace s databází a uložení

Jako první bylo potřeba se postarat o získání správných souborů z databáze. Jedná se o JSON reprezentace konkrétních položek z kolekce `inventory`. Byly vytvořeny dva skripty, které se starají o bezpečný přístup a manipulaci s daty uloženými v MongoDB prostřednictvím SSH tunelu. Skript `config.py` zahrnuje konfigurační nastavení a uživatelské interakce potřebné pro získání přihlašovacích údajů k SSH. Uživatel je vyzván k zadání svého SSH uživatelského jména a hesla, přičemž heslo je

získáváno bezpečně pomocí funkce `getpass` pro zamezení jeho zobrazení v konzoli. Dále skript definuje parametry pro připojení, včetně hostname, portu a lokálního portu pro přesměrování MongoDB.

Hlavní skript `extract.py` využívá tyto konfigurační nastavení pro vytvoření SSH tunelu s využitím knihovny `sshtunnel`. Po úspěšném navázání tunelu skript vytvoří spojení s databází MongoDB, odkud čte data z kolekce `inventory` databáze `rudolf`. Skript dále načte a omezeně ukládá dokumenty do JSON souborů. Tento proces zahrnuje i zachycení a řešení případných výjimek, čímž zajišťuje robustnost a odolnost celého procesu vůči chybám. Po dokončení operací skript bezpečně uzavírá SSH tunel a uvolňuje všechny zdroje, což minimalizuje riziko bezpečnostních hrozeb. Tento přístup demonstruje, jak lze bezpečně manipulovat s daty vzdálené databáze přes SSH tunel, což je klíčové pro ochranu citlivých informací a udržení integrity dat v průběhu jejich transferu a zpracování.

## 7.1.2 Transformace JSON souborů do IIIF manifestu

Hlavní skript `transform.py` transformuje JSON data do IIIF manifestů podle IIIF Presentation API 3.0.

Skript začíná načtením předaného JSON souboru, který obsahuje data o digitálních objektech. Tato data jsou přečtena a načtena do slovníku v Pythonu pomocí funkce `json.load()`.

Na základě načtených dat vytváří skript základní strukturu manifestu, která zahrnuje kontext `@context`, identifikátor `id`, typ `type`, že se jedná o manifest, a základní popis `label`, který je získán z atributu `caption`.

Dále se skript stará o vytváření obecných metadat o položce. Metadata jsou extrahována z dat a převedena do formátu, který vyhovuje specifikaci IIIF. To zahrnuje informace jako `xml_id`, autora, název a další relevantní údaje. Metadata jsou organizována do seznamu slovníků, kde každý slovník obsahuje štítek `label` a hodnotu `value`.

Dále je potřeba vytvořit canvas pro každý obázek, který položka obsahuje. na základě počtu objektů v načtených datech skript rozhoduje o počtu canvasů v manifestu. Pokud je objekt pouze jeden (tj. obsahuje pouze informace o autorovi) a neobsahuje obrázek, skript vytvoří prázdný canvas tak, aby byl manifest validní pro zobrazení. Pokud je objektů více, pro každý z nich vytvoří canvas s příslušnými metadaty a referencí na obrázek.

V případě, že analyzovaný JSON dokument postrádá určité atributy, jsou tyto atributy při kompilaci manifestu vynechány. Tento postup byl rovněž aplikován ve webové aplikaci Inventaria, kde se podobným způsobem zpracovávají `missing data`.

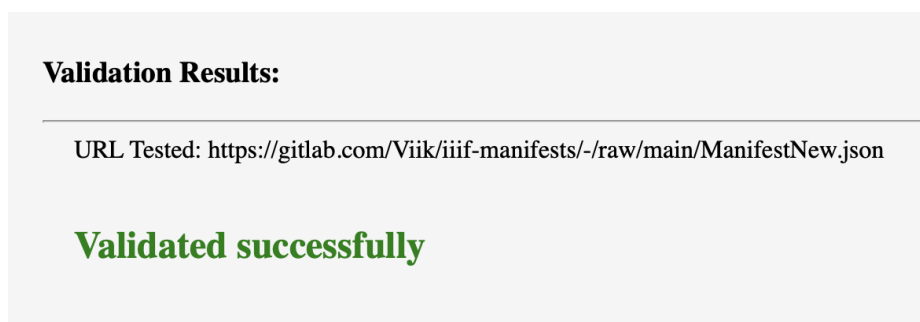
## 7.2 Testování

Tato sekce se zaměří na testování praktické části, která byla strukturována do dvou samostatných oddílů. První oddíl se soustředí na ověření platnosti manifestu, což je výstup praktické části, za účelem zajištění jeho plné validity a kompatibility s verzí 3 Presentation API. Druhý oddíl pak zkoumá funkčnost implementovaného skriptu. Funkčnost byla ověřena prostřednictvím testovacích scénářů a testování bylo prováděno souborem pěti testovacích osob.

Testování bylo prováděno na souboru pěti JSON dokumentů, z nichž každý představoval odlišný typ datové struktury. Konkrétně byly testovány dokumenty obsahující různé kombinace atributů: od těch, které obsahovaly pouze metadata bez jakýchkoli obrazových souborů, přes dokumenty s jedním či více obrazy, až po dokumenty s chybějícími daty či dokumenty s maximálně možným počtem atributů. Tento různorodý soubor dokumentů umožnil detailně prozkoumat schopnost skriptu zpracovávat a validovat různorodé vstupy podle specifikací, což je klíčové pro jeho univerzální použitelnost a spolehlivost v reálném provozu. Cílem tohoto rozsáhlého testování bylo ověřit, že skript je plně funkční a kompatibilní se všemi možnými formáty a typy vstupních dat.

### 7.2.1 Validace manifestu

Výsledný IIIF manifest generovaný vyvinutým skriptem byl důkladně testován pomocí IIIF Presentation Validátoru dostupného na stránkách IIIF. Tento krok ověření zajistil, že manifest je plně kompatibilní s verzí 3.0 Presentation API. Díky této validaci lze potvrdit správnost a validnost struktury manifestu, což zaručuje, že skript je řádně otestován a funkční. Tento proces validace je klíčový pro zajištění, že výsledné manifesty vyhovují standardům, jsou validní a příslušné Image viewery je dokáží uživatelům bez problémů zobrazit.



