Building Science Gateways for Humanities

Jun Zhou

Research Computing, University of South Carolina, South Carolina, USA, zhouj@mailbox.sc.edu

Karen Smith

Department of Natural Resources, University of South Carolina, South Carolina, USA, smithky@dnr.sc.gov

Greg Wilsbacher

Moving Image Research Collection, University of South Carolina, South Carolina, USA, gregw@mailbox.sc.edu

Paul Sagona

Research Computing, University of South Carolina, South Carolina, USA, sagona@mailbox.sc.edu

David Reddy

Research Computing, University of South Carolina, South Carolina, USA, reddy@mailbox.sc.edu

Ben Torkian[†]

Research Computing, University of South Carolina, South Carolina, USA, torkian@mailbox.sc.edu

[†] Corresponding author

ABSTRACT

Building science gateways for humanities content poses new challenges to the science gateway community. Compared with science gateways devoted to scientific content, humanities-related projects usually require 1) processing data in various formats, such as text, image, video, etc., 2) constant public access from a broad audience, and therefore 3) reliable security, ideally with low maintenance. Many traditional science gateways are monolithic in design, which makes them easier to write, but they can be computationally inefficient when integrated with numerous scientific packages for data capture and pipeline processing. Since these packages tend to be single-threaded or nonmodular, they can create traffic bottlenecks when processing large numbers of requests. Moreover, these science gateways are usually challenging to resume development on due to long gaps between funding periods and the aging of the integrated scientific packages. In this paper, we study the problem of building science gateways for humanities projects by developing a service-based architecture, and present two such science gateways: the Moving Image Research Collections (MIRC) – a science gateway focusing on image analysis for digital surrogates of historical motion picture film, and SnowVision - a science gateway for studying pottery fragments in southeastern North America. For each science gateway, we present an overview of the background of the projects, and some unique challenges in their design and implementation. These two science gateways are deployed on XSEDE's Jetstream academic clouding computing resource and are accessed through web interfaces. Apache Airavata middleware is used to manage the interactions between the web interface and the deep-learning-based (DL) backend service running on the Bridges graphics processing unit (GPU) cluster.

CCS CONCEPTS

• Software and its engineering • Human-centered computing • Applied computing

KEYWORDS

Science Gateways, Humanities, Deep-Learning, Angular, Java Play platform

1 Introduction

Science Gateways are virtual environments that accelerate scientific discovery by enabling scientific communities to more efficiently and effectively utilize distributed computing and data resources. Successful Science Gateways provide access to sophisticated and powerful resources, while shielding their users from the underlying complexities. Jetstream, the scalable cloud environment for XSEDE, maintained by the Texas Advanced Computing Center (TACC), enables science gateways to be easily deployed on customized virtual machines and other computing resources with the assistance of consultants from the Science Gateways Community Institute (SCGI).

XSEDE Science Gateway has attracted attention from many researchers in various fields, including Digital Humanities (DH). Descended from the field of Computational Humanities, DH is an interdisciplinary area at the intersection of computing or digital technologies and the disciplines of the humanities, such as history, library, archeology, etc. It includes the systematic use of digital resources in the humanities, as well as the analysis of their applications. Typical DH applications include digitization and preservation of heritage artifacts, such as the manuscripts of Edmund Spenser [1] and pottery sherds from Woodland periods [3]; visualization of historical objects, e.g., the Virtual Forbidden City [3]; and development of analytic tools, e.g., the Paragon image collation software [4]. Building humanities science gateways can be challenging and non-trivial for several reasons:

1) Digitized objects can be in various formats, such as text, image (2D and 3D), video, musical scores, etc. Transferring and manipulating video and image data over low-bandwidth internet requires intelligent use of optimization techniques.
2) These applications are usually expecting access from a larger public audience than typical science gateways, such as a community of historians and many history lovers, or K-12 students. Therefore, parallelization is essential to the success of these science gateways.
3) The support systems for humanities science gateways are usually less reliable because they are usually developed and maintained by short-term professionals.

In this paper, we present two science gateways built for humanities: SnowVision - a science gateway on studying pottery fragments in southeastern North America, and the MIRC – a science gateway focusing on image analysis for digital surrogates of historical motion picture film. For each science gateway, we present an overview of the background of the projects, along with some unique challenges in their design and implementation, and how we addressed these challenges.

