DATA SCIENCE JOURNAL

Almas, B 2017 Perseids: Experimenting with Infrastructure for Creating and Sharing Research Data in the Digital Humanities. *Data Science Journal,* 16: 19, pp. 1–17, DOI: https://doi.org/10.5334/dsj-2017-019

PRACTICE PAPER

Perseids: Experimenting with Infrastructure for Creating and Sharing Research Data in the Digital Humanities

Bridget Almas

Perseids Project, Tufts University, 5 the Green, Medford, MA 02155, US

balmas@gmail.com

The Perseids project provides a platform for creating, publishing, and sharing research data, in the form of textual transcriptions, annotations and analyses. An offshoot and collaborator of the Perseus Digital Library (PDL), Perseids is also an experiment in reusing and extending existing infrastructure, tools, and services. This paper discusses infrastructure in the domain of digital humanities (DH). It outlines some general approaches to facilitating data sharing in this domain, and the specific choices we made in developing Perseids to serve that goal. It concludes by identifying lessons we have learned about sustainability in the process of building Perseids, noting some critical gaps in infrastructure for the digital humanities, and suggesting some implications for the wider community.

Keywords: infrastructure; digital humanities; data sharing; interoperability; research data

Overview

The Perseids project provides a platform for creating, publishing, and sharing research data, in the form of textual transcriptions, annotations and analyses. An offshoot and collaborator of the Perseus Digital Library (PDL), Perseids is also an experiment in reusing and extending existing infrastructure, tools, and services.

This paper discusses infrastructure in the domain of digital humanities (DH). It outlines some general approaches to facilitating data sharing in this domain, and the specific choices we made in developing Perseids to serve that goal. It concludes by identifying lessons we have learned about sustainability in the process of building Perseids, noting some critical gaps in infrastructure for the digital humanities, and suggesting some implications for the wider community.

General Approaches

What constitutes infrastructure, and how does it facilitate data sharing in the domain of DH, and in the Perseids project in particular? According to Mark Parsons, Secretary General of the Research Data Alliance (RDA), infrastructure can be defined as 'the relationships, interactions and connections between people, technologies, and institutions that help data flow and be useful (Parsons 2015).'

In the realm of DH, any of the following might be considered infrastructure: original digital collections, linked data providers, general purpose and domain-specific platforms, content management systems (CMSs), virtual research environments (VREs), online tools and services, repositories and service providers, aggregators and portals, APIs and standards. **Table 1** provides some specific examples of these in the DH and digital classics (DC) communities, illustrating the diversity and breadth of infrastructure in this community.

Enabling data sharing includes ensuring that data objects have persistent, resolvable identifiers, providing descriptive and structural metadata, providing licensing and access information, and using standard data formats and ontologies. The recent W3C recommendation 'Data on the Web Best Practices' (Loscio, et. al. 2016) cites many strategies such as providing version history, provenance information, and data quality information.

Infrastructure type	Examples in DH and DC
Original digital collections	PDL, Papyri.info, NINES, Digital Latin Library, Coptic Scriptorium, Roman de La Rose
Linked data providers and gazetteers	Pleiades, PeriodO, Syriaca.org, VIAF, Getty, Trismegistos, DBPedia
General purpose platforms, CMS, VREs, tools and services	Omeka, MediaWiki, Heurist, TextGrid, Voyant, Mirador, CollateX, JUXTA, Neatline
Domain-specific platforms, CMS, VREs, tools and services	Perseids, Recogito, Symogih, PECE
Repositories and service providers	CLARIN, DARIAH, EUDAT, MLA Commons/CORE, HumaNum, Hathi Trust Research Center, California Digital Library
Aggregators and portals	Europeana, Digital Public Library of America, HuNi, EHRI
APIs and standards	IIIF, OA, TEI, OAUTH, Shibboleth/SAML, CTS

Table 1: Examples of infrastructure in digital humanities and digital classics.

Above and beyond this, ensuring that adequate editorial and/or peer review has taken place before data is shared is often an important criteria for data sharing in the humanities.

Background

Perseids evolved to fill a critical need of the digital classics community of scholars and students (Bodard and Romanello 2016): infrastructure that supports textual transcription, annotation, and analysis at a large scale, with review, in both scholarly and pedagogical contexts. Such infrastructure would give us the ability to work with text-centric publications containing a variety of different data types, and would include:

- stable, persistent identifiers for all publications;
- a versioned, collaborative editing environment;
- the ability to extend the environment with data type-specific behaviors and tools;
- customized review workflows.

Perseids is, in part, a successor to a prior ambitious, but ultimately unsuccessful, infrastructure effort in the humanities, Project Bamboo (Dombrowski 2014). One of the aims of Project Bamboo was to develop a Service Oriented Architecture (SOA) that could serve a wide variety of use cases and requirements for textual analysis and humanities research. This accorded with the goal of the PDL: to begin to decouple the many services making up the Perseus 4 application, so that they could be recombined and reused to build new applications (Almas 2015). The PDL's contribution to Bamboo included development (and implementation) of service APIs for morphological analysis and syntactic annotation. These services, intended to be shared on the Bamboo Services Platform, reused code from two main sources: the PDL's web application and the Alpheios Project's reading environment, and were designed to be easily extended to serve additional languages and use cases. They provided essential functionality for textual analysis and annotation.

At the same time, we also began separately investigating development of a scalable solution for engaging undergraduate students in the production of original transcriptions and translations of Medieval Latin Manuscripts and Greek Epigraphy. This work was inspired by, and involved reuse of architecture and tools from, two major projects in digital classics, the Homer Multitext and Papyri.info (Almas and Beaulieu 2013).

One thing that prevented Bamboo from succeeding was the assumption that scholars would be willing to give up their domain-specific tools and services for a more general infrastructure to which everyone would contribute (Dombrowski 2015). Humanities use cases at the time appeared too diverse for that, and technologies were moving very fast. It is unclear whether or not Bamboo could have succeeded but the project ended before we could develop a critical component needed for our own use cases, a platform for management of the data and scholarly workflow which would allow for full peer and professorial review.

Perseids took up in part where Bamboo left off, but with a more modest goal of providing infrastructure for our own specific set of use cases. We reused the services we built for Bamboo in Perseids, and also reused

an existing piece of infrastructure from another project, the Son of SUDA Online (SoSOL), to fill the role of managing the data and review workflows.

Drawing on the experiences of Bamboo, we decided that Perseids would support a looser coupling of existing tools and services. One goal of infrastructure is to connect what already works, adding value and capacity without reinventing solutions. Our development approach for Perseids was thus based on three principles:

1. data interoperability;
2. flexibility and agility;
3. tool interoperability.

We wanted not only to support our scholarly workflows, but also to be sure that the outputs would be fully sharable and preservable.

Perseids currently serves an active user base, averaging between one and two thousand sessions by at least five hundred unique users per month during the academic year, the majority of which come from six active DH communities: Tufts, the University of Nebraska at Lincoln, the College of Letters and Science of the Sao Paulo State University, the University of Leipzig, the University of Lyon, and the University of Zagreb. Several external projects also connect to Perseids's tools and review workflow via its API.

Functionality

Use Cases

Perseids offers functionality for creation, curation and review of texts, translations and annotations. It enables its users to:

1. Create and edit a new text transcription.
2. Edit an existing text transcription.
3. Create and edit a new text translation.
4. Edit an existing text translation.
5. Create and edit a new commentary annotation.
6. Create and edit a new treebank[1] annotation.
7. Create and edit a new text alignment[2] Annotation.
8. Ingest and edit simple annotation data from external sources.
9. Create and edit simple annotations on texts.

The process of creating a publication on Perseids involves workflows fulfilling one or more of these use cases (**Figure 1**).

Workflows

A workflow, in this context, is a series of actions carried out by a user to achieve some goal. In a typical workflow on Perseids the user creates a publication containing one or more of the supported data types. She uses an editing tool appropriate to the data type to edit and curate her work and then submits it to a review board for acceptance. For example, she may choose to create and edit a Treebank annotation using the Arethusa editing tool (**Figure 2**).

If the work is being done in the context of a pedagogical assignment, the review board is likely to be made up of the professor and teaching assistants for the class. If the work is being done in the context of a specific project or community, the review board will be composed of peers or expert members of an editorial team (**Figures 3** and **4**).

The ability to support peer-review functionality is a distinguishing feature of the Perseids infrastructure, and an important driving factor behind the architectural decision to built it upon the SoSOL platform. As we discuss further below, a common driver for external projects to integrate with Perseids is to take advantage of the flexible review workflow features it offers.

[1] Annotation of morphology, syntax and sentence structure.

[2] N-to-N word-level alignment across two texts.

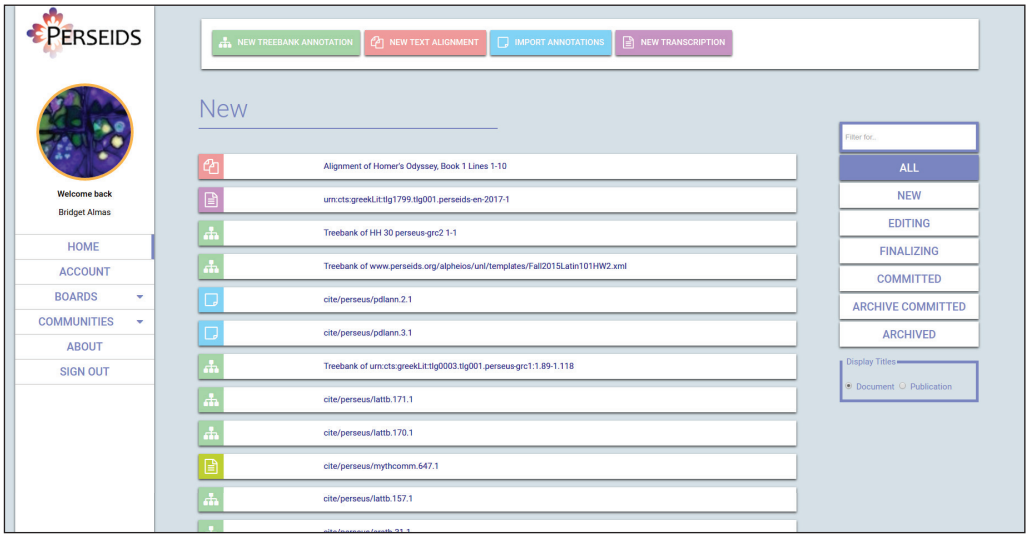

Figure 1: The Perseids home screen, showing a variety of data types and actions.

