The International Image Interoperability Framework (IIIF): A community & technology approach for web-based images

Stuart Snydman, Stanford University, Stanford, CA; Robert Sanderson, Stanford University, Stanford, CA; Tom Cramer, Stanford University, Stanford, CA

Abstract

The International Image Interoperability Framework (http://iiif.io) is a community of academic and national libraries, research institutions, museums, archives, nonprofits and commercial organizations that are committed to interoperable image delivery on the web. IIIF has defined common application programming interfaces (APIs) for image and metadata delivery, has supported the development of best-in-class image delivery and comparison software, and has cultivated a growing and sustainable community model for interoperable image repositories. This paper provides an overview of IIIF, with a focus on its origins, rationale and benefits to users and institutions. We describe the motivating use cases and applications for image interoperability, walk through the technical framework and supported APIs, and highlight IIIF-compatible software for image analysis, comparison and annotation. We conclude with a discussion of the IIIF community process and future directions.

Introduction

Images are primary carriers of global cultural heritage, and digital images are the medium by which important artifacts are shared with and interpreted by the widest international audience. The digital imaging programs of cultural heritage institutions have evolved over the years to achieve efficiency and quality in large-scale digitization, and our repositories have come a long way in effectively preserving and delivering these important digital materials. Yet still, the infrastructures we build to store and manage digital images, as well as the online web applications that we build to deliver them, remain monolithic and institution-specific: our images are trapped in virtual silos. For research and memory organizations, reducing the friction and unnecessary overhead of information delivery is a common, critical need. Yet cultural heritage image repositories have been built as walled gardens of technology, with institutions implementing similar solutions in vastly different ways with few mechanisms for easy exchange of data and little sharing of code or methodology. A scholar wanting to study or compare images held by multiple institutions must switch from website to website, and must learn different interfaces and tools specific to each site she visits. Tools for online comparison and analysis are underwhelming, limiting to users, and technically unsustainable. The International Image Interoperability Framework (http://iiif.io/) addresses these limitations by defining a common technical framework for delivering cultural heritage images on the web, promoting a community of adopters and fostering a marketplace of robust software implementations—both commercial and open source.

The IIIF initiative started small. It was conceived on the back of a napkin at a Cuban restaurant in Palo Alto, California at a dinner of technologists from Stanford University, the British Library, and Oxford. From these humble beginnings, the list of IIIF partners, and its impacts on digital cultural heritage, have steadily grown (http://iiif.io/community.html). Over the past three years, this community of academic and national libraries, museums, archives, nonprofits and commercial organizations has worked together to agree on common APIs for image delivery and developed an ecosystem of compliant systems.

Rationale

Over the past quarter of a century cultural heritage institutions have embarked on ambitious programs of mass digitization, reproducing hundreds of millions of cultural and historical artifacts as high quality images. The goals of these programs are to provide broad access to cherished visual resources in support of teaching at all levels, research, curiosity and recreation. The Google Books Project, Internet Archive, Flickr Commons and the broad reaching digitization efforts of national and public libraries, academic libraries, museums, and archives have transformed the way in which we teach with, learn from, and make meaning of visual resources.

However, despite the sophistication of both imaging and web technologies, current delivery of high resolution cultural heritage imagery does not meet many of the basic expectations of savvy users:

- For end users, image delivery is ***too slow***. In particular for large, high resolution images, users are often faced with long loading times for static images, delays in waiting for tiles to appear to complete the whole image, or long periods of viewing degraded or pixelated images before a full image comes into clear focus.
- Users trying to work with images from a variety of sources find image delivery to be ***too disjointed***. Image sources use a variety of image servers and clients, resulting in a different user experience at every site and requiring users to learn new features, controls and navigation each time they try to use an image from a different source. This results in inefficient scholarly workflows and ultimately frustration.
- For content providers, setting up a high performance image server is ***too complex***. There are many image delivery options to choose from, requiring a substantial institutional investment in hardware, software, system administration and integration support to provide anything beyond basic static image download. For cultural institutions with complex digital asset management systems or repositories, they are either bound to a solution that may not meet their needs or are faced with even more complex integration challenges.