Obrázek 7.1: Ukázka úspěšné validace

Tento přístup testování byl také použit v průběhu implementace skriptu. Bylo provedeno celkem 20 iterací, které sloužily k validaci výstupů a postupnému zdoko-

nalování skriptu. Díky tomu byla zajištěna jeho robustnost vůči různorodým vstupům. Tento postup umožnil efektivně identifikovat a opravit potenciální slabá místa v algoritmu, čímž byla výrazně zvýšena jeho spolehlivost a univerzálnost použití.

## 7.2.2 Testovací scénáře

V průběhu realizace diplomové práce byly vyvinuty testovací scénáře, které systematicky pokrývají celý proces zpracování dat. Scénáře začínají extrakcí dat z databázového systému nebo importem JSON souborů, následuje aplikace vyvinutého skriptu, a končí validací výsledného manifestu. Tento manifest představuje konečný výstup, jehož kvalita a kompatibilita jsou klíčové pro úspěšné nasazení a funkčnost celého systému.

### 7.2.2.1 Příprava prostředí

Je nezbytné pečlivě připravit testovací prostředí v souladu s uživatelskou dokumentací uvedenou v příloze A. Tato příprava zahrnuje instalaci požadované verze programovacího jazyka Python. Kromě toho je třeba nastavit vývojové prostředí podle individuální volby, což umožní efektivní testování skriptu.

### 7.2.2.2 Scénář pro zisk vstupu

První testovací scénář popisuje první fázi praktické části projektu, která zahrnuje připojení k databázi a následnou extrakci dokumentů do formátu JSON. Tyto soubory reprezentují jednotlivé položky díla z projektu Inventaria. Tento krok je klíčový pro zajištění správného zpracování a strukturování dat, což je nezbytné pro další fáze testování a validace výstupů.

1. Příprava prostředí viz příloha A.
2. Otevřít si příkazovou řádku a napsat `python3 extract.py`.
  - Po spuštění je k připojení do databáze přes SSH vyžadováno zadání uživatelského jména (username), které odpovídá orionu a hesla (password), které je totožné s osobním číslem. Po připojení specifikovat počet dokumentů, které chcete extrahovat.
  - v případě, že tester nebude působit přímo v prostředí Západočeské univerzity, budou potřebné dokumenty pro testování poskytnuty zvlášť.

**Výsledek:** Po aktivaci skriptu dojde k výběru a extrakci určeného počtu dokumentů, které jsou následně uloženy do stejné složky, kde je spuštěn tento modul.

### 7.2.2.3 Scénář pro provedení transformace

Druhý scénář se zaměřuje na hlavní část praktického segmentu projektu. Vstupem pro tento scénář jsou JSON soubory získané v předchozím kroku, které obsahují položky určené k transformaci do struktury manifestu. Tento manifest musí být plně kompatibilní s Presentation API IIIF frameworku. Scénář zahrnuje přípravu skriptu, což znamená nastavení vhodného dokumentu pro transformaci a jeho spuštění.

1. Příprava prostředí viz příloha A.
2. Otevřít si příkazovou řádku a napsat `python3 transform.py`.
3. Po spuštění skriptu specifikuje vstupní soubor (případně specifikovat cestu k danému souboru, pokud se nenachází ve stejné složce).

**Výsledek:** Výsledkem procesu bude manifest uložen v souboru, který adaptuje specifické informace z původního JSON dokumentu do struktury plně kompatibilní s IIIF frameworkem.

### 7.2.2.4 Scénář pro ověření validace výstupu

Poslední scénář se zaměřuje na validaci manifestu, který představuje výstup z předchozího testovacího scénáře. Pro účely správné validace je nezbytné zveřejnit manifest na testovacím veřejném serveru, přičemž lze využít platformu jako je GitLab. Po publikaci manifestu se URL adresa vloží do oficiální validační stránky IIIF, kde proběhne konečná verifikace kompatibility a správnosti struktury.

V rámci testovacího scénáře bude zahrnuta možnost uložení manifestu na platformě GitLab.

1. Přejít na stránku projektu GitLabu.
2. Nahrát manifest, který byl vytvořený skriptem.
3. Přejít na validační stránku IIIF.
4. Vložit URL adresu manifestu do vstupního pole a kliknout na tlačítko *Go!* pro spuštění validace.

**Výsledek:** Výsledek validace znázorňuje správnost struktury manifestu s Presentation API verze 3. Pokud se zobrazí zeleně *Validated successfully*, pak je vložený manifest plně validní. Pokud se zobrazí *Validation Error* červenou barvou, zobrazí se následně také informace o nalezené chybě.

## 7.3 Evaluace výsledku implementace

Návrh byl plně implementován a funkčnost skriptu byla ověřena. Výsledkem skriptu je manifest validní s nejnovější verzí Presentation 3.0 API a je možno ho prohlížet v IIIF image vieweru.

### 7.3.1 Ukázka vytvořeného manifestu

Pro ukázkou výstupu byl vygenerován příkladový manifest z dokumentu s jedním obrázkem v příloze C. Pro kratší ukázkou byla odmazána metadata a manifest obsahuje jedno plátno pro jeden obrázek.

Výsledný manifest obsahuje úvodní hlavičku, kde jsou specifikovány elementy `context`, `id` a `type`, které označují dokument jako manifest. Dále obsahuje `label`, který identifikuje dílo, například jeho název nebo jiný identifikátor.

Následuje sekce s metadaty, která aktuálně zahrnuje `xml id`, `author`, `subject` a `inventory name`. Pro díla umístěná na Pražském hradě je zde navíc zahrnuto `room` a `place`. Metadata představují prostor pro další možné rozšíření atributů, které mohou být zobrazeny.

Sekce `items` manifestu slouží jako seznam dalších položek, v tomto případě pláten a jejich metadata. Tato struktura je nezbytná, jelikož vstupní skript nemusí nutně obsahovat obrázky, nebo může obsahovat více než jeden obrázek, a skript musí zaručit správné zobrazení všech možných případů.