2 SnowVision Science Gateway

2.1 Background

Our team of archaeologists, computer scientists, and digital humanists unites outstanding problems in art and archaeology and cutting-edge computer-vision research – an example of digital humanities par excellence – to automate the matching of stamped pottery sherds in order to reconstruct this lost yet immeasurably valuable southeastern Native American art form [5]. This art form is comprised of connected and intertwined curve pattern designs carved on stamps and has been practiced for more than two thousand years. Called "SnowVision", our science gateway will enable individuals and institutions both to upload and store sherd images, and to use our machine-learning based matching algorithms to identify the original stamped designs from which their fragments descend. The SnowVision project will enrich sherd data for researchers, while also creating a "virtual" place which centralizes a new, richer, collective knowledge about the diversity, evolution, and geographical reach of stamped pottery designs over time.

SnowVision assists researchers in the study of designs on paddle-stamped pottery that dates from AD 1-800, such as the sample pottery sherds and designs shown in Fig. 1. Some 1,820 archaeological sites with such pottery have been recorded across the present-day states of Alabama, Florida, Georgia and South Carolina. Hundreds of thousands of pottery sherds from these sites reside in curation facilities and laboratories. Before, making sense of these large and scattered collections was a daunting task for researchers. We are using 3D scanner to digitize these sherds. To date, more than 4,000 sherds have been digitized and 25,000 of them are on the way, and about 900 reconstructed designs were digitized and added into our collection. Our computing team has developed a deep-learning-based design matching method to automatically identify these pottery sherds and further help archaeologists analyze and understand their collections. As the project goes forward, it is imperative to build a science gateway to share these collections and studies among archeologists.

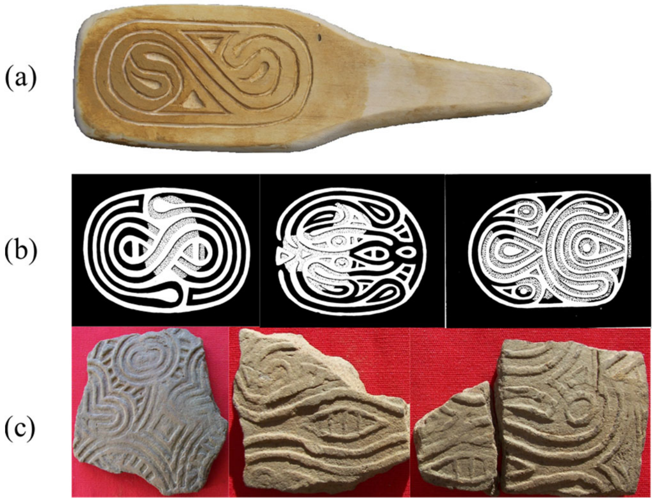


Figure 1: (a) Reconstructed sample paddle stamp. (b) Reconstructed sample paddle designs. (c) Sample pottery sherds.

2.2 Challenges

Many traditional science gateways are monolithic in design, which makes them easy to write, but they can be computationally inefficient to integrate with numerous scientific packages for data capture and pipeline processing. Typically, these packages are single-threaded or nonmodular, which can create traffic bottlenecks when processing many requests. To address these challenges, we built the SnowVision Science Gateway with these modules: a UI frontend service (Angular), an API backend service (Java Play platform), an authentication service (Auth0), a middleware service (Apache Airavata) and an AI backend sherd-to-design matching service (Caffe). In the following, we describe the architecture of our science gateway, the implementation of each module, and continuing support for security and maintenance.

2.3 Design and Implementation

The architecture of the SnowVision Science Gateway is illustrated in Fig. 2. The SnowVision UI service, along with its API backend service, authentication service and storage are hosted on a TACC Jetstream VM. The middleware service connects the API backend service to the AI backend sherd-to-design matching service, running on Bridges GPU clusters.