Image repositories themselves tend to be silos of image data, with closed or no APIs, locally specific authentication methods and few mechanisms for interoperability with other systems. The web applications that use images in repositories are either closed commercial systems, or one-off custom applications that are expensive to maintain.

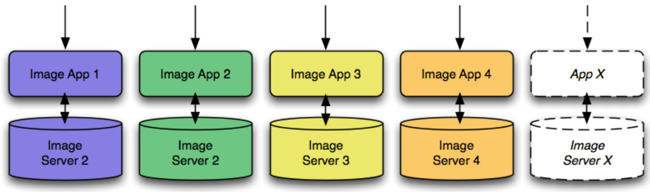

Figure 1: *Image repositories are implemented as technological silos, with image servers and client applications tightly coupled.*

The IIIF was conceived of to address this problem of silos and duplication by leveraging the power of common APIs to promote simplified infrastructure and reusable and interchangeable parts. APIs enable reuse of data by multiple applications or clients. If both images and metadata are made available by repositories using simple REST APIs, the same data could be used for many purposes: simple image presentation, enhancement with annotations and transcriptions, or embedding images in blogs and other third party web applications.

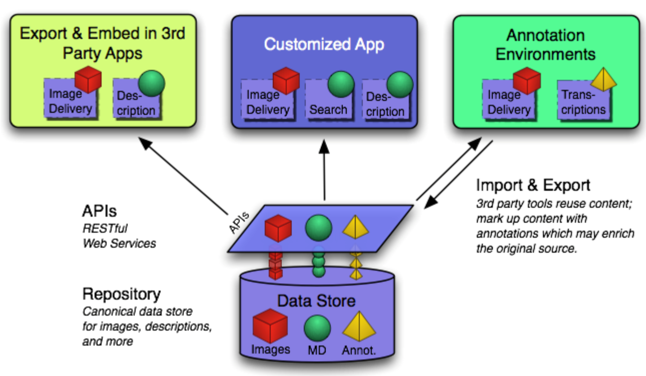

Figure 2: *Common API's allow for reuse of data by multiple applications or clients.*

If multiple repositories supported the same APIs for access to images, metadata and annotations, one can envision the beginnings of an ecosystem that leverages shared application code across different sites, and interoperable image-based resources, enabling simultaneous access to multiple scholarly repositories through a single interface.

With these concepts in mind, IIIF aims to reduce inefficiency and needless redundancy born from incompatibility in the image delivery ecosystem, and promotes the building of a global and interoperable framework by which image based resources could be easily shared and reused across institutions using any combination of image servers or client viewing software.

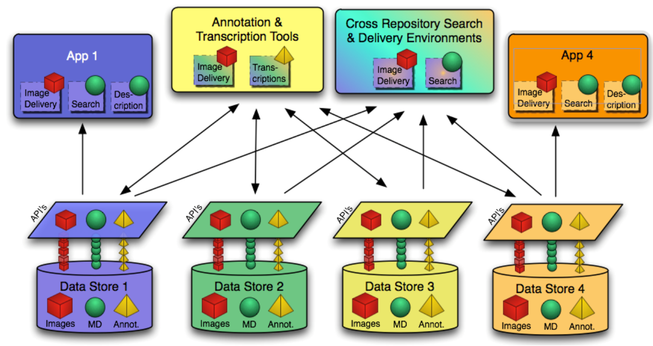

Figure 3: *Multiple repositories supporting common API's enables true interoperability between servers and clients.*

Technical Framework

IIIF, at the time of writing, has two APIs. The Image API provides access to the image content and technical descriptions. The Presentation API gives just enough structural and descriptive information about the image's context to appropriately render it in an arbitrary, web-based, viewing environment. Both APIs have been defined and vetted by an international community of stakeholders, with a primary emphasis on real-world use cases and technical environments.

Image API

The IIIF Image API [1] specifies a web service that returns an image in response to a standard HTTP or HTTPS request. The URI specifies the region, size, rotation, quality characteristics and format of the requested image. A URI can also be constructed to request basic technical information about the image to support client applications. The Image API was designed to facilitate systematic reuse of image resources in any digital image repository, and can be used to retrieve static images in response to a properly constructed URI.