V případě, že dílo obsahuje pouze metadata bez obrázku, skript vytvoří v atributu `items` prázdné plátno. Image viewer následně zobrazí informace o díle, ale bez příslušného obrázku.

```
1  "items": [  
2  {  
3    "id": "https://example.org/iiif/blank/canvas",  
4    "type": "Canvas",  
5    "label": {  
6      "en": [  
7        "Blank Canvas"  
8      ]  
9    },  
10   "height": 1000,  
11   "width": 2000,  
12   "items": []  
13 }  
14 ]
```

Zdrojový kód 7.1: Příklad prázdného plátna

V případě, že skript detekuje přítomnost obrázků, vytvoří v sekci `items` odpovídající počet pláten. Pro každé plátno jsou definovány vlastnosti `id`, `type`, `label`

a rozměry pro vykreslení. Dále manifest zahrnuje vnořenou sekci metadata, která obsahuje specifické informace o jednotlivých obrázcích. Tyto informace mohou zahrnovat například míru jistoty přiřazení obrázku k dílu, lokalizaci, kde se obraz nacházel, nebo malířský styl, kterým byl obrázek vytvořen. Vzhledem k tomu, že tyto údaje se mohou u jednotlivých obrázků lišit, je nutné je přiřadit každému obrázku (plátnu) zvlášť.

Sekce annotation page obsahuje v hlavičce id, type a items s konkrétními informacemi o obrázku. Plátno dále obsahuje atribut motivation, který specifikuje, že se jedná o *painting*. Následuje sekce body s id, která obsahuje URL odkaz na konkrétní obrázek. Další specifikace zahrnují type s označením *image* a atributy format, height a width, které poskytují informace o formátu a rozměrech obrázku

```

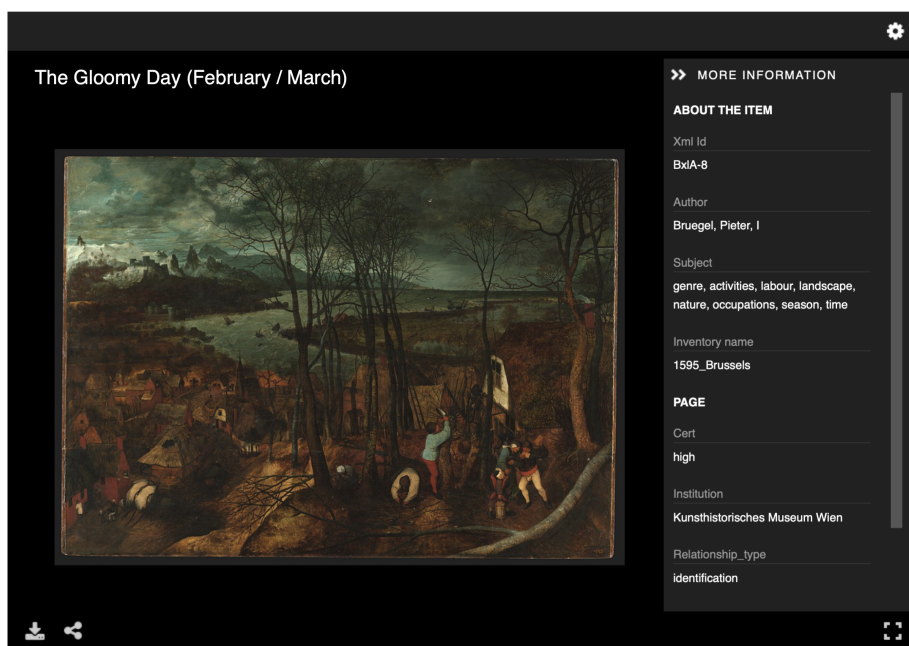
1  "items": [{
2    "id": "https://ex.org/iiif/item1234/canvas/1",
3    "type": "Canvas",
4    "label": {
5      "en": [
6        "Identifikace obrázku"
7      ]},
8    "height": 118,
9    "width": 163,
10   "metadata": [
11     ...
12   "items": [{
13     "id": "https://ex.org/iiif/item1234/page/1/1",
14     "type": "AnnotationPage",
15     "items": [{
16       "id": "https://ex.org/iiif/item1234/annotation/1",
17       "type": "Annotation",
18       "motivation": "painting",
19       "body": {
20         "id": "https://inventariarudolphina.com/static/
21         images/BxlA-8.jpg",
22         "type": "Image",
23         "format": "image/jpeg",
24         "height": 118,
25         "width": 163},
26       "target": "https://ex.org/iiif/item1234/canvas/1"
27     }]}
28   }
29 ]
30 }
31 ]

```

Zdrojový kód 7.2: Příklad vyplněného plátna

### 7.3.1.1 Ukázka výsledného obrázku v prohlížeči

Následuje ukázka jednoduchého manifestu, který byl vygenerován skriptem v prohlížeči. Tento manifest obsahuje jedno plátno. Pro zobrazení ukázky byl použit prohlížeč Universal Viewer. Tento nástroj byl vybrán pro testování manifestů v rámci této práce díky jeho schopnosti jednoduchého online testování, které nevyžaduje instalaci ani modifikace manifestu.



Obrázek 7.2: Ukázka vykresleného obrázku pomocí Universal viewer



# Náměty na budoucí rozšíření

## 8

Po vytvoření a ověření implementace se nabízí řada možností pro další vylepšení, které by bylo vhodné zvážit pro budoucí rozvoj.

### 8.1 Rozšíření manifestu

Manifest, který byl vygenerován skriptem, slouží jako základní prototyp pro strukturu, kterou by mohla Inventaria v budoucnu využívat. Obsahuje všechny potřebné vlastnosti a atributy pro to, aby manifest byl validní na základě Presentation API verze 3.0 a je zobrazitelný v prohlížečích obrázků, které podporují odpovídající verzi zmíněného API.