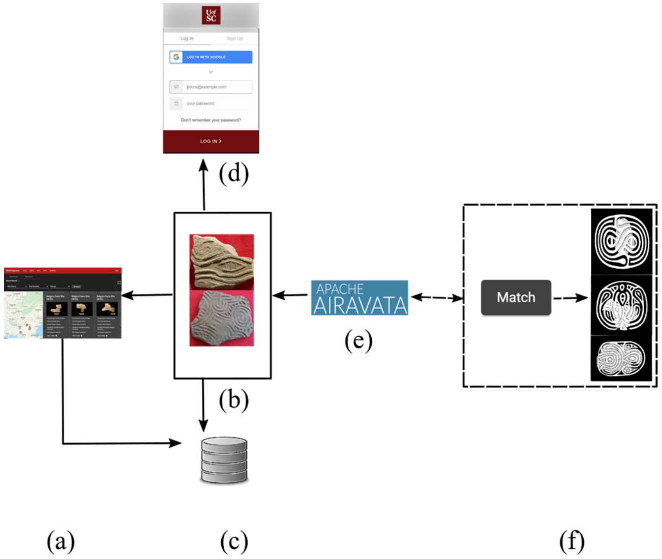
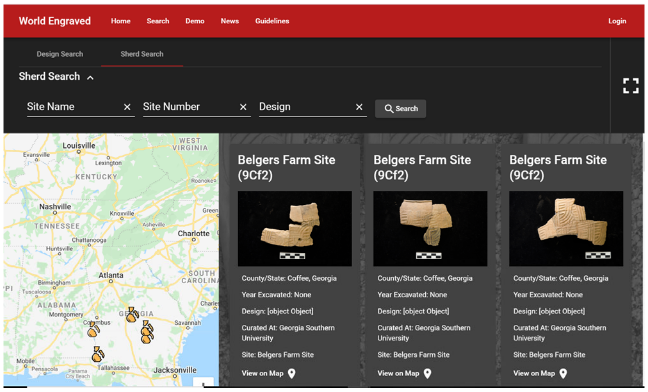


Figure 2. The architecture of the SnowVision Science Gateway: (a) SnowVision UI frontend service, (b) SnowVision API service, (c) storage, (d) SnowVision authentication service, (e) Middleware service (Apache Airavata API), and (e) SnowVision AI backend sherd-to-design matching service.

2.3.1 UI Service

Called worldengraved.org, the SnowVision website allows a user to query pottery sherd and design collections, as a sample page shown in Fig. 3.

A user would be able to manage his own collections through the user profile, in which the user can also submit unidentified sherds to the SnowVision AI sherd-to-design matching service to find their corresponding designs. For example, we illustrate three designs that match the sample sherd shown in Fig. 4.



Belgers Farm Site (9Cf2)

County/State: Coffee, Georgia

Year Excavated: None

Design: [object Object]

Curated At: Georgia Southern University

Site: Belgers Farm Site

View on Map

Belgers Farm Site (9Cf2)

County/State: Coffee, Georgia

Year Excavated: None

Design: [object Object]

Curated At: Georgia Southern University

Site: Belgers Farm Site

View on Map

Belgers Farm Site (9Cf2)

County/State: Coffee, Georgia

Year Excavated: None

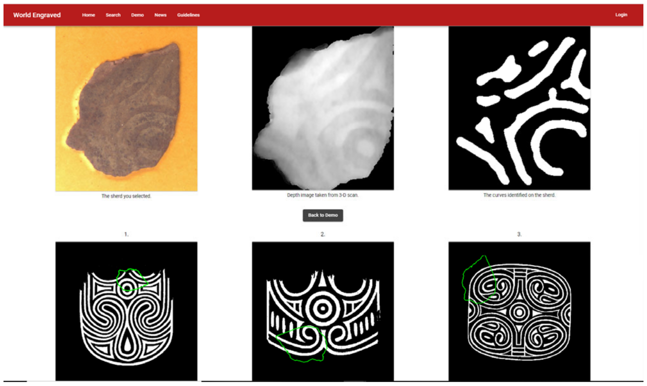
Design: [object Object]

Curated At: Georgia Southern University

Site: Belgers Farm Site

View on Map

Figure 3. Query page and sample results of a sherd query



The sherd you selected.

Depth image taken from 3-D scan.

The curves identified on the sherd.

Figure 4. Matching result of a sample sherd. The top row (from left to right) shows the RGB image of a sherd, the depth image from 3D scanner of this sherd, and extracted curve structure of this sherd. The bottom row (from left to right) shows three best possible matches to designs. Original design reproduced with permission, courtesy of Frankie Snow, South Georgia State College.

The SnowVision UI service is implemented using Augular[1][6], a single-page app made for dynamic web applications. It offers a high level of abstraction, eliminating the need for redundant code, hence making the creation of dynamic web applications much simpler versus plain HTML and CSS. Moreover, it can keep the interface separate from the backend API service, which allows for a better workflow during development, as each module can be worked on and changed separately without affecting the other.