The basic structure of any IIIF Image API request must include the URI scheme (http or https), the name of the image server host, an optional prefix identifying the specific IIIF service, and the identifier of the image resource being requested. If the request is for an image, then it must also include parameters for region, size, rotation, quality and file format. The URI template for an image request is:

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

or, by way of an example: http://www.example.org/image-service/abcd1234/full/full/0/default.jpg

The IIIF community has determined that combinations of the five basic parameters will supply images to web presentation software to support both basic and sophisticated features for display, navigation and manipulation of images on the web, and satisfy a significant percentage of common requirements. While there are many dozens of specialized properties of images that an image server might transform before delivering an image to a browser, IIIF has focused on the "80/20 rule" [2]— supporting the

20% of the functions that satisfy 80% of the needs for web-based, image viewing experiences. The five basic parameters are:

- **Region**: The region parameter defines the rectangular portion of the full image to be returned. Region can be specified by pixel coordinates, percentage or by the value "full", which specifies that the entire image should be returned. The region request makes it simple to request square tiles of images for image-pyramid based zoom and pan clients. It also enables delivery of arbitrarily cropped regions of images to browsers without having to make separate files from the original.
- **Size**: The size parameter determines the dimensions to which the extracted region is to be scaled, and can be expressed in pixels or percentages. The parameter can be constructed to maintain aspect ratio or allow for distortion of the image if height and width are expressed explicitly. A request can also be formatted to deliver the size of image that best fits within a specific height and width. This parameter makes it simple for a web application to offer a variety of sizes of an image from a single source.
- **Rotation**: The rotation parameter specifies mirroring and rotation. A leading exclamation mark ("!") indicates that the image should be mirrored by reflection on the vertical axis before any rotation is applied. The numerical value represents the number of degrees of clockwise rotation, and may be any floating point number from 0 to 360. Rotation and mirroring are useful for allowing the user to appropriately orient an image that may have been photographed at the wrong orientation or perspective.
- **Quality**: The quality parameter determines whether the image is delivered in color, grayscale or black and white. This allows for basic degradation of an image for performance or differential rights. Certainly from an image science perspective this parameter is overly simplified. However, most common applications don't require more sophisticated quality transformations, and for broad adoption a basic approach was taken.
- **Format**: The format of the returned image is expressed as an extension at the end of the URI. The most common formats are supported and include jpeg, tiff, png, gif, jpeg2000, pdf and webp.

The sequence of the parameters in a request is critical, because applying the same parameters in a different sequence will often result in the delivery of a different image than is expected. The image manipulations performed by the image server before delivering the image follows the same sequence as the parameters in the URL:

Region THEN Size THEN Rotation THEN Quality THEN Format

Figure 4 illustrates the order of operations that the IIIF Image API applies to a base image. Different combinations of parameters applied in a set pattern allow remote image clients to request dynamic views of a base file, suited to particular display needs.

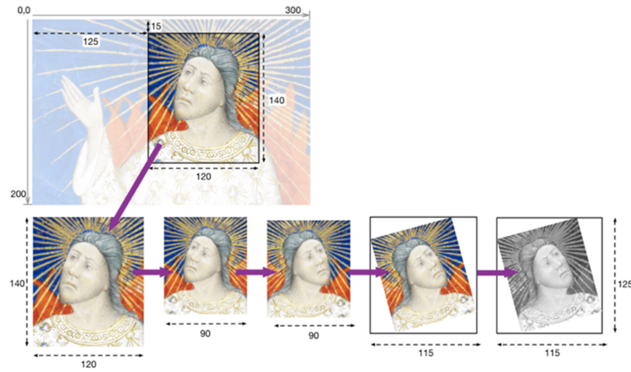

***Figure 4**: The IIIF image API specifies region size, rotation, quality and file format. In this case: region=125,15,120,140 size=90, rotation=!345 quality=gray .../125,15,120,140/90,/!345/gray.jpg . Graphic by Christopher Jesudurai.*