První, co se nabízí jako vhodné rozšíření, je zobrazení dalších metadat. Stávající originální položka obsahuje bohaté množství různých informací o díle a jednotlivých obazech. Zároveň si tahá dodatečné informace o autorovi z *Getty ulan*. Po důkladné analýze z hlediska výzkumníků stojí za zvážení rozšíření těchto informací do manifestů ve vlastnosti metadata.

Dalším krokem by mělo být zahrnutí vlastnosti `thumbnail`, která by reprezentovala vizuální představení jednotlivých děl. Rovněž je možné přidat informace o autorských právech (vlastnost `rights`), které by manifest obohatily o údaje o licenci a právech spojených s dílem.

Nabízí se také vlastnost `partOf` či `seeAlso`. Vlastnost `partOf` by mohla v případě Inventaria sloužit pro propojení děl (manifestů), které spolu nějakým způsobem souvisí. Tedy díla, která jsou stejná či jsou si podobná z více různých inventářů. Toto řešení by dokázalo udržet a zachovat společnou mezi těmito díly.

### 8.2 Integrace poznámek

Jako další by se nabízelo zahrnutí poznámek k jednotlivým dílům. Momentální řešení Inventaria je rozděleno to několika kolekcí. Prozatím se brala v úvahu pouze kolece `inventory`, kde se nachází jednotlivé položky včetně jejich všech metadat.

Dalším užitečným rozšířením by bylo zachování poznámek výzkumníků. v databázi je přítomna kolekce notes, kde každý dokument obsahuje následující strukturu:

```

1  {
2    "_id": {
3      "12345"
4    },
5    "note": "Text poznámky.",
6    "items": [
7      "VieA-26"
8    ],
9    "uuid": "12345",
10   "created": 12345,
11   "updated": 12345,
12   "created_by": "Tereza Nová",
13   "created_by_id": "12345",
14   "updated_by": "Jana Novotná",
15   "updated_by_id": "12345",
16   "replied": 12345
17 }

```

Zdrojový kód 8.1: Ukázka položky z kolekce notes

Jak je patrné z ukázky, každá poznámka obsahuje atribut `items`, který odkazuje na specifické `xml_id` z kolekce `inventory`. Tento atribut poskytuje možnost efektivně spárovat poznámky s odpovídajícím manifestem. Pro realizaci tohoto spojení by bylo možné využít *Annotation page* [39].

Každý manifest by mohl na svém konci obsahovat sekci, která by sloužila jako odkaz na *Annotation Page file* konkrétní poznámky. Tento přístup by umožnil přímé spojení mezi manifestem a relevantními poznámkami, což by zvýšilo informační hodnotu a užitečnost celého digitálního archivu.

```

1  "annotations": [
2    {
3      "id": "https://example.cz/annotationpage1.json",
4      "type": "AnnotationPage"
5    }
6  ]

```

Zdrojový kód 8.2: Ukázka přidání Annotation do manifestu

Dále by se pro každou poznámku vytvořil *Annotation Page file* (s odpovídajícím názvem v manifestu), která by obsahovala všechny potřebné atributy včetně textu samotné poznámky. Mohla by vypadat takto:

```

1  {
2    "@context": "http://example.cz/context.json",
3    "id": "https://example.cz/annotationpage.json",
4    "type": "AnnotationPage",

```

```

5   "items": [
6     {
7       "id": "https://example.cz/canvas-1/annopage-2/anno-1",
8       "type": "Annotation",
9       "motivation": "commenting",
10      "body": {
11        "type": "TextualBody",
12        "language": "cz",
13        "format": "text/plain",
14        "value": "Text poznámky"
15      },
16      "target": "https://example.cz/canvas-1"
17    }
18  ]
19 }

```

Zdrojový kód 8.3: Ukázka přidání Annotation page file

Tímto způsobem by se dalo jednoduše integrovat poznámky do nového řešení tak, aby byly viditelné i z rozhraní prohlížeče obrázků.

## 8.3 IIIF kolekce jako inventáře

Inventaria aktuálně uchovává jednotlivé položky v rámci inventářů, které jsou organizovány jako stromová struktura. Tyto inventáře by mohly být transformovány do struktury IIIF pod formou kolekcí (*collections*). Jak bylo popsáno v kapitole o seznámení s IIIF, mezi různé typy zdrojů patří i kolekce. Tyto kolekce mohou efektivně shrnovat související manifesty.

Pro každý inventář, například *1595 Brussels*, by Inventaria mohla vytvořit odpovídající kolekci. Tato kolekce by následně udržovala seznam manifestů, které jsou zařazeny do daného inventáře.

Kolekce by mohla vypadat následovně:

```

1   "type": "Collection",
2   "label": { "en": [ "1595 Brussels" ] },
3   ....
4   "items": [
5     {
6       "id": "https://example.org/iiif/1/manifest",
7       "type": "Manifest",
8       "label": { "en": [ "Example Manifest 1" ] },
9       ...
10    },
11    ....
12  ]

```

Zdrojový kód 8.4: Ukázka kolekce

Tento způsob by mohl efektivně reprezentovat manifesty v inventářích tak, aby se zachovala stávající struktura Inventaria.

# Zhodnocení výsledků

## 9

Tato kapitola se zaměří na zhodnocení výsledků práce, která se věnovala analýze, návrhu a vybrané implementaci IIIF v rámci projektu *Inventaria Rudolphina*. Cílem bylo nejen zkoumání teoretických aspektů frameworku, ale především teoretická i praktická aplikace a ověření jeho přínosů pro zlepšení dostupnosti a interoperability obrazových dat. Tato kapitola poskytuje přehled o dosažených výsledcích, jejich srovnání s původními očekávaními a cíli projektu, a reflektuje přínosy a omezení zavedení IIIF do praxe *Inventaria Rudolphina*.

Implementace IIIF do digitálního archivu *Inventaria Rudolphina* představil značné změny ve způsobu správy a prezentace obrazových dat. Stávající řešení projektu se potýkalo s několika omezeními, která komplikovala efektivní správu a především široké využití digitálních sbírek. Mezi hlavní problémy patřilo omezené rozhraní pro interakci s obrazy, nedostatečné možnosti pro detailní vizuální analýzu a složitá správa metadat.