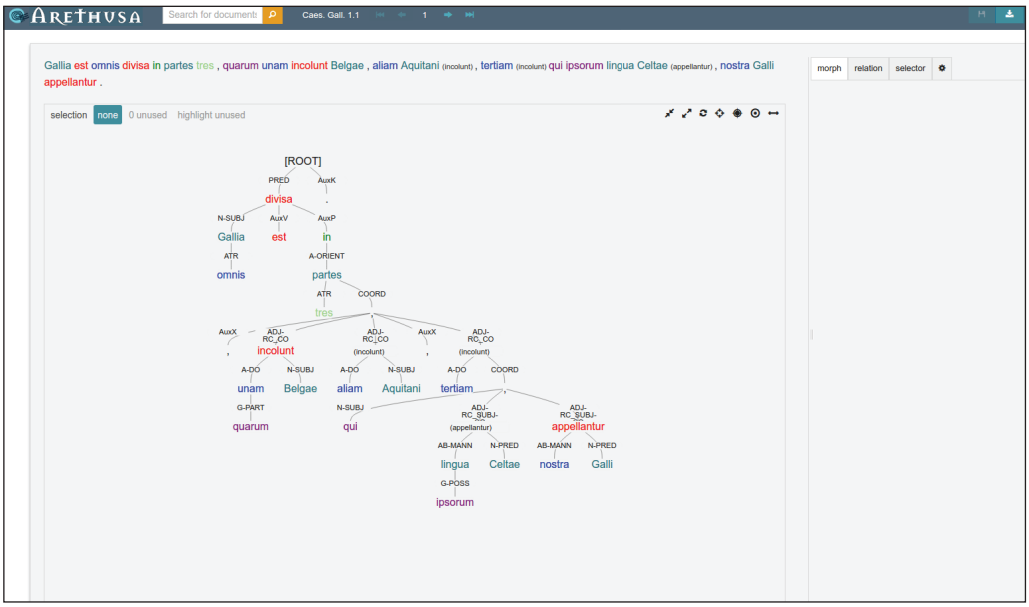

Figure 2: Annotating a Treebank in Arethusa.

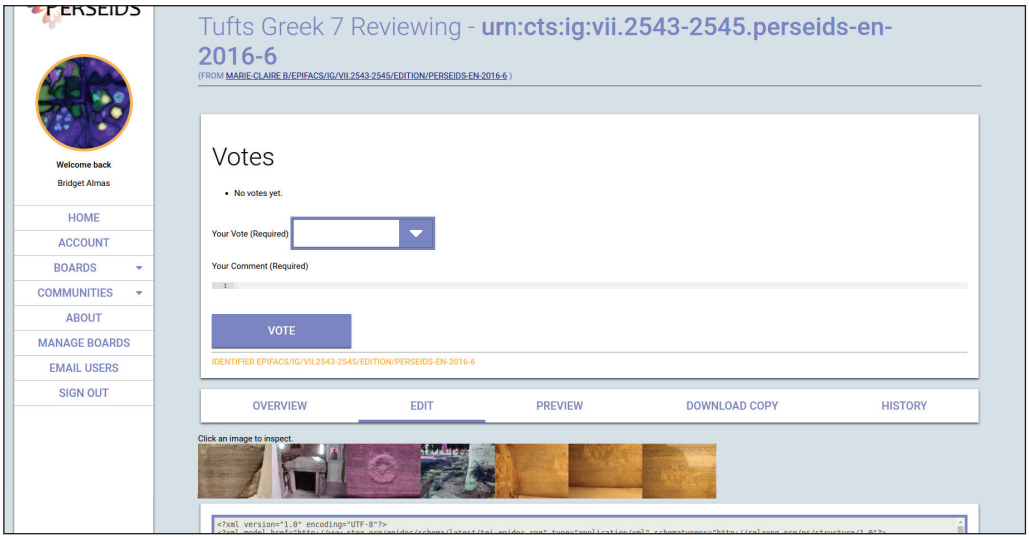

Figure 3: Perseids user interface – voting on a publication.

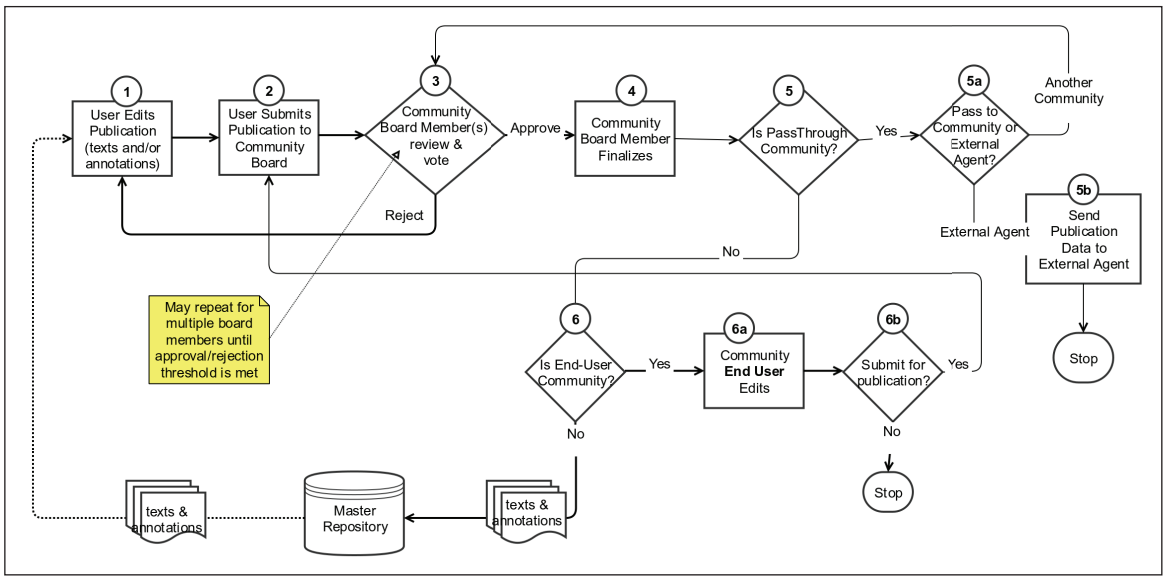

Figure 4: Perseids review workflow.

Architecture

The Perseids architecture (**Figures 5–7**) supports these workflows through a complex sequence of interactions between its core components, hosted tools and services, 3rd party applications and platforms and external identity providers and content repositories.

SoSOL is the core of the Perseids platform. It is a Ruby on Rails application, built on top of a Git repository, that provides an open-access, version-controlled, multi-author web-based editing environment that supports working with collections of related data objects as publications. SoSOL was developed for the Papyri.info site by the Integrating Digital Papyrology project, a multi-institution project aimed at supporting interoperability between five different digital papyrological resources (Baumann 2013) and is now maintained jointly by the Duke Collaboratory for Classics Computing and the Perseids project.

A Git repository provides versioning support for all documents, annotations and other related objects managed on the platform. SoSOL also provides additional functionality on top of Git's, including document validation, templates for documentation creation, review boards, and communities. It uses a relational database (MySQL) to store information about document status and to track the activity of users, boards, and communities. SoSOL uses the OpenID and Shibboleth/SAML protocols to delegate responsibility for user authentication to social or institutional identity providers. Social identity providers (IdP) are supported through a third-party gateway, currently Janrain Engage.

The Perseids deployment of SoSOL incorporates the Canonical Text Services (CTS) protocol. The CTS specification defines an API protocol and a URN syntax for identifying and retrieving text passages via machine-actionable, canonical identifiers (Smith and Blackwell 2012). To support CTS, as well as provide features such as tokenization of texts, the Perseids deployment of SoSOL delegates some functionality to external databases and services.

The SoSOL application itself provides lightweight user interfaces for creating and editing documents and annotations, but in order to support an open-ended set of different editing and annotation activities, we rely on integrations with external web-based tools for editing and annotating. These integrations are enabled by API interactions between the tools and the SoSOL application.

The Perseids Client Applications component acts as a broker between the end-user, the SoSOL platform, external repositories and services, and the web-based editing and annotation tools.[3] Built on the Python Flask framework, this component implements a client-side workflow for the creation of new annotations of text passages identified by CTS URN. It uses the CTS abstraction libraries from CapiTainS infrastructure for CTS URN resolution and processing, as does the Nemo browsing interface, which offers a discovery interface for identifying texts to annotate and an anchoring point for front-end annotation tools and visualizations.

[3] The Perseids Client Applications were co-developed by Perseids and The Humboldt Chair for Digital Humanities at the University of Leipzig.