A second form of the image API request can be made, not for an image, but rather for basic technical information about the image. This is called the Image Information Request, and is expressed in the format:

{scheme}://{server}{/prefix}/{identifier}/info.json

or, by way of example: http://www.example.org/image-service/abcd1234/info.json

The Image Information document contains both metadata about the image, such as full height and width, and functionality available for it, such as the formats in which it may be retrieved. Its primary purpose is to give image clients sufficient technical metadata in order to anticipate how to efficiently request images from the image server,

The Image API supports the notion that different servers may implement IIIF at different levels of sophistication or compliance. This is intended to encourage broad adoption of the specification for even the most rudimentary image server. The lowest level of compliance, Level 0, only requires an image server to deliver a full sized jpeg image, with no rotation at its original quality level. Higher levels of compliance require support for variations and combinations of parameters. A server may also declare an arbitrary set of features that it supports.

Presentation API

The IIIF Presentation API [3] specifies a web service that returns JSON-LD structured documents that together describe the structure and layout of a digitized object or other collection of images and related content. Rather than trying to implement a new universal metadata standard, the IIIF Presentation API focuses on the metadata most critical for driving a remote viewing experience, such as image sequence, display labels, licensing, attribution, and links to "more information". Many different styles of viewer can be implemented that consume this information to enable a rich and dynamic user experience, consuming content from across collections and hosting institutions. Existing viewers include page turning applications, simple gallery views, and rich cross-institutional comparison systems.

The API is based on the Shared Canvas data model [4], which consists of the following types of resources:

- **Collection**: A group of objects and/or sub-collections. Collections are intended for discovery and navigation between objects.
- **Manifest**: The resource that groups the images, metadata and other content that make up a particular object. The definition of object is left intentionally vague to allow a born-digital comic book, a digitized scroll, or a slideshow to all fit into the same space.
- **Sequence**: A particular order of the sub-components of an object which are depicted by images. An object might have multiple sequences when, for example, it represents a manuscript that was rebound with the quires in a different order.
- **Canvas**: A blank space, similar to an empty PowerPoint slide or blank physical canvas, on to which digital content resources are "painted." The content might be text, images, video, audio or any other representation of the object.
- **Annotations**: As the content may be distributed between multiple institutions, and provided without the explicit knowledge of the institution hosting the Canvas, content is painted by annotation. The annotations might be discovered by a dynamic service, or in a more straightforward implementation scenario simply included within the JSON-LD response for the Manifest. The Annotations of images then include a reference to the Image API service from which tiles and other manipulations are available.
- **Range**: These allow for the description of parts of the object, for example to create a table of contents within a book, or to link discontinuous article sections in a newspaper. There are no semantics for a Range, allowing them to be used for both physical and textual segmentation at the same time.
- **AnnotationList**: As there may be many annotations linking small segments of textual content, such as a line or even a single word, to its location in the Canvas, there is a grouping construction for these Annotations so they may be retrieved together in a single call.
- **Layer**: Similarly, AnnotationLists can be grouped into Layers that span multiple Canvases, such as all of the Annotations that make up a transcription as opposed to a translation of the text depicted.

Although any of the resources in the model above may have their own URIs, the Presentation API recommends three primary interactions. The most commonly implemented is the Manifest, with the default sequence and all of the annotations of image content embedded within it. AnnotationLists are then linked to from the Manifest on a Canvas-by-Canvas basis allowing the client to retrieve them only if the user navigates to that particular canvas. This pattern was decided upon after observing how existing clients made use of pre-IIIF data structures. Finally Collections should be a tree of individual interactions, as the hierarchy is unable to be predicted between institutions and use cases.