2.3.2 API Service

The SnowVision API service is developed using the Java Play Framework[2] [7], an open-source architectural pattern that helps to create lightweight, and highly scalable web applications. The pattern the Play Framework uses is exceptionally scalable due to its fully asynchronous design. By handling every request asynchronously and using short, non-blocking operations, Play can quickly scale with a small fixed thread pool. The feature allows for multiple users to make the non-blocking request at the same time without being forced to wait for each other's operations to complete. This proves to be extremely scalable given that, as the user base for a site grows, the potential slowdown from repeated requests is lessened.

The SnowVision API service interacts with the AI backend sherd-to-design matching service using the Apache Thrift protocol[3]. Given the vast number of languages Apache Thrift supports, any assortment of separate applications could be used in conjunction with Java Play. Therefore, deploying this module for other projects can be relatively simple.

2.3.3 User Authentication Service

To allow users to create accounts as well as to keep track of user data we make use of Auth0[4], an easy to implement user authentication service. Auth0 uses cookies and tokens to authenticate a user. Cookies are stored locally, which removes the risk of overloading web server. Tokens are assigned by an external remote authentication server. Auth0 is a modern and convenient solution to the difficult yet vital problem of user authentication. We are using the free plan of Auth0 service, which allows user authentication, but not authorization. Concerns may be raised when Auth0 changes its pricing policy, or upgrading/deprecating its

[1] https://angular.io/

[2] https://www.playframework.com/

[3] https://thrift.apache.org/

[4] https://auth0.com/

APIs. However, it was well worth the effort, as we handed the security over to professionals. Moreover, alternatives are emerging, for example, the free open source CIlogon5. Auth0 provides authentication an API which allows user to manage all aspects of user identity. It offers endpoints so users can log in, sign up, log out, access APIs, and more. The API supports various identity protocols, like OpenID Connect, OAuth 2.0, and SAML. By using Auth0, we can be sure that security and authentication are handled properly, without the hassle of implementing and maintaining a reliable, secure system by ourselves.

2.3.4 Middleware Service

If a user wants to find designs that correspond to unidentified sherds, the API service will submit a request along with the sherds' files through a middleware service implemented using the Apache Airavata6 API [8] (Pierce, Marru and Demeler). This service will compose, manage, execute, and monitor the jobs submitted remotely on the AI sherd-to-design matching service on the Bridges GPU cluster.

2.3.5 AI Backend Sherd-to-Design Matching Service

The AI sherd-to-design matching service matches unidentified sherds to a database of known designs. Users will submit sherds in a 3D point cloud format such as XYZ or STL. The matching service will first convert the point clouds to depth maps, and then extract curve patterns from these depth maps [9]. After that, the matching service will match the curve patterns to each design in the database [10][11][12]. If no design is found, we will present the sherds to users for manual assessment. Inside the matching service, the curve extraction and design matching are built on deep-learning models, which are run on Bridges GPU clusters. This service is implemented using the Point Cloud Library (PCL) for generating depth map from 3D sherd image files and Caffe [13] for curve extraction and sherd-to-design matching.

2.4 Security and maintenance

Most science gateways will stop development after the research funding ends and will only continue development if they get new funding. As a result, it is tough for successors to resume the work on the aging applications.

We have two approaches to help solve this issue. First, we use the Auth0 software-as-a-service (SAAS) for our user authentication, login, and session security, as well as the handling of forgotten passwords. This third-party service is actively maintained, and its security is regularly updated. Second, we use the Angular and Java Play frameworks that are heavily used by industry. The benefits of using these frameworks are two-fold, we can get quick updates in case of any security vulnerabilities, and the frameworks usually provide direct upgrade paths to newer versions78.

3 Moving Image Research Collections (MIRC) Science Gateway

3.1 Background

For almost a century, celluloid-based imagery was the dominant medium for recording the history of the world, creating a global library of still and moving images the full epistemological weight of which has yet to be felt. Even though still image digitization has reached a mature state, the archival digitization of motion picture film is still in a developmental phase because motion picture film is derived from a complex system and results in data intensive files. Ideally, future digital variants of celluloid film will have an information complexity that authentically mirrors their original sources. Given the rise of deep fake technology, it is