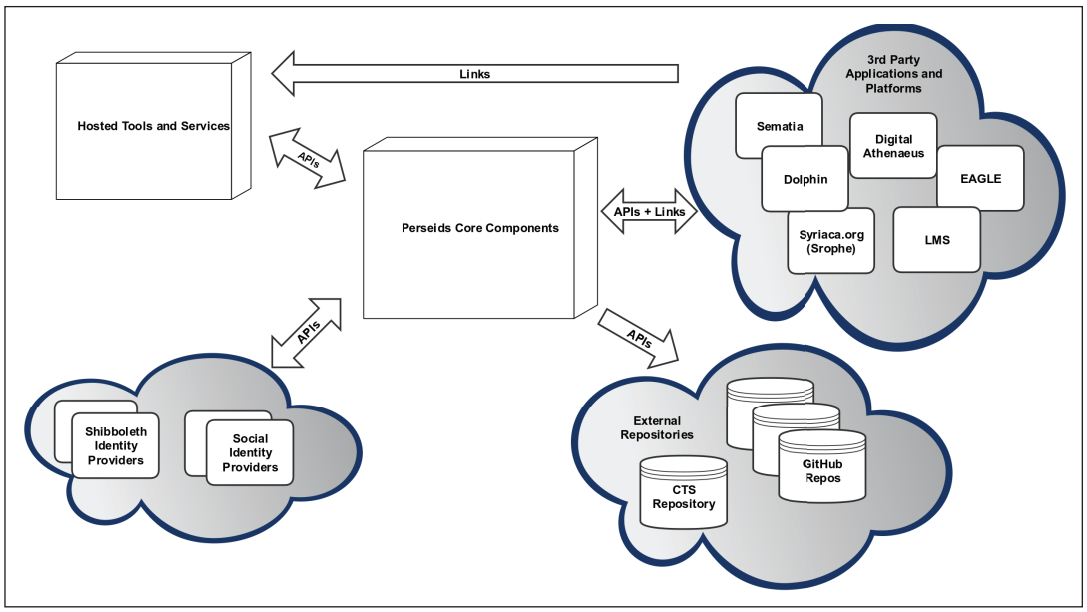

Figure 5: Perseids infrastructure and ecosystem.

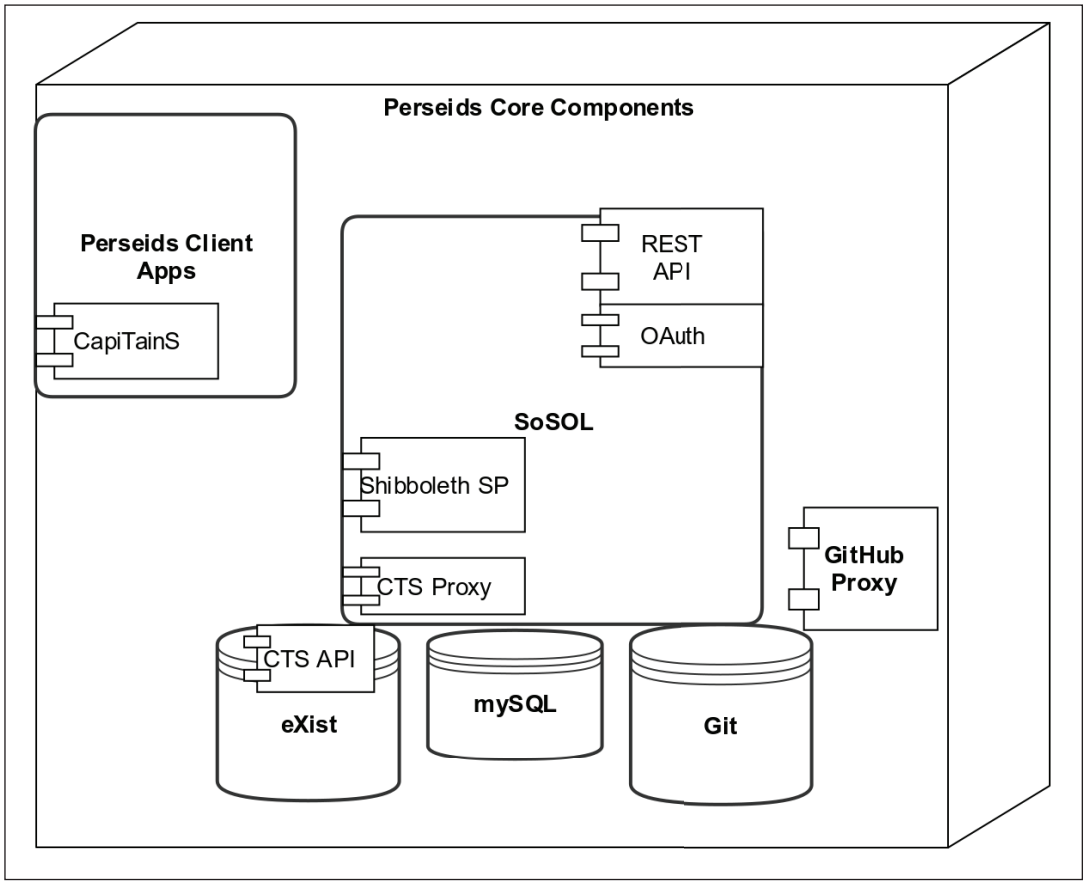

Figure 6: Perseids core components.

A recent addition to the platform is a Flask based GitHub Proxy Service which enables us to send data directly to external GitHub repositories after it has been through the review workflow.[4] (See the 'Tool Interoperability' section below for further details on these scenarios.)

The role that each component of the architecture and ecosystem plays in supporting the workflow is described in the 'Tools Interoperability' section below.

[4] Development of this component was supported by an NEH-funded collaboration with the Syriaca.org project.

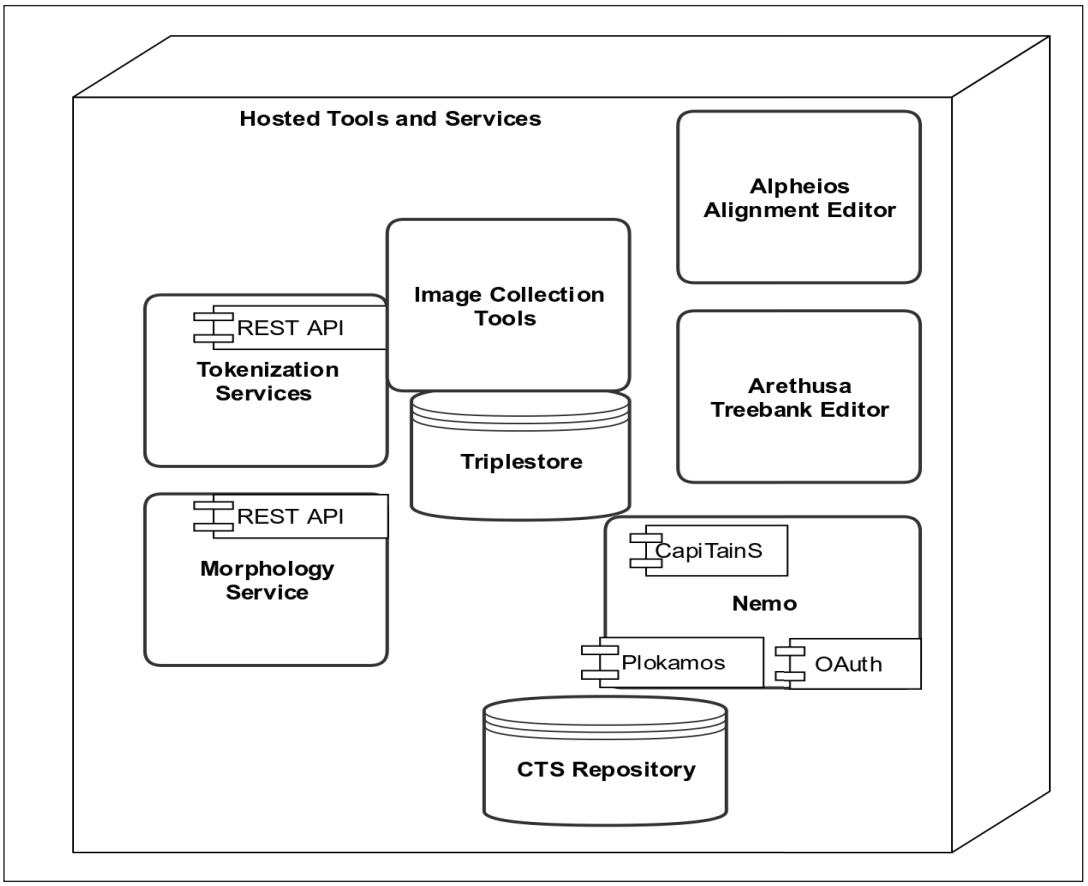

Figure 7: Perseids hosted tools and services.

Information Model

Data publications produced on Perseids are collections of related data objects of different types. The SoSOL information model was designed for this type of publication. The "Publication" is the container for a collection of data objects belonging to a parent abstract class of "Identifier." Different type object types are implemented as derivations of the "Identifier" class, which add type-specific behaviors and properties, such as schema validation rules. **Figure 8** shows how this design applies in Perseids.

However, Perseids publications can also be thought of as research objects (Bechhofer, et. al. 2013), where the object of the research is a passage or passages of canonically-identifiable text. **Figure 9** shows our original vision for a CTS-focused publication on Perseids[5] (**Figure 9**).

Tool interoperability

Decoupling data creation tools from the sources and destinations of the data was a key part of our design approach. APIs and standards are critical components of infrastructure, and integration and sharing require that data be retrievable from and persistable to any source (Hilton 2014).