## 9.1 Vylepšení zavedením IIIF manifestů

Implementace skriptu pro transformaci stávající struktury do struktury IIIF manifestů přinesla několik klíčových vylepšení, která řeší předchozí nedostatky a rozšiřují možnosti uživatelů přistupovat a interagovat s obrazovým materiálem.

### 9.1.1 Nedostatky stávajícího řešení

Stávající řešení se potýkalo s několika klíčovými omezeními, která omezila jeho efektivitu a uživatelskou přívětivost.

- **Špatně navržená struktura:** Data byla často nekonzistentní a chaotická, což komplikovalo jejich správu a zpracování. Nedostatek standardizace ztěžoval aktualizace a integraci s jinými systémy.

- **Omezené sdílení:** Digitální sbírky byly kompatibilní pouze s interním systémem a nedaly se snadno sdílet s externími platformami nebo využívat v běžných nástrojích pro digitální knihovny.
- **Omezená interaktivita:** Uživatelé tudíž neměli k dispozici pokročilé nástroje pro prohlížení obrazů, což omezilo jejich schopnost detailně zkoumat a analyzovat obrazový materiál.

## 9.1.2 Výhody nového řešení

Implementace IIIF manifestů přinesla řadu vylepšení, která adresovala tyto výzvy.

- **Vylepšení interaktivity a přístupnosti:** IIIF manifesty nyní umožňují pokročilé funkce prohlížení, včetně hlubokých zoomů a možnosti porovnávat obrazy vedle sebe. Toto zlepšení výrazně obohatilo uživatelskou zkušenost a umožnilo podrobnější studium obrazových detailů.
- **Efektivnější správa metadat:** Díky strukturovanému formátu IIIF manifestů je správa metadat systematickejší a transparentnější. Metadata jsou nyní úzce spojená s obrázkovými daty.
- **Zvýšená interoperabilita:** IIIF manifesty jsou kompatibilní s řadou platform a nástrojů, což umožňuje Inventariu snadněji sdílet digitální sbírky s mezinárodními výzkumnými institucemi a zájemci o umění. Toto bylo vnímáno jako primární cíl Akademie věd.

## 9.1.3 Možná omezení nového řešení

Přestože implementace manifestů přináší mnoho výhod, existují i potenciální omezení.

- **Potřeba technické znalosti:** Správa a údržba IIIF manifestů vyžaduje určité technické dovednosti.
- **Potřeba implementace dalších návrhů:** Pro plné využití tohoto řešení je potřeba implementace vybraného image vieweru do řešení Inventaria.

## 9.2 Celkové zhodnocení

Analýza možností využití IIIF v rámci projektu *Inventaria Rudolphina* poskytla důležité poznatky o stávajících omezeních a potenciálu pro zlepšení. Ukázalo se, že standardizace datové struktury a rozšíření funkcí pro interaktivní prohlížení mohou významně zlepšit jak správu obrazových sbírek, tak uživatelskou zkušenost.

Analýza odhalila klíčové oblasti, ve kterých IIIF nabízí výhody, a nastínila možnosti jeho aplikace v akademickém a výzkumném prostředí, což otevírá nové možnosti pro výzkum.

Implementace IIIF do projektu *Inventaria Rudolphina*, v podobě vytvoření manifestů, přinesla významné vylepšení v přístupu k obrazovým datům a jejich interaktivitě. Zavedení IIIF manifestů umožnilo překonání stávajících omezení spojených s neefektivním zpracováním a sdílením obrazových sbírek. Standardizace datové struktury a rozšíření funkcí pro možné interaktivní prohlížení zlepšily uživatelské prostředí a otevřely nové možnosti pro výzkum a vzdělávání. I přes jisté technické a provozní výzvy, které s sebou nové řešení přináší, lze projekt hodnotit jako krok správným směrem s důrazem na další rozvoj a integraci IIIF nástrojů, aby bylo možné maximálně využít jejich potenciál pro správu kulturního dědictví.

Hlavním cílem práce bylo prozkoumat možnosti využití International Image Interoperability Framework (IIIF) pro potřeby projektu *Inventaria Rudolphina*. Byla provedena důkladná analýza IIIF, zmapování aktuálního využití tohoto frameworku v akademickém světě a identifikace možností pro jeho aplikaci v rámci projektu. Z této analýzy vyplynul konkrétní návrh na transformaci stávající struktury položky do IIIF manifestů, který byl následně implementován a otestován.

Kromě tohoto realizovaného návrhu byly identifikovány a navrženy další dva návrhy pro budoucí rozšíření projektu: prvním je sdílení obrazů na IIIF kompatibilním serveru a druhým je integrace image vieweru do webové aplikace *Inventaria Rudolphina*. Tyto návrhy nabízejí možnosti pro zvýšení efektivity správy a přístupnosti digitálních sbírek.

Tato práce ukazuje, že integrace IIIF do systému *Inventaria Rudolphina* má praktický smysl, zejména pro zlepšení sdílení digitálních sbírek a začlenění standardizovaného přístupu do jejich správy. Výsledky a poznatky z této práce poskytují pevný základ pro budoucí plné nasazení IIIF, které může výrazně rozšířit možnosti interoperability a přístupnosti obrazových dat.



# Uživatelská dokumentace



Tato kapitola se bude věnovat instalaci a spuštění všech skriptů z praktické části. Dokumentace je zaměřena na podrobný popis obsahu balíčku, instrukce k přípravě prostředí a postup spuštění skriptů.

## A.1 Obsah balíčku

- `extract.py` - Skript pro extrakci dokumentů z databáze MongoDB.
- `config.py` - Konfigurační soubor obsahující parametry pro připojení k databázi MongoDB a SSH.
- `transform.py` - Skript pro transformaci extrahovaných JSON dokumentů na manifesty IIF.
- `readme.txt` - *Read me* soubor pro instrukce přiložených skriptů.

## A.2 Příprava prostředí

Pro správné fungování skriptů je nezbytné mít nainstalovaný Python ve verzi 3.x a několik externích knihoven, které umožňují manipulaci s databází MongoDB a vytváření SSH tunelu. Následuje postup přípravy prostředí.