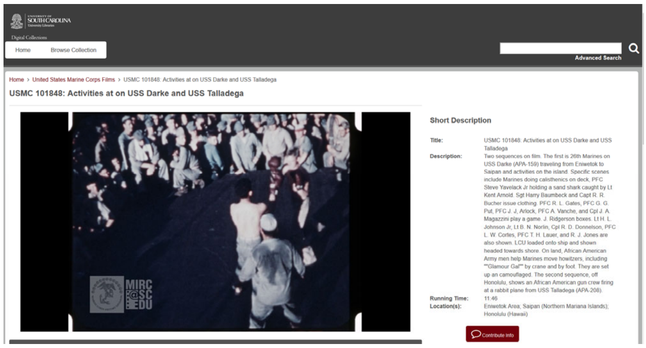
[5] https:// https://cilogon.org/

[6] https://airavata.apache.org/

[7] https://www.playframework.com/documentation/2.8.x/Migration28

[8] https://update.angular.io/#8.0:9.0

essential that mature systems are developed soon. Moving Image Research Collections (MIRC) at the University of South Carolina recently entered a partnership with the United States Marine Corps History Division to preserve, digitize and make accessible the legacy 16mm and 35mm film collection housed at Marine Corps University, Quantico. The collection is very large, containing over 18,000 cans of film (a typical can contains 7 to 8 minutes of footage). MIRC has been scanning films at 2K (2048 x 1531 pixels) and currently has over 2,000 digitized films, typically scanning 60 to 75 cans per week. Digitizing films is only one component of the project. This collection has a high research value for historians in many fields and is of sufficient size to create a data set able to train image analysis algorithms. MIRC seeks to identify new methods for deploying these digital film assets as trusted historical resources (in contrast to the chaos of user-contributed online video). To accomplish this, MIRC is partnering with the university's Computer Vision Lab (led by Dr. Song Wang) and Research Computing at the University of South Carolina to develop and deploy three initial projects: Deep Learning algorithms for identifying and tracking textual information in historical imagery; DL algorithms for facial recognition in historical imagery; and a new method for certifying the chain of historical provenance from a celluloid film to a master digital surrogate copy, and then to all subsequent copies derived from that master. The MIRC science gateway is built as a virtual home which not only hosts the online collections for the public, but also allows researchers and developers to collaborate with others and experiment with mechanisms for above mentioned image/video analysis projects, a sample page shown in Fig. 5.



Home > United States Marine Corps Films > USMC 101848: Activities at on USS Darke and USS Talladega

USMC 101848: Activities at on USS Darke and USS Talladega

Short Description

Title:	USMC 101848: Activities at on USS Darke and USS Talladega
Description:	Two sequences on film. The first is 26th Marines on USS Darke (APA-159) traveling from Eniwetok to Saipan and activities on the island. Specific scenes include Marines doing calisthenics on deck, PFC Steve Yavelack Jr holding a sand shark caught by Lt Kent Arnold. Sgt Harry Baumbeck and Capt R. R. Bucher issue clothing. PFC R. L. Gates, PFC G. G. Put, PFC J. J, Arlock, PFC A. Vanche, and Cpl J. A. Magazzini play a game. J. Ridgerson boxes. Lt H. L. Johnson Jr, Lt B. N. Norlin, Cpl R. D. Donnelson, PFC L. W. Cortes, PFC T. H. Lauer, and R. J. Jones are also shown. LCU loaded onto ship and shown headed towards shore. On land, African American Army men help Marines move howitzers, including ""Glamour Gal"" by crane and by foot. They are set up an camouflaged. The second sequence, off Honolulu, shows an African American gun crew firing at a rabbit plane from USS Talladega (APA-208).
Running Time:	11:46
Location(s):	Eniwetok Area; Saipan (Northern Mariana Islands); Honolulu (Hawaii)

Figure 5. Sample page of MIRC science gateway.

3.2 Challenges

The major concern in building the MIRC science gateway is the need to provide fast video streaming while preventing unauthorized video downloading. Currently, digitized videos in the collection are in .mov and .dpx formats. Each video is average of 12 minutes long, at 2048x1531 pixels per frame, 24 frames per second, with an average file size 20 GB. Efficient streaming of such videos online can be challenging. Additionally, unauthorized downloading of copyrighted media should be discouraged, if not avoided completely. To address these challenges, MIRC needed to carefully choose a streaming method and a video player. We studied and tested several existing video players and found that multipart streaming with Fluid Player9 works the best.

3.3 Design and Implementation

The architecture of the MIRC Science Gateway is illustrated in Fig. 6. It consists three modules: the MIRC UI frontend service (Angular), its backend API service (Java Play framework) and the authentication service (Auth0). Inside the backend API service, we use REST API to listen to HTTP requests. The MIRC Science Gateway is implemented in a TACC Jetstream VM.