Perseids offers an API for Create, Read, Update, and Delete update operations for all data types supported by the platform. API clients can authenticate using the OAuth 2.0 protocol (Hardt 2012) or co-hosted tools have the option of using a shared session cookie. These approaches enable integration with specific tools and services, such as the Arethusa Annotation Framework and the Alpheios Alignment Editor, as well as external projects such as Sematia (Vierros and Henriksson 2016) and the Syriaca.org Gazetteer (**Figure 10**).

Perseids also uses external APIs to pull data from other infrastructures. We use the Canonical Text Services URN protocol and API (Smith and Blackwell 2012) to identify and retrieve textual transcription, translation, and annotation targets (**Figures 11** and **12**).

[5] The vision in **Figure 3** has largely been implemented, with the exception of CITE collections server component. We now expect this function to be filled by an implementation of a multidisciplinary Collections API we are working on as part of the Research Data Alliance's Research Data Collections Working Group.

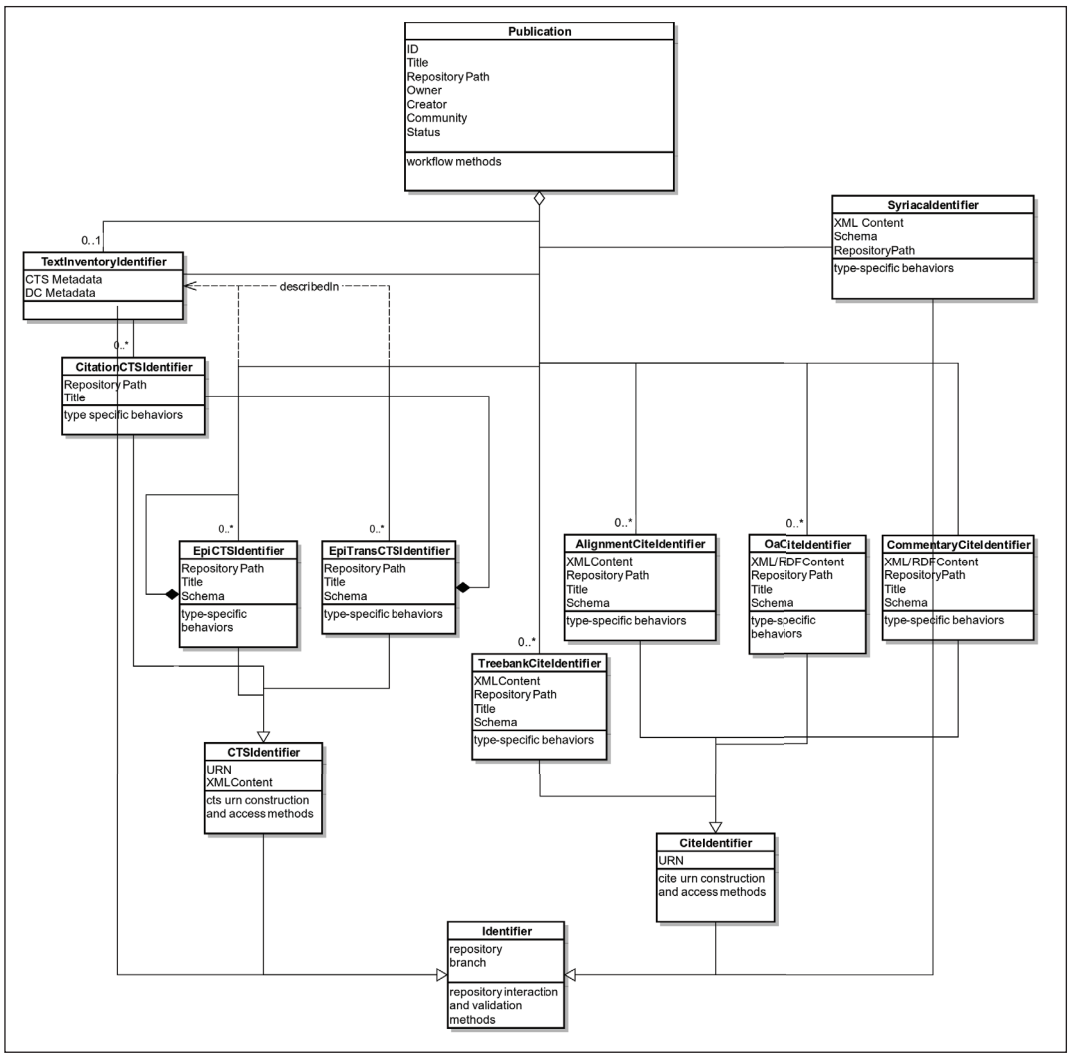

Figure 8: Perseids information model.

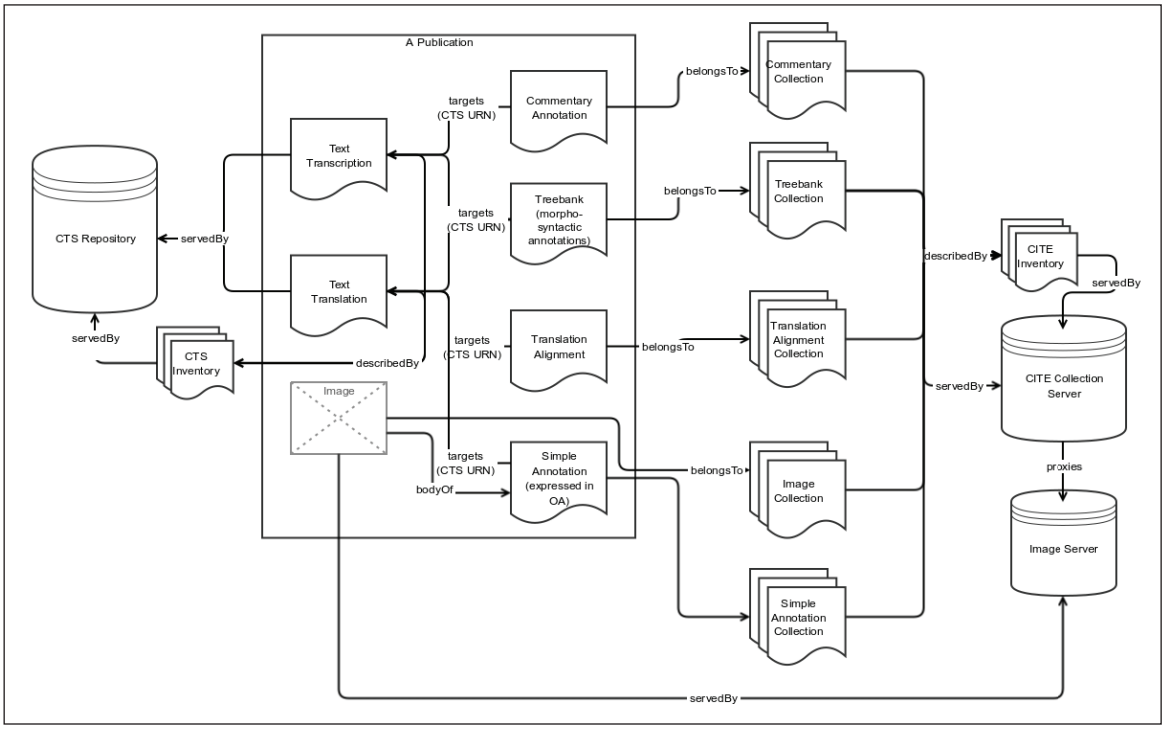

Figure 9: Perseids publication as a CTS focused research object.

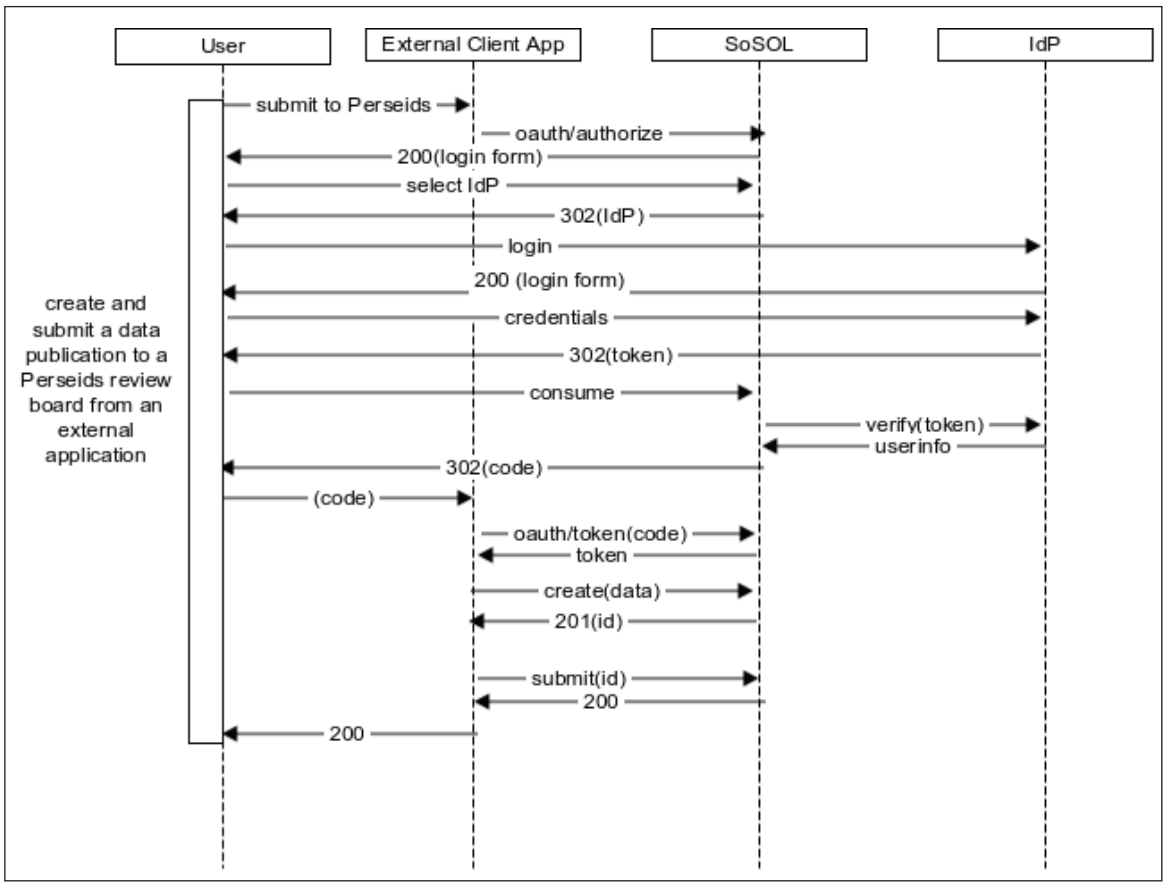

Figure 10: Creating and submitting a publication from an external application using OAuth2.