### A.2.1 Instalace Pythonu

Je potřeba se ujistit, že je na systému nainstalovaná verze Pythonu 3.x. Pokud není, je ke stažení z oficiálních stránek Pythonu a nainstalujte podle pokynů na těchto stránkách.

### A.2.2 Instalace potřebných knihoven

V terminálu spusťte následující příkaz pro instalaci knihoven potřebných pro běh skriptů:

```
pip3 install pymongo sshunnel
```

## A.2.3 Spuštění skriptů

Tato sekce obsahuje instrukce pro spuštění jednotlivých skriptů, které jsou součástí tohoto balíčku.

### A.2.3.1 Spuštění skriptu `extract.py`

Před prvním spuštěním se ujistěte, že konfigurační soubor `config.py` obsahuje správné informace pro vaše SSH a MongoDB nastavení.

Skript spusťte v terminálu pomocí následujícího příkazu:

```
python3 extract.py
```

Skript nejprve požaduje zadání jména a hesla. Po úspěšném připojení do databáze vás skript vyzve, abyste specifikovali počet dokumentů k extrakci z MongoDB. Tyto dokumenty jsou poté uloženy jako JSON soubory s názvem `[xml_id].json`, přičemž `[xml_id]` odpovídá unikátnímu identifikátoru dokumentu, ve složce, kde skript běží.

### A.2.3.2 Spuštění skriptu `transform.py`

Po úspěšné extrakci dokumentů použijte skript `transform.py` k jejich převodu na manifesty IIIF:

```
python3 transform.py
```

Po spuštění skriptu budete vyzváni k zadání cesty k JSON souborům, které byly vytvořeny pomocí `extract.py`, a následně je skript transformuje do formátu manifestů IIIF. Je nutné správně zadat cestu k těmto vstupním souborům. Výsledkem bude soubor s názvem `[xml_id]-manifest.json`.

## Ukázka struktury položky Inventarie

B

```
1 order: 1,
2 xml_id: "PrgA -811",
3 subject: [
4 "stillLife",
5 "genre",
6 "market",
7 "fruit"
8 ],
9 search_subject: [
10 "stillLife",
11 "genre",
12 "market"
13 ],
14 iconclass_external_id: "41C6",
15 concordances: [
16 {
17   id: "PrgB -811",
18   cert: "same_as"
19 },
20 {
21   id: "PrgD -49",
22   cert: "medium",
23   resp: "MJ"
24 },
25 {
26   id: "PrgD -262",
27   cert: "low",
28   resp: "MJ"
29 },
30 {
31   id: "PrgD -385",
32   cert: "low",
33   resp: "MJ"
34 }
```

```
35 ],
36 text: " Ein Obstmarckt , vom Langen Peter , Original stuck. " ,
37 title: "Ein Obstmarckt , vom Langen Peter , Original stuck." ,
38 object: [
39 {
40   caption: "Vegetable and Fruit Market" ,
41   type: "painting" ,
42   digitalisation_layer: true ,
43   name: [
44     {
45       value: "Langen Peter" ,
46       role: "artist" ,
47       getty_external_id: "500027884" ,
48       getty_data: {
49         getty_external_id: "500027884" ,
50         name: "Aertsen , Pieter" ,
51         description: "Netherlandish painter , 1507/1508 -1575" ,
52         gender: "male" ,
53         est_start: "1507" ,
54         est_end: "1575" ,
55         birth_place_name: "Amsterdam" ,
56         death_place_name: "Amsterdam" ,
57         display_name: "Pieter Aertsen" ,
58         altname: [
59           "Aertsz. , Pieter" ,
60           "Aertsen , Lange Pier"
61         ] ,
62         note: "Netherlandish painter known for genre and still
63         life paintings that subtly incorporate religious
64         scenes. Comment on works: Still life ; Genre ;
65         Kitchen and Market scenes ; Religious" ,
66         nationality: "Netherlandish"
67       }
68     }
69   ]
70 },
71 {
72   xml_id: "alt -1-PrgA-B-811" ,
73   relationship_type: "identification" ,
74   images: [
75     {
76       file: "PrgA-811_PrgB-811.jpg" ,
77       text: "Pieter Aertsen , Vegetable and Fruit Market , 1569 ,
78       Hallwylska museet , Stockholm "
79     }
80   ] ,
81   cert: "high" ,
82   caption: "Vegetable and Fruit Market" ,
```

```
83 object_name: "Vegetable and Fruit Market",
84 id_number: "XXXII: B.20.",
85 institution: "Hallwylska museet",
86 country: "Sweden",
87 city: "Stockholm",
88 latitude: 59.3331,
89 longitude: " 18.0745",
90 physical_description: "Bibliography:
91 Christina 1966, p. 505, Nr. 1271. For more information
92 see ....",
93 name: [
94 {
95   role: "artist",
96   getty_external_id: "500027884",
97   getty_data: {
98     getty_external_id: "500027884",
99     name: "Aertsen, Pieter",
100    description: "Netherlandish painter, 1507/1508–1575",
101    gender: "male",
102    est_start: "1507",
103    est_end: "1575",
104    birth_place_name: "Amsterdam",
105    death_place_name: "Amsterdam",
106    display_name: "Pieter Aertsen",
107    altname: [
108      "Aertsz., Pieter",
109      "Aertsen, Lange Pier"
110    ],
111    note: "Netherlandish painter known for genre and still
112    life paintings that subtly incorporate religious
113    scenes. Comment on works: Still life; Genre;
114    Kitchen and Market scenes; Religious",
115    nationality: "Netherlandish"
116  }
117 }
118 ],
119 material: "oil
120 on panel",
121 type: "painting",
122 dimensions: {
123   unit: "cm",
124   height: 83.5,
125   width: 169.5
126 },
127 origDate: "1569",
128 provenance: "May 4, 1898 bought by Walther von Hallwyl at
129 Henryk Bukowski, Stockholm."
130 }
```

```
131 ],
132 inventory: {
133   name: "1621_Prague_A",
134   label: "1621 Prague A",
135   order: 6
136 },
137 room: {
138   original_description: "Folgen ferner die Gemahl, welche in
139   den fordern gang sein.",
140   id: 5
141 },
142 place: {
143   original_description: "oben auf dem gesims",
144   id: 3
145 },
146 prevItem: false ,
147 nextItem: "PrgA -812"
```