[9] https://github.com/fluid-player/fluid-player

3.3.1 UI Service

The MIRC website holds digitized film collections of US Marine Corps. It allows a user to search our archives by various criteria, such as dates, titles, hull numbers of vessels, names of historical figures or general keywords in descriptions. The MIRC UI service is implemented using Angular for dynamic web programming and separation from the backend API service for better modular development. The videos are streamed on to Fluid Player, an open source HTML5 player using HTTP Live Streaming (HLS) protocol, an HTTP- based adaptive bitrate streaming communications protocol developed by Apple Inc.. Support for this protocol is widespread in media players, web browsers, mobile devices, and streaming media servers.

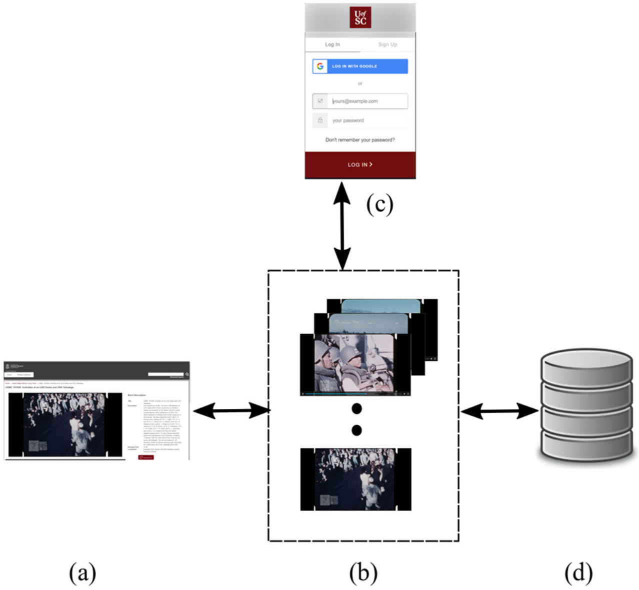


Figure 6. The architecture of MIRC Science Gateway: (a) MIRC UI frontend service, (b) MIRC backend API service, (c) MIRC authentication service and (d) storage

3.3.2 API Service

The MIRC API service uses the REST API to accept requests from the UI and send responses back. The REST API is implemented in the Java Play framework. As mentioned in Session 2.3.2, the pattern the Play Framework uses is exceptionally scalable due to its fully asynchronous design. The Play Framework allows developers to follow the Model-View-Controller pattern. In MIRC, we implement view and controller, in which the view is used to render the web application, and the controller is responsible for responding to events within the rendered web pages, as well as processing these events and potentially making changes to the pages. The controller uses the REST API to listen to HTTP requests and determine if changes to the currently rendered page are needed or if a new render is required.

The API service uses HLS to convert an MP4 file to multiple MPEG transport stream files. When creating these MPEG transport stream files, an M3U8 file is also created. This M3U8 file is a UTF-8 encoded playlist file that determines the order in which the MPEG transport stream files will buffer and play. Each MPEG transport stream file is a separate HTTP request by the client. This not only makes the video streaming more secure, due to the much larger number of video segments being sent, but allows for the HTTP responses to have far more control over how the video plays. With each video being played in small sections, we can seamlessly change video resolution when required, and accurately keep track of what section a user is currently watching. To reduce the traffic in live streaming, we preprocess each video by first converting it to MP4 format and then breaking it down to multiple files. All these files are kept in the storage.

3.3.3 Authentication Service

MIRC authentication uses the Auth0 service for user account creation and user data tracking. We reused the authentication service from the SnowVision Science Gateway, with minor changes to adapt it to the MIRC users.

3.4 Security and Challenges

Like the SnowVision Science Gateway, the security updates and maintenance overhead of the MIRC Science Gateway will heavily benefit in reduced development and maintenance overhead through using the commercial Auth0 service for user authentication, and the industry-standard Angular and Java Play frameworks.

3.5 Future Work

The MIRC Science Gateway is still under development. Currently we only work with QuickTime File Format (QTFF) ".mov" files, along with converted MP4 files. The demand for new video standard, such as the "smart" OpenEXR[10] [14], is emerging. As described in Section 3.1, the MIRC Science Gateway is the virtual home for several research projects involving automatic text recognition and extraction, facial classification and video provenance integration. Our goal is to realize automatic video understanding and description metadata generation. OpenEXR allows this information to be embedded into every frame of a video. In the future, our video streaming server and video player will need to deliver this information along with our streaming.