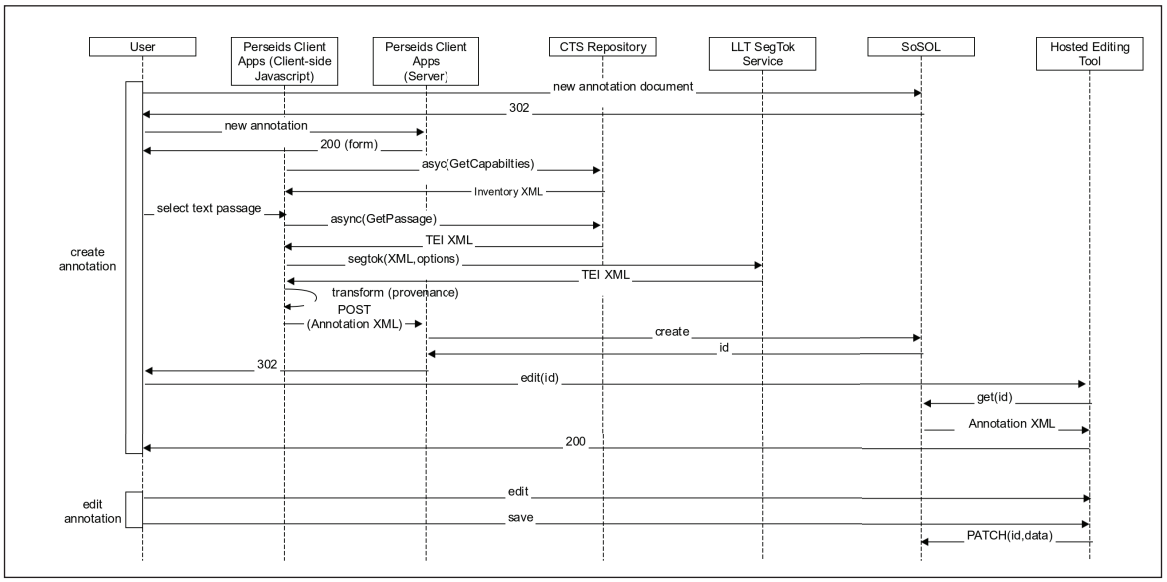

Figure 11: Sequence of API interactions for creating and editing a CTS-focused annotation template using the Perseid Client Apps and a locally hosted editing tool.

We also offer a lightweight URL-based API which lets individual scholars and smaller projects, particularly those without the time or skills to develop client software, pull their own data in or integrate Perseids with their application. Professors such as Robert Gorman at University of Nebraska Lincoln (Gorman and Gorman, forthcoming) are using this feature to produce templates for new annotations that they publish on their university Learning Management Systems (LMS). They then include links to Perseids in their syllabi that instruct Perseids to pull the templates from the LMS to create a new annotation publication (**Figure 13**).

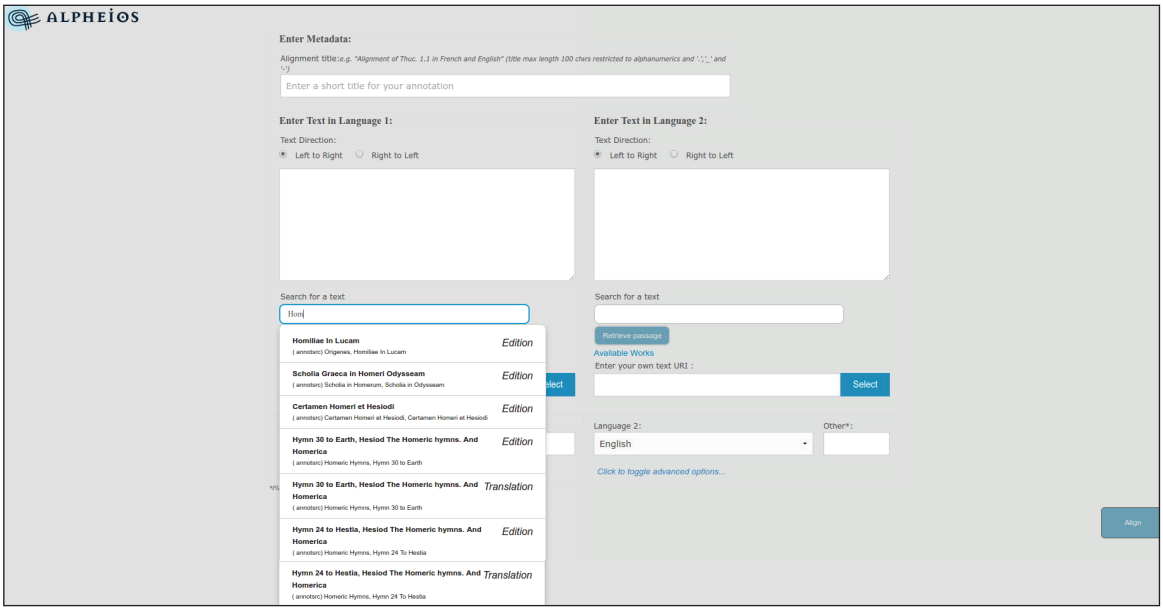

Figure 12: Using the Perseids Client Apps to create a new translation alignment annotation in Perseids for editing via the Alpheios Alignment Editor. Texts available for use are populated via a call to the CTS API.

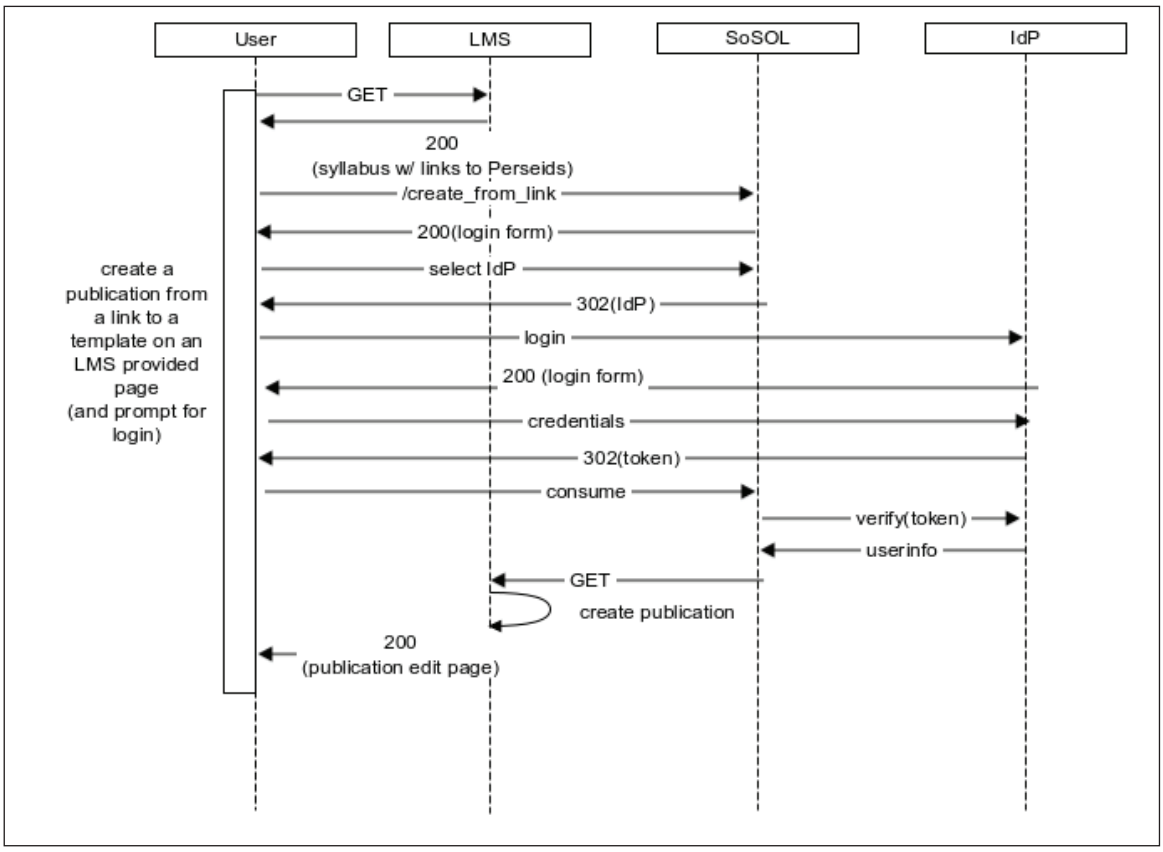

Figure 13: Sequence of actions for creating a publication from an LMS-hosted syllabus and annotation template.