Zdrojový kód B.1: Příklad struktury detailu položky

# Příklad vytvořeného manifestu

C

```
1 "@context": "http://iiif.io/api/presentation/3/context.json",
2 "id": "https://example.org/iiif/manifest",
3 "type": "Manifest",
4 "label": {
5   "en": [
6     "Identifikace díla"
7   ]
8 },
9 "metadata": [
10 // prostor pro metadata
11 ],
12 "items": [
13 {
14   "id": "https://example.org/iiif/item1234/canvas/1",
15   "type": "Canvas",
16   "label": {
17     "en": [
18       "Value"
19     ]
20   },
21   "height": 800,
22   "width": 1000,
23   "metadata": [
24 // prostor pro metadata
25   ],
26   "items": [
27 {
28   "id": "https://example.org/iiif/item1234/page/1/1",
29   "type": "AnnotationPage",
30   "items": [
31 {
32   "id": "https://example.org/iiif/item1234/annotation/1",
33   "type": "Annotation",
34   "motivation": "painting",
35   "body": {
```

```
36     "id": "https://inventariarudolphina.com/static/  
37     images/BxlA-8.jpg",  
38     "type": "Image",  
39     "format": "image/jpeg",  
40     "height": 800,  
41     "width": 1000  
42   },  
43   "target": "https://example.org/iiif/item1234/canvas/1"  
44 }  
45 ]  
46 }  
47 ]  
48 }  
49 ]
```

Zdrojový kód C.1: Příklad vytvořeného manifestu ze skriptu



# Seznam zkratek

IIIF - International Image Interoperability Framework

IIIF-C - Konsorcium IIIF

API - Application Programming Interface

UX - User experience (uživatelská přívětivost)

OCR - Optical character recognition

URI - Uniform Resource Identifier

URL - Uniform Resource Locator

HTTP(S) - Hypertext Transfer Protocol (Secure)

JSON - JavaScript Object Notation

XML - Extensible Markup Language

RDF - Resource Description Framework

PDF - Portable Document Format

CORS - Cross-origin resource sharing

DNS - Domain Name System

DOM - Document Object Model

SSH - Secure Shell

# Bibliografie

1. *Centrum studia Rudolphina* [online]. Ústav dějin umění, Akademie věd České republiky, 2020. [cit. 2023-12-05]. Dostupné z: <https://www.udu.cas.cz/cz/centra/studia-rudolphina-centre>.
2. KLOUČKOVÁ, Jitka. *Rudolf II. a jeho vztah k umění* [online]. 2014. [cit. 2024-10-18]. Dostupné z: [https://dspace5.zcu.cz/bitstream/11025/13633/1/BP\\_Klouckova.pdf](https://dspace5.zcu.cz/bitstream/11025/13633/1/BP_Klouckova.pdf).
3. KAUFMANN, Thomas DaCosta. *The school of Prague: painting at the court of Rudolf II.* University of Chicago Press Chicago, 1988.
4. ZLATOHLÁVEK, Martin. *Umění na dvoře rudolfa II.* [online]. [B.r.]. [cit. 2024-10-18]. Dostupné z: <https://udu.ff.cuni.cz/wp-content/uploads/sites/166/2021/03/41-Umeni-doby-Rudolfa-II.pdf>.
5. *IIIF introduction* [online]. IIIF. [cit. 2023-12-05]. Dostupné z: <https://iiif.io/get-started/how-iiif-works/>.
6. *Community Map* [online]. IIIF. [cit. 2024-04-17]. Dostupné z: <https://iiif.io/community/map/>.
7. *What is Interoperability?* [online]. AWS. [cit. 2023-11-02]. Dostupné z: <https://aws.amazon.com/what-is/interoperability/>.
8. CRAMER, Tom. *IIIF consortium formed* [online]. Tom Cramer, 2015 [cit. 2023-11-02]. Dostupné z: <https://iiif.io/news/2015/06/17/iiif-consortium/>.
9. *About the Consortium* [online]. IIIF. [cit. 2023-12-05]. Dostupné z: <https://iiif.io/community/consortium/>.
10. *What is an API* [online]. GeeksForGeeks, 2024. [cit. 2024-01-22]. Dostupné z: <https://www.geeksforgeeks.org/what-is-an-api/>.
11. *Image API 3.0* [online]. IIIF. [cit. 2024-01-27]. Dostupné z: <https://iiif.io/api/image/3.0/>.
12. *Presentation API 3.0* [online]. IIIF. [cit. 2024-01-27]. Dostupné z: <https://iiif.io/api/presentation/3.0/>.