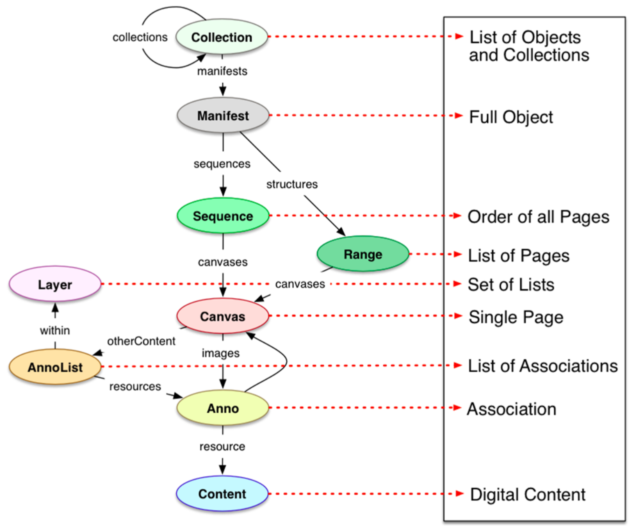

Figure 5*: The IIIF Presentation API is based on the Shared Canvas data model which specifies the relationships between resources.*

Software

Any web server can deliver an image file over HTTP, and provided the image is of the right file format (jpeg, png, gif) modern web browsers will render a static version of that image. Light-weight browser plugins, or more complex CSS, might empower the browser to render additional formats or provide more sophisticated functionality such as basic zoom and pan, rotation, or simple transformations. As imaging technology has evolved to produce much larger image files, and the potential uses of images on the web have become more sophisticated, a class of servers and clients has emerged that is dedicated to efficient and feature rich delivery of images on the web. Image servers are web servers that not only deliver images to the web, but also apply some transformation to the image to support enhanced functionality. For example, dynamic tile-based image servers optimize for zoom and pan by assembling small square regions of the high resolution source image to fit only the viewport and chosen zoom level requested. Dedicated image viewing clients know how to request these tiles based on user selections.

Since the release of the first version of the IIIF APIs an increasing number of existing servers and clients have been adapted to support the IIIF specification, and even some new software has been built to fully leverage the framework.

Image Servers

In the cultural heritage image repository space, a relatively small number of image servers have been adopted by institutions that deliver images to digital collection sites and online visual resource catalogs. Of these, the following two image servers have been adapted or built anew to deliver images in response to IIIF Image API requests:

- **Loris** - Built originally as a demonstration of IIIF's first image API, Loris has become a compelling image server option for cultural heritage institutions. Developed at

Princeton University, Loris is a native IIIF image server, built in Python and available as open source [5].
- **IIPImage Server** - IIPImage Server has long been a mainstay open source image server offering that has broad adoption by museums, libraries and many other image cultural heritage image services. Written in C++, IIPImage Server supports advanced image features such as 16 and 32 bit color depths, floating point data, CIELAB colorimetric images and scientific imagery such as multispectral or hyperspectral images. IIPImage Server has been adapted to be compliant with the latest IIIF API's [6].

In addition to Loris and IIPImage, several other image servers have had adapters built by third parties to translate their image delivery syntax into IIIF. Translators have been built for many products including Aware, ContentDM, Djatoka, FSI server, LuraTech, Chronicling America's image server, the Cloudinary image hosting service and others.

Image Viewing Clients

The landscape of image viewing technologies for cultural heritage is extensive, but each new offering reinvents at great expense the previous iteration. Newly developed clients often add an incremental step up in functionality, but in many cases just reimplement what has come before with a slightly different user experience or branding. IIIF enables clients to support necessary features, such as multi-image object rendering, pan, deep zoom, and annotation, while following a standard set of interaction patterns across multiple content repositories. A number of image viewers and clients have been built or adapted to support the IIIF framework. Some notable examples include OpenSeadragon, Leaflet, IIPMooViewer, OpenLayers, the Internet Archive BookReader and the Wellcome Player. We focus on one, Mirador, which has been developed by multiple institutions within the community.

Mirador

Partners within the IIIF community have built an image viewing, comparison and annotation platform that leverages all aspects of the IIIF framework. Mirador [7] is an open-source, web based, multi-window, image viewing platform with the ability to zoom, display, compare and annotate images from around the world. It retrieves images via the IIIF Image API, represents their structure and metadata via the IIIF Presentation API, and creates and reads comments and transcriptions in the Open Annotation [8] format. It is based on web standards and best practices, leveraging Javascript and HTML5.