Other applications such as Digital Athenaeus use Perseids's URL API to offer links to Perseids with specific content already identified for transcribing, translating, or annotating (**Figures 14** and **15**).

We also implemented a workflow for Marie-Claire Beaulieu's Journey of the Hero course which allows students to use the Hypothes.is annotation tool to annotate named entities and social networks of mythological characters from Smith's Dictionary of Greek Names. This workflow uses the Hypothes.is API to pull the annotations into Perseids for review and publication (**Figure 16**).

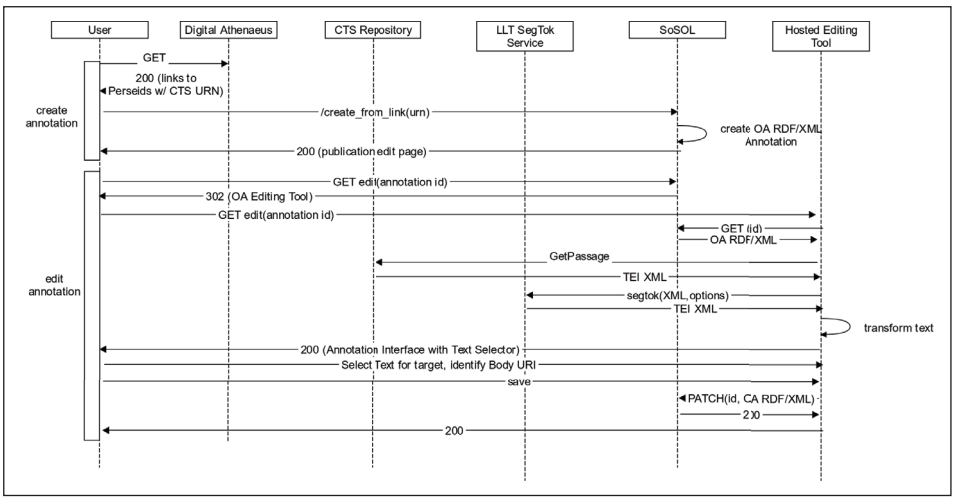

Figure 14: Sequence of actions for creating a CTS targeted text annotation publication from a link from Digital Athenaeus.

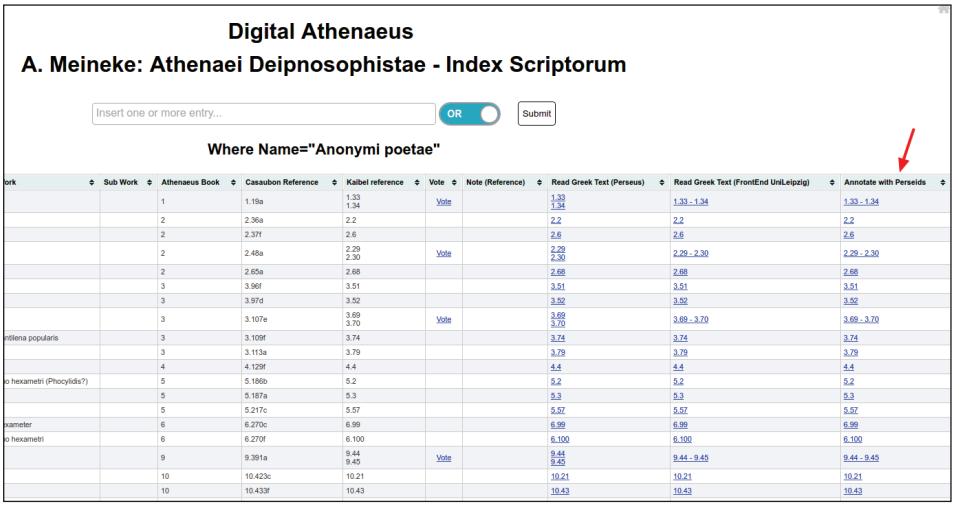

Figure 15: Screenshot of the Digital Athenaeus interface (at http://www.digitalathenaeus.org) showing the links to Annotate in Perseids.

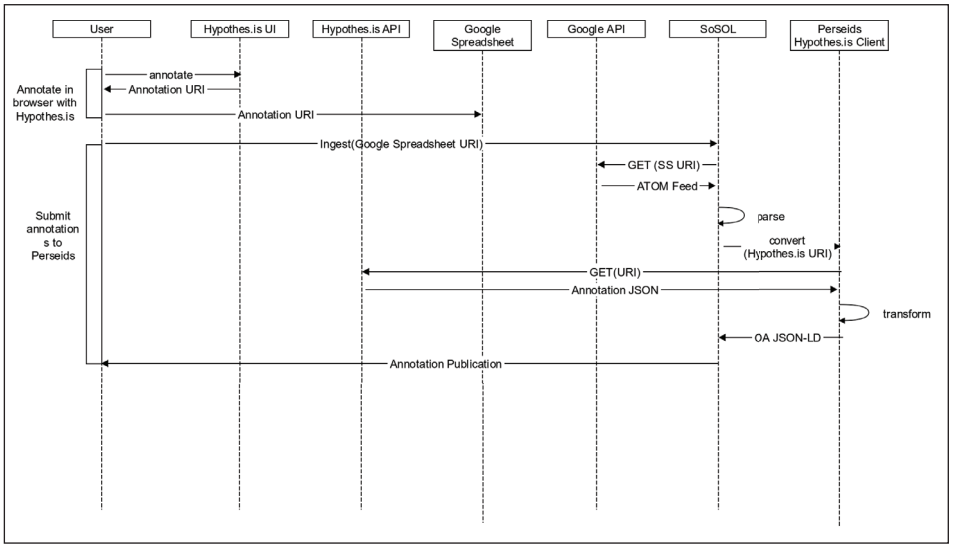

Figure 16: Perseids Hypothes.is workflow.

The Perseids/EAGLE integration uses a combination of both of these pull strategies: links from EAGLE to Perseids identify a resource on the EAGLE site, and trigger a callback to the EAGLE MediaWiki API to pull metadata and data from that resource into new translation publications on Perseids (**Figures 17** and **18**).

We also use external APIs to push data to external repositories. For the EAGLE project integration, Perseids uses the MediaWiki API to publish data to the EAGLE repository once it has passed through a review workflow. Through a new NEH-funded collaboration with the Syriaca.org project, we have developed a service which allows us to push data to external GitHub repositories at the end of the review workflow (See **Figure 4**, Step 5b). Eventually we'd like to be able to support pushing data to any external API endpoint.

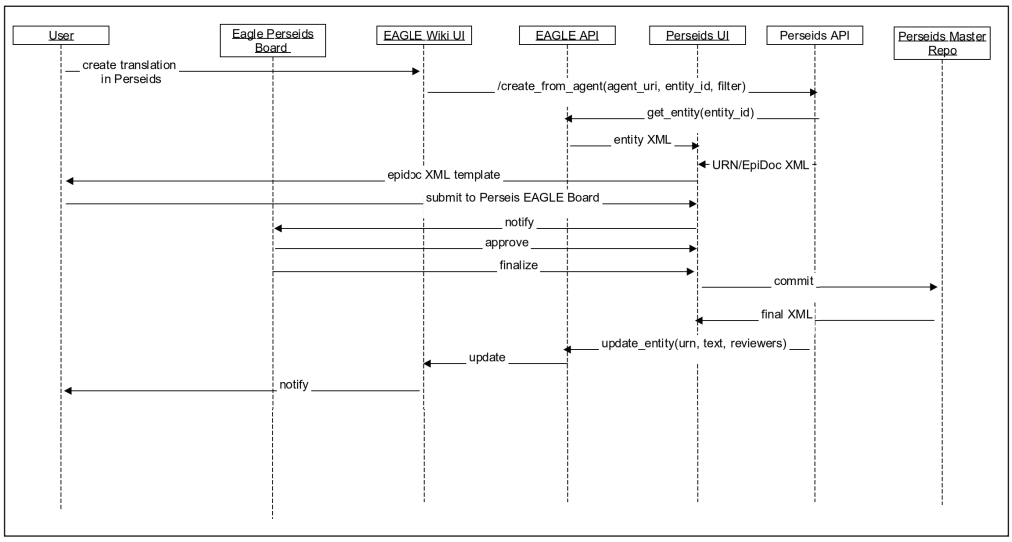

Figure 17: Perseids/EAGLE workflow.