4 Conclusion

In this paper, we study the problem of building science gateways for humanities projects and present two such science gateways: the Moving Image Research Collections (MIRC) and SnowVision. For each science gateway, we present an overview of the background, some unique challenges, design and implementation, security and maintenance. For the SnowVision Science Gateway, we built a modular gateway using Angular for the frontend UI service, Java Play framework for the backend API service, Auth0 for user authentication, and Apache Airavata for middleware communication with remote GPU clusters, especially, Caffe and PCL, which are used for 3D image manipulation and the DL-based matching service. For the MIRC Science Gateway, we utilized Fluid Player with an HLS streaming server to help avoid unauthorized video downloading while optimizing streaming of large video files. We followed the same modular development used in the SnowVision Science Gateway in the MIRC Science Gateway for the frontend UI service, backend API service and user authentication service. The modules built for the SnowVision and the MIRC Science Gateways are highly re-deployable, and easily adapted to other similar science gateway projects.

ACKNOWLEDGMENTS

This work used the Extreme Science and Engineering Discovery Environment (XSEDE) Jetstream, Bridges resources, ECSS support through allocation DBS180011 for the SnowVision Science Gateway and allocation HUM190002 for the MICR Science Gateway. We would like to show our appreciation to Suresh Marru, Dr. Marlon Pierre, and Robert Quick at Indiana University for Science Gateway consultation. We are very grateful to Dr. Song Wang, Yuhang Lu for researching and developing the SnowVision Science Gateway sherd-to-design matching service. We would like to thank Deja Scott, Jake Bukuts and Andrew Tant at the University of South Carolina for developing the science gateways.

[10] https://www.openexr.com/

REFERENCES

[1] Owen, W. J. B. 1949. Edmund Spenser and the ‘Faerie Queene’. *Journal of the English Association* 7.42: 292–293.

[2] Anderson, David G. 2002. The Woodland Southeast. *The University of Alabama Press*.

[3] Tan, Beng-Kiang, and Hafizur Rahaman. 2009. Virtual heritage: Reality and criticism.

[4] Miller, David Lee, and Song Wang. 2015. PARAGON: Intelligent Digital Collation and Difference Detection.

[5] Smith, Karen Y., and Vernon J. Knight Jr. 2017. Swift Creek paddle designs and the imperative to be unique. *Southeastern Archaeology*. 36.2: 122-130.

[6] Chiaretta, Simone. 2018. Front-end Development with ASP. NET Core, Angular, and Bootstrap. John Wiley & Sons.

[7] Karunakaran, Prem Kumar. 2014. Play Framework 2-For Java Developers.

[8] Pierce, Marlon, Suresh Marru, Borries Demeler, Raminderjeet Singh, and Gary Gorbet. 2014. The Apache Airavata Application Programming Interface: Overview and Evaluation with the UltraScan Science Gateway. *Gateway Computing Environments Workshop*

[9] Yuhang Lu, Jun Zhou, Jun Chen, Jing Wang, Karen Smith, Wilder Colin, and Song Wang. 2018. Curve-structure segmentation from depth maps: A CNNbased approach and its application to exploring cultural heritage objects. *AAAI Conference on Artificial Intelligence.*

[10] Jun Zhou, Haozhou Yu, Karen Smith, Colin Wilder, Hongkai Yu, and Song Wang. 2017. Identifying designs from incomplete, fragmented cultural heritage objects by curve-pattern matching. *Journal of Electronic Imaging* 1.26: 011022– 011022.

[11] Jun Zhou, Yuhang Lu, Karen Smith, Colin Wilder, Song Wang, Paul Sagona, and Ben Torkian. 2019. A Framework for Design Identification on Heritage Objects. PEARC. 1-8.

[12] Jun Zhou, Haozhou Yu, Karen Smith, Colin Wilder, Hongkai Yu, and Song Wang. "Identifying designs from incomplete, fragmented cultural heritage objects by curve-pattern matching." Journal of Electronic Imaging 1.26 (2017): 011022– 011022.

[13] Jia, Yangqing, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. ACM *international conference on Multimedia*.

[14] Kainz, Florian, Rod Bogart, and Piotr Stanczyk. 2009. Technical introduction to OpenEXR. *Industrial light and magic* 21.