Mirador has its origins in the digital medieval manuscript scholarly domain, as a way to compare and contrast page images hosted by different libraries, or as a way to reassemble manuscripts from fragments owned by multiple institutions. It was quickly realized that this utility was not limited to just medieval content and the entire cultural heritage sector would benefit from having access to its capabilities.

Mirador's key features implement the users' needs and desires for interacting with image based cultural heritage content. Seamless deep zoom and pan of images is the most loved feature, allowing users to inspect the tiniest details of objects, often far more accurately than if the object was physically in front of them. Comparison requires multiple objects to be visible at once, which Mirador enables with a flexible and intuitive layout system. The image rendering is done by the OpenSeadragon library, but the layout feature includes multiple viewing modes for paged objects, single images, thumbnail strips and tables of contents for navigation within objects.

In order to bring the objects into the viewing environment, Mirador offers an easily configurable list of objects from multiple repositories. Alternatively, the URL of an object can be entered directly. Once fully loaded, the layout of the workspace, down to the specific zoom level and position of the image content, can be bookmarked and shared. This functionality is especially useful in teaching and learning situations, or for providing evidence of a hypothesis in research situations.

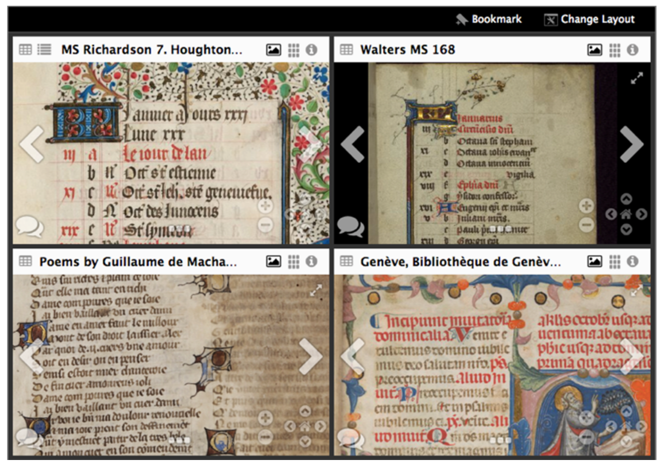

***Figure 6**: The Mirador viewer allows the researcher to compare and annotate images from geographically distributed image repositories.*

Rectangular areas of the image can be annotated, those annotations stored in different repositories, and then retrieved for viewing. A key distinction is between commentary about the object and the transcription of the text that the object carries, and while both use annotations, Mirador can distinguish and present them appropriately. Finally, it is able to be embedded, is responsive to small form-factor devices, and is touch event enabled.

Mirador demonstrates the kind of infrastructure that collaborative research communities can take advantage of if they agree to share and open their repositories to their peers. What used to be impossible—viewing all the shards of a manuscript reassembled from dozens of institutions or comparing multiple images from repositories on the opposite side of the globe—is now simple with IIIF and Mirador.

Community

Interoperability will only have the desired impacts on transformative uses of images on the web if a large number of cultural institutions and image services adopt and implement the technical framework, develop compatible software, and advocate for its uses and benefits. The IIIF community originated in the library world, as a small number of research and national libraries recognized that they were reinventing the wheel at great expense while not supporting the types of sophisticated comparative research scholars wanted to conduct with their images. The community continues to grow, not just in number but also in diversity as archives, museums, non-profit aggregators and commercial firms joined the effort.

Participation in the IIIF community can take many forms. Representatives from a large number of institutions have had a hand in defining the APIs and providing feedback and use cases to support their development. Commercial firms, institutions and individual open source developers have contributed to the growing market of image server and client software that have been developed or adapted to support the APIs. Representatives from over a dozen institutions meet twice a year as a working group to advance the agenda of IIIF, further enhance and refine the technical framework, and support and promote new adoption.

The community has a thriving and active email distribution list with more than 150 participants, and hosts an open community call every other week where new use cases are described, software demonstrated and issues discussed.