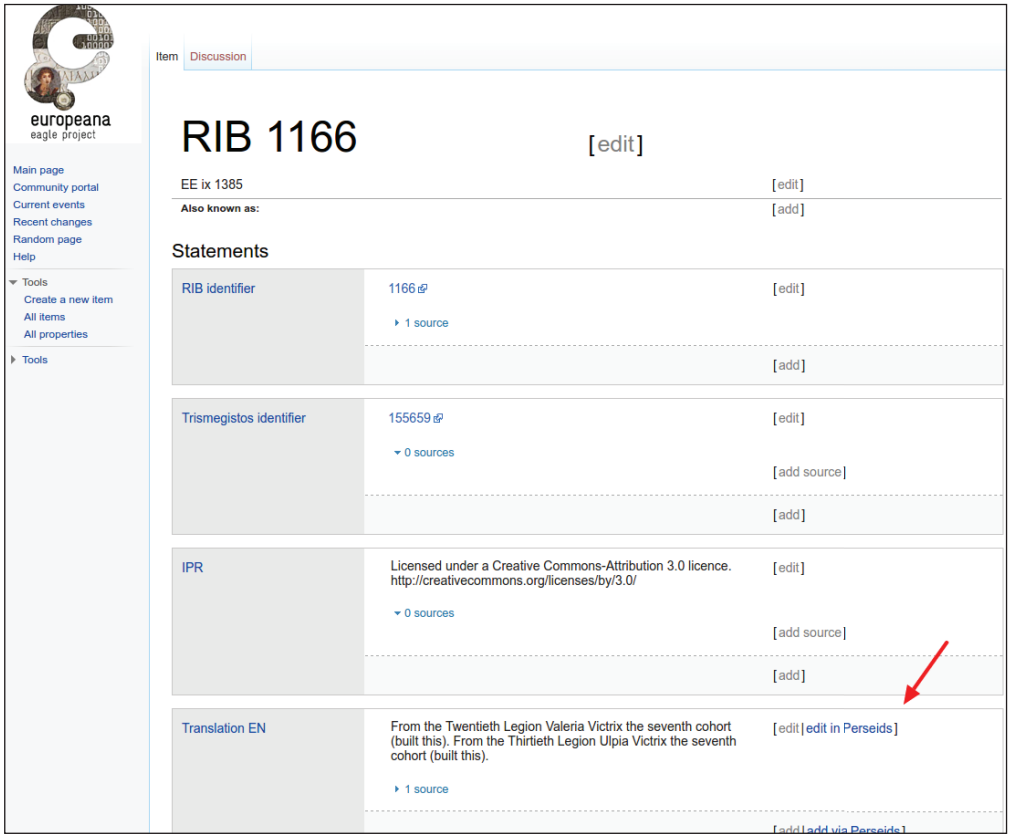

Figure 18: Screenshot of the EAGLE Portal (http://www.eagle-network.eu/wiki) showing a link to edit a translation in Perseids.

Designing for Flexibility and Agility

From the outset, we have taken an agile approach to development of Perseids. While we do not use official sprints and strictly scheduled iterations, we approach planning in short increments, guided by a long-term vision and goals. In addition, we aim to deploy features to users as quickly as possible, so that we can get feedback from them. We do this not only for internal-facing features, but also to prototype new integrations with external services and projects. This flexibility allows us to try many things, keeping those that work and prove to be useful and deprecating those that do not.

To support this approach, we could not commit to a specific set of hardware requirements in advance, as we needed the flexibility to extend and reduce resources used as development proceeded. We therefore chose to budget for cloud-based resources on the Amazon Web Services (AWS) platform rather than using university IT resources. Full ownership and control over our infrastructure allowed us to experiment with features and integrations that otherwise would not have been possible; however, it did have some drawbacks and unexpected costs. These are described in the 'Sustainability' section below.

Standards for Data

Data Interoperability

A strategic principle in our development is to take steps to ensure data interoperability through the use of stable identifiers and standard formats.

We use CTS URNs to identify both texts and annotation targets. These URNs can be considered stable identifiers, but do not quite qualify as persistent identifiers as they are not universally resolvable or guaranteed to be available. Other identifier systems, such as Handles (Sun et. al. 2003), are designed for persistence, and one approach we might take in the future to address this would be to map CTS URNs to the Handles (Almas and Schroeder, 2016), but in the absence of this piece of infrastructure, the CTS URNs do provide stable, machine actionable identifiers that are technology independent.

We also use other types of stable identifiers within our annotations and texts, including the URIs published by the Pleiades Gazetteer. We are working towards ensuring that any data published by the platform has a persistent identifier as well. We are therefore participating in the Research Data Alliance's Research Data Collections working group to develop a multidisciplinary, collections-based approach to data management that supports persistent identifiers for the collections themselves, and for the items within a collection.

We also strive to use standard data formats and ontologies for our data and to validate all objects against these. The primary data format standards supported on the platform include the TEI Epidoc Schema for textual transcriptions and translations, the Open Annotation protocol for annotations, the ALDT/ALGT schemas for treebank data, the Alpheios Alignment Scheme for translation alignments, and the SNAP ontology for social network annotations.

Provenance and Preservation

Incorporating provenance information in our publications is an important enabling factor for data sharing. We have taken steps in this direction, for example by supporting Shibboleth/SAML protocol for authentication on Perseids in order to to be able to ensure a chain of authority for university repository systems. We have also included provenance information for tokenization services and tools in our annotation documents, and have explored models for more comprehensive approaches (Almas, Berti, et. al. 2013). However, capturing and recording provenance information reliably across a diverse ecosystem of tools and services is difficult, and we need general-purpose solutions that we can reuse. As articulated by Padilla (2016): "A researcher should be able to understand why certain data were included and excluded, why certain transformations were made, who made those transformations, and at the same time a researcher should have access to the code and tools that were used to effect those transformations. Where gaps in the data are native to the vagaries of data production and capture, as is the case with web archives, these nuances must be effectively communicated." We recognize that we fall short of meeting these goals currently and aim to do a more complete job of this in the future.

It is also very important to us that the research data produced with Perseids be preserved. However, our data models and approach to publications are constantly evolving, making coordination with the university library to preserve this data challenging, as they don't necessarily fit the data models the library is already able to support. As a publicly available and open infrastructure, we also have many users from many institutions across the world, and it is not clear what responsibility Tufts, the university hosting the infrastructure, should have for data created by external users. We mitigate this with Perseids by providing links

that users can use to access and download their data, and encouraging them to take responsibility for publishing and preserving it on their own. We continue to explore general models such as the Research Object (Belhajjame, et. al. 2015), or BagIt, which will enable users to export data in a format that is ready to store in a repository. Another question is that of software preservation (Rios 2016). As the Perseids software is under active development, it is difficult to keep the code for digital publications up to date with all the underlying services providing the data (Rios and Almas 2016). We need to plan better for this preservation, including taking into account the need to represent interdependencies between visualizations and the underlying services and software (Lagos and Vion Dury 2016).

Sustainability

Human and Governance Factors

We have learned much about infrastructure building throughout the course of this project. The technical hurdles to interoperability and sharing are usually much less difficult to overcome than those of social issues, funding, and governance. Even where there was a clear interest in interoperability and it was technically possible, we failed sometimes to implement or sustain an integration because doing so wasn't in the funded mandate of the partner project. This was the case for us with the Recogito application from the Pelagios Project. But even where explicit funding support doesn't exist, interoperability can still succeed if one project can fill a key gap in another, and if there are people willing to champion the effort to ensure its success. One example was our integration with the EAGLE project, where Perseids provides a review workflow for EAGLE, and which was implemented without being a funded deliverable for either project, but it remains to be seen if we can sustain it indefinitely. This is an area where more formal governance structures, such as those offered by larger research infrastructures such as CLARIN and DARIAH (Lossau 2012) could be useful. The key challenge for the community is to encourage and support ad-hoc collaborations to get initial solutions working, and then move from there to more formal agreements to ensure sustainability.

Hardware and Software Factors

Laura Mandell talks about the various models being considered for where and how to position DH, and points out that the question of how to support diverse infrastructure needs is still unsolved (Dinsman 2016). A second lesson we have learned from our experience on Perseids is that for development of interoperable infrastructure to succeed and be sustainable, we need better collaborative models for working with our university Information Technology departments and libraries. We knew we needed the flexibility to change our hardware requirements as we developed, and to deploy new code and services quickly to support rapid prototyping. This allows us to develop and try out new solutions more rapidly than we would have been able to if we had to go through university policies and procedures, but it also involved a lot of extra system administration work we had not anticipated, leaving us with a somewhat over-complicated infrastructure at the end of the first phase of the project. Accordingly, in the second phase we built in funding for a devops consultant, who helped us move to a fully configuration-managed system, so that the Perseids platform can be deployed easily by others and sustained for the long term. This is a critical characteristic for software-related infrastructure - in order for it to be reproducible by others, building and deploying it must be automated. In hindsight, having such consultancy from the outset would have been beneficial; collaboration between developers and the IT staff responsible for deploying and sustaining software is a more viable model than throwing code 'over the wall' at the end of a project (Arundel 2016). As cloud computing becomes increasingly cost-efficient, and new models of deployment, such as container-based solutions, are introduced, there is a need for models in which university IT departments can partner with projects to provide expertise and facilities (for example, private cloud or container infrastructure, or extending university infrastructure to the public cloud).

Conclusion

With Perseids, we have explored an agile approach to infrastructure development, emphasizing reuse of both software and data. This has been successful on many levels. Reuse of existing infrastructure components leads to collaborations which increase the chances of sustainability, such as the joint maintenance of the SoSOL application. Agile approaches to prototyping cross-project integration also benefit all parties involved. However, transitioning to more formal governance models and increased engagement with host institutions will be essential to longer term success.

Acknowledgements
The author wishes to thank her colleagues, John Arundel, Frederik Baumgardt, Marie-Claire Beaulieu and Thibault Clérice for contributing their energy and ideas in reviewing this paper.

Competing Interests
The author has no competing interests to declare.

Author Information
Bridget Almas is the software architect and co-director of the Perseids Project at Tufts University. Bridget has worked in software development since 1994, in roles which have covered the full spectrum of the software development life cycle, focusing since 2007 in the fields of digital humanities and pedagogy. Bridget served as an elected member of the Technical Advisory Board of the Research Data Alliance (RDA), from 2013–2015, and currently is co-chair of the Research Data Collections Working Group, the Data Fabric Interest Group and acts as liaison between the Alliance of Digital Humanities Organizations (ADHO) and RDA.