13. *IIIF Authorization Flow API 2.0* [online]. IIIF. [cit. 2024-01-27]. Dostupné z: <https://iiif.io/api/auth/2.0/>.
14. *IIIF Change Discovery API 1.0* [online]. IIIF. [cit. 2024-01-27]. Dostupné z: <https://iiif.io/api/discovery/1.0/>.
15. *IIIF Content Search API 2.0* [online]. IIIF. [cit. 2024-01-27]. Dostupné z: <https://iiif.io/api/search/2.0/>.
16. *IIIF Content State API 1.0* [online]. IIIF. [cit. 2024-01-27]. Dostupné z: <https://iiif.io/api/content-state/1.0/>.
17. *API Specifications - International Image Interoperability Framework* [online]. IIIF. [cit. 2024-01-27]. Dostupné z: <https://iiif.io/api/index.html>.
18. *Viewer Matrix* [online]. IIIF. [cit. 2024-03-18]. Dostupné z: <https://iiif.io/api/cookbook/recipe/matrix/>.
19. *Mirador* [online]. Mirador. [cit. 2024-04-09]. Dostupné z: <https://projectmirador.org>.
20. *IIIF Viewers* [online]. IIIF. [cit. 2024-04-09]. Dostupné z: <https://iiif.io/get-started/iiif-viewers/>.
21. *Universal Viewer* [online]. Universal Viewer. [cit. 2024-04-09]. Dostupné z: <https://universalviewer.io>.
22. *OpenSeadragon* [online]. OpenSeadragon. [cit. 2024-04-09]. Dostupné z: <https://openseadragon.github.io>.
23. *Tify: A slim and mobile-friendly IIIF document viewer* [online]. Tify. [cit. 2024-04-09]. Dostupné z: <https://tify.rocks>.
24. *IIIF Curation Viewer*. Center for Open Data in the Humanities. Dostupné také z: <http://codh.rois.ac.jp/software/iiif-curation-viewer/>.
25. TURNER, Lauren. *(IIIF) for Museums and Cultural Institutions* [online]. GallerySystems. [cit. 2024-04-12]. Dostupné z: <https://www.gallerysystems.com/international-image-interoperability-framework-iiif/>.
26. *Using IIIF for Image Retrieval in Digital Libraries: Experimentation of Deep Learning Techniques* [online]. IIIF. [cit. 2024-02-21]. Dostupné z: <https://iiif.io/event/2019/goettingen/program/6/>.
27. *IIIF: Improving the Interoperability of Digital Materials* [online]. IIIF. [cit. 2024-02-21]. Dostupné z: <https://www.oclc.org/research/areas/data-science/iiif.html>.
28. *Cantaloupe* [online]. Cantaloupe. [cit. 2024-04-09]. Dostupné z: <https://cantaloupe-project.github.io>.

29. *Try MongoDB Tools Free* [online]. MongoDB. [cit. 2024-05-02]. Dostupné z: <https://www.mongodb.com/try/download/compass>.
30. *Studio 3T* [online]. Studio 3T. [cit. 2024-05-02]. Dostupné z: <https://robomongo.org>.
31. *MongoDB Compass vs Robo 3T* [online]. Stackshare. [cit. 2024-05-02]. Dostupné z: <https://stackshare.io/stackups/mongodb-compass-vs-robo-3t>.
32. *iiif-processor* [online]. npmjs. [cit. 2024-05-18]. Dostupné z: <https://www.npmjs.com/package/iiif-processor>.
33. *NPM - Build amazing things* [online]. Npm js. [cit. 2024-05-25]. Dostupné z: <https://www.npmjs.com>.
34. *Node.js documentation* [online]. Node.js. [cit. 2024-05-25]. Dostupné z: <https://nodejs.org/api/stream.html#stream>.
35. *iiif-processor* [online]. Npm.js. [cit. 2024-05-25]. Dostupné z: <https://www.npmjs.com/package/iiif-processor#dimension-function>.
36. *Vue.js: Options: Lifecycle* [online]. Vue.js. [cit. 2024-05-25]. Dostupné z: <https://vuejs.org/api/options-lifecycle.html#mounted>.
37. *Vue.js: props* [online]. Vue.js. [cit. 2024-05-25]. Dostupné z: <https://vuejs.org/guide/components/props.html>.
38. *Vue.js: watchers* [online]. Vue.js. [cit. 2024-05-25]. Dostupné z: <https://vuejs.org/guide/essentials/watchers.html>.
39. *Embedded or referenced Annotations* [online]. IIIF. [cit. 2024-02-11]. Dostupné z: <https://iiif.io/api/cookbook/recipe/0269-embedded-or-referenced-annotations/>.

# Seznam obrázků

2.1	Zobrazení záložek nad kartou detailu . . . . .	10
2.2	Jméno autora a název díla . . . . .	10
2.3	Příklad zobrazení <i>Inventory item</i> . . . . .	10
2.4	Příklad označení pokoje a místa . . . . .	10
2.5	Příklad poznámky . . . . .	11
2.6	Ukázka <i>IconClass</i> . . . . .	11
2.7	Klíčová slova <i>Subject</i> . . . . .	11
2.8	Dodatečné informace k dílu . . . . .	11
2.9	Vizualizace plánu Pražského hradu . . . . .	12
2.10	Příklad karty s informacemi o autorovi . . . . .	12
2.11	Ukázka obrazu s označením <i>Identified</i> a se střední mírou jistoty . . . . .	13
2.12	Ukázka obrazu s označením <i>Resembling</i> . . . . .	14
2.13	Ukázka obrazu s označením <i>Related</i> . . . . .	14
3.1	Příklady správného použití parametru <i>region</i> . . . . .	22
3.2	Příklady správného použití parametru <i>size</i> . . . . .	24
3.3	Příklady správného použití parametru <i>Rotation</i> . . . . .	25
3.4	Příklady správného použití parametru <i>Quality</i> . . . . .	25
3.5	Data model (čerpáno z [12]) . . . . .	31
4.1	Grafová ukázka procentuálního užití <i>Image Viewers</i> z vybraného vzorku dat (vlastní zpracování) . . . . .	46
4.2	Zobrazení využití <i>IIIF</i> ve světě [6] . . . . .	48
6.1	Procesní diagram transformace . . . . .	57
7.1	Ukázka úspěšné validace . . . . .	67
7.2	Ukázka vykresleného obrázku pomocí <i>Universal viewer</i> . . . . .	72

# Seznam výpisů

3.1	Ukázka použití parametrů v URI . . . . .	26
6.1	Struktura výsledného URI . . . . .	60
7.1	Příklad prázdného plátna . . . . .	70
7.2	Příklad vyplněného plátna . . . . .	71
8.1	Ukázka položky z kolekce <code>notes</code> . . . . .	74
8.2	Ukázka přidání Annotation do manifestu . . . . .	74
8.3	Ukázka přidání Annotation page file . . . . .	74
8.4	Ukázka kolekce . . . . .	75
B.1	Příklad struktury detailu položky . . . . .	83
C.1	Příklad vytvořeného manifestu ze skriptu . . . . .	87



1101001 1100001  
10101100001110010 1100001  
101011010101 10



11010011101101001 10101  
01100001 10101  
111000101011 101