Future Work

As with any community and technical ecosystem, there is a drive to make further progress. The community is considering new APIs to support related areas of broad interest. In 2015 these primary areas of work include (1) authentication of access to images supplied by the Image API, (2) searching texts and annotations associated with objects and described by the Presentation API and the (3) discovery of IIIF resources across the web.

The authentication work aims to be as minimally intrusive into the workflows and requirements of image servers as possible, while still allowing clients to have some expectation of interoperability. This will allow institutions to expose their image-based resources via IIIF even if they are not open access, and offer some level of login or differentiated access (for example a high resolution vs. a watermarked or lower-resolution image) for authorized or anonymous users.

Searching within objects will support a wide class of well-established use cases, including full text search of books and newspapers with hit-highlighting. It will also support an emerging set of needs around the transcription, translation, annotation or commentary of image-based resources.

Finally, discovery of interoperable images on the open web will become increasingly important as the number of interoperable resources (books, images, maps, newspapers, manuscripts, architectural plans, etc.) grows into the tens and hundreds of millions. We plan to adopt common commercial practices by leveraging the Google site-maps specification to allow the enumeration of images and objects that a repository makes available. This allows systems to crawl and find the content that they are interested in, and then create appropriate interfaces and navigation functionality for their users.

It is also interesting to consider that the potential of interoperable digital data repositories is not limited to still images. We can imagine powerful applications if the same principles are applied to other digital media formats, including 3D renderings and time-based media (audio and video). The museum community has expressed considerable interest in applying IIIF principles, models and APIs to its collections.

Conclusion

We have described the three main facets of IIIF: its technical framework, software and community organization. IIIF has emerged over the past three years as an international community of cultural heritage organizations that agree on specifications for interoperability and provide both open source and commercial implementations of those APIs. The result is a flourishing ecosystem of tools and interoperable content that supports diverse and sophisticated use cases for image analysis, comparison, and annotation. We are already seeing IIIF's impact on both teaching and scholarly workflows as new methods of inquiry and discourse about visual resources are now available by virtue of interoperable content. While its origins are in libraries, the community is rapidly expanding to include museums, archives and image services of all types, creating new opportunities for exchange and collaboration across sectors. By adopting the IIIF and becoming part of the community, institutions gain access to well supported and sustainable technologies, and enrich scholarly use of their materials.

References

[1] IIIF Image API 2.0. http://iiif.io/api/image.

[2] Pareto principle. In *Wikipedia*. http://en.wikipedia.org/wiki/Pareto_principle.

[3] IIIF Presentation API 2.0. http://iiif.io/api/presentation.

[4] Shared Canvas Data Model. http://iiif.io/model/shared-canvas.

[5] Loris. https://github.com/loris-imageserver/loris.

[6] IIPImage Server. http://iipimage.sourceforge.net/.

[7] Mirador. http://projectmirador.org.

[8] Open Annotation Collaboration. http://www.openannotation.org.

Acknowledgements

This paper would not have been possible without the ideas and contributions of founding IIIF contributors and editors, especially Benjamin Albritton, Mike Appleby, Neil Jefferies, Matthew McGrattan, Jon Stroop and Simeon Warner.

Author Biographies

Stuart Snydman is Associate Director for Digital Strategy at the Stanford University Libraries (SUL). He directs SUL's digital library access program, overseeing development of its various discovery and delivery services. He has overseen SUL's digital imaging program since 2002. He is an editor of the first version of the IIIF Image API. He received his BA in history from the University of Virginia (1992) and his MA in Education from Stanford University (1995).

Dr. Robert Sanderson is an information scientist in Stanford University Libraries. He received his PhD in French and History from the University of Liverpool (2003). His research focuses on digital libraries, archives and museums and their interaction via linked open data and the web. He is an editor of the IIIF specifications. He was won international awards for his research in this area, including the 2010 Digital Preservation Award and the Vannevar Bush Best Paper award at JCDL2011 for his work on what has become the IIIF Presentation API.

Tom Cramer is the Chief Technology Strategist at the Stanford University Libraries. In addition to directing Stanford's digital library endeavors, he is the managing editor for IIIF, and the original scribe at the Cuban restaurant where IIIF was born.