References
Almas, B 2015 The Road to Perseus 5 – why we need infrastructure for the digital humanities. Blog post on the Perseus Updates Blog (18, May 2015). Available at: http://sites.tufts.edu/perseusupdates/2015/05/18/the-road-to-perseus-5-why-we-need-infrastructure-for-the-digital-humanities/.

Almas, B and **Beaulieu, M-C** 2013 Developing a New Integrated Editing Platform for Source Documents in Classics. *Literary and Linguistic Computing*, 28: 493–503. DOI: https://doi.org/10.1093/llc/fqt046

Almas, B, Berti, M, Choudhury, S, Dubin, D, Senseney, M and **Wickett, K** 2013 Representing Humanities Research Data Using Complementary Provenance Models. In Building Global Partnerships - RDA Second Plenary Meeting, Washington, D.C.: RDA. Available at: https://www.rd-alliance.org/filedepot_download/694/158.

Almas, B and **Schroeder, C T** 2016 Applying the Canonical Text Services Model to the Coptic SCRIPTORIUM. *Data Science Journal*, 15: 13. DOI: http://doi.org/10.5334/dsj-2016-013

Arundel, J 2016 Build bridges not walls: devops is about empathy and collaboration. Available at: http://bitfieldconsulting.com/bridges-not-walls.

Baumann, R 2013 The Son of Suda Online. In: Dunn, S and Mahoney, S (Eds.) *The Digital Classicist 2013*. Offprint from BICS Supplement-122. London: The Institute of Classical Studies University of London, pp. 91–106.

Bechhofer, S, Ainsworth, J, Bhagat, J, Buchan, I, Couch, P, Cruickshank, D, Delderfield, M, Dunlop, I, Gamble, M, Goble, C, Michaelides, D, Missier, P, Owen, S, Newman, D, De Roure, D and **Sufi, S** 2013 Why Linked Data is Not Enough for Scientists. *Future Generation Computer Systems*, 29(2): 599–611. DOI: https://doi.org/10.1016/j.future.2011.08.004

Belhajjame, K, Zhao, J, Garijo, D, Gamble, M, Hettne,K, Palma, R, Mina,E, Corcho,O, Gómez-Pérez,J M, Bechhofer,S, Klyne,G and **Goble, C** 2015 (May) Using a suite of ontologies for preserving workflow-centric research objects. *Journal of Web Semantics*, 32: 16–42. DOI: https://doi.org/10.1016/j.websem.2015.01.003

Bodard, G and **Romanello, M** (eds.) 2016 *Digital Classics Outside the Echo-Chamber*. London. Ubiquity Press.

Dinsman, M 2016 (April 24) The Digital in the Humanities: An Interview with Laura Mandell - Los Angeles Review of Books. Available at: https://lareviewofbooks.org/article/digital-humanities-interview-laura-mandell/.

Dombrowski, Q 2014 What Ever Happened to Project Bamboo? *Literary and Linguistic Computing*, 29(3): 326–339. DOI: https://doi.org/10.1093/llc/fqu026

Gorman, R and **Gorman, V** forthcoming Approaching questions of text reuse in Ancient Greek using computational syntactic stylometry. *Open Linguistics* Topical Issue on Treebanking and Ancient Languages.

Hardt, D (ed.) 2012 The OAuth 2.0 Authorization Framework, RFC 6749. Available at: http://www.rfc-editor.org/info/rfc6749. DOI: https://doi.org/10.17487/RFC6749

Hilton, J L 2014 Enter Unizin. *EDUCAUSE Review*, 49(5).

Lagos, N and **Vion-Dury, J-Y** 2016 (September 13–16) Digital Preservation Based on Contextualized Dependencies. Doc Eng. Available at: http://www.xrce.xerox.com/content/download/93294/1307736/file/2016-031.pdf.

Loscio, B F, Burle, C and **Calegari, N** 2016 (30 August) W3C. 2016 Data on the Web Best Practices. W3C Candidate Recommendation. Available at: https://www.w3.org/TR/2016/CR-dwbp-20160830/.

Lossau, N 2012 An Overview of Research Infrastructures in Europe - and Recommendations to LIBER. *LIBER Quarterly,* 21(3–4): 313–329. DOI: https://doi.org/10.18352/lq.8028

Padilla, T 2016 Humanities Data in the Library: Integrity, Form, Access. *D-Lib Magazine*, 22(3/4). DOI: https://doi.org/10.1045/march2016-padilla

Parsons, M 2015 (22 September) e-Infrastructures & RDA for data intensive science. Available at: https://rd-alliance.org/sites/default/files/attachment/Infrastructures,%20relationship,%20trust%20and%20RDA_MarkParsons.pdf.

Rios, F 2016 The Pathways of Research Software Preservation: An Educational and Planning Resource for Service Development. *D-Lib Magazine*, 22(7/8). DOI: https://doi.org/10.1045/july2016-rios

Rios, F and **Almas, B** 2016 Preserving Digital Scholarship in Perseids: An Exploration. Blog Post. DOI: https://doi.org/10.5281/zenodo.159569

Smith, N and **Blackwell, C W** 2012 Four URLs, limitless apps: Separation of concerns in the Homer Multitext architecture. In *Donum natalicium digitaliter confectum Gregorio Nagy septuagenario a discipulis collegis familiaribus oblatum: A Virtual Birthday Gift Presented to Gregory Nagy on Turning Seventy by His Students, Colleagues, and Friends.* Boston: The Center of Hellenic Studies of Harvard University

Sun, S, Lannom, L and **Boesch, B** 2003 Handle System Overview, RFC 3650. Available at: http://www.rfc-editor.org/info/rfc3650. DOI: https://doi.org/10.17487/RFC3650

Vierros, M and **Henriksson, E** 2016 Preprocessing Greek Papyri for Linguistic Annotation. Hal-01279493. Preprint. Available at: https://hal.archives-ouvertes.fr/hal-01279493.

Projects, Websites, Software

Alpheios [WWW Document] n.d. Available at: http://alpheios.net/ (accessed 9.29.16).

Alpheios Alignment Editor [Software] n.d. Available at: https://github.com/alpheios-project/alignment-editor (accessed 9.29.16).

Arethusa [Software] n.d. Available at: https://github.com/alpheios-project/arethusa (accessed 9.29.16).

CapiTainS [WWW Document] n.d. Available at: http://capitains.github.io/ (accessed 9.29.16).

Digital Athenaeus - A digital edition of the Deipnosophists of Athenaeus of Naucratis [WWW Document] n.d. Available at: http://digitalathenaeus.org/ (accessed 9.29.16).

EpiDoc Guidelines 8.22 [WWW Document] n.d. Available at: http://www.stoa.org/epidoc/gl/latest/ (accessed 9.29.16).

Flask (A Python Microframework) [WWW Document] n.d. Available at: http://flask.pocoo.org/ (accessed 11.8.16).

flask-github-proxy: Github proxy to push resource to github [Software] n.d. Available at: https://github.com/PonteIneptique/flask-github-proxy (accessed 9.29.16).

Hypothes.is [WWW Document] n.d. Available at: https://hypothes.is/ (accessed 9.29.16).

Journey of the Hero [WWW Document] n.d. Available at: http://perseids.org/sites/joth/#index (accessed 9.29.16).

Morphological Analysis Service Contract Description - v1.1.1 [WWW Document] n.d. Available at: https://wikihub.berkeley.edu/display/pbamboo/Morphological+Analysis+Service+Contract+Description+-+v1.1.1 (accessed 9.29.16).

OpenID Authentication 2.0 - Final [WWW Document] n.d. Available at: http://openid.net/specs/openid-authentication-2_0.html (accessed 11.3.16).

Pleiades Gazetteer [WWW Document] n.d. Available at: https://pleiades.stoa.org/ (accessed 9.29.16).

RECOGITO [WWW Document] n.d. Available at: http://pelagios.org/recogito (accessed 9.29.16).

Research Data Collections WG [WWW Document] n.d. Available at: https://rd-alliance.org/groups/pid-collections-wg.html (accessed 9.29.16).

Sematia [WWW Document] n.d. Available at: http://sematia.hum.helsinki.fi (accessed 9.29.16).

Shibboleth [WWW Document] n.d. Available at: https://shibboleth.net/ (accessed 11.3.16).

Standards for Networking Ancient Prosopographies [WWW Document] n.d. Available at: https://snapdrgn.net/ontology (accessed 9.29.16).

Syntactic Annotation Service Contract Description - v1.1.1 [WWW Document] n.d. Available at: https://wikihub.berkeley.edu/display/pbamboo/Syntactic+Annotation+Service+Contract+Description+-+v1.1.1 (accessed 9.29.16).

Syriaca.org: The Syriac Reference Portal [WWW Document] n.d. Available at: http://syriaca.org/ (accessed 9.29.16).

The Ancient Greek and Latin Dependency Treebank by PerseusDL [WWW Document] n.d. Available at: https://perseusdl.github.io/treebank_data/ (accessed 9.29.16).

The BagIt File Packaging Format (V0.97) [WWW Document] n.d. Available at: https://tools.ietf.org/html/draft-kunze-bagit-08 (accessed 9.29.16).

How to cite this article: Almas, B 2017 Perseids: Experimenting with Infrastructure for Creating and Sharing Research Data in the Digital Humanities. *Data Science Journal,* 16: 19, pp. 1–17, DOI: https://doi.org/10.5334/dsj-2017-019

Submitted: 10 November 2016 **Accepted:** 17 March 2017 **Published:** 18 April 2017

]u[*Data Science Journal* is a peer-reviewed open access journal published by Ubiquity Press.

OPEN ACCESS