DHQ: Digital Humanities Quarterly
2019
Volume 13 Number 4

Decoupling Quality Control and Publication: The Digital Latin Library and the Traveling Imprimatur.

Samuel J. Huskey <huskey_at_ou_dot_edu>, University of Oklahoma
Jeffrey C. Witt <jcwitt_at_loyola_dot_edu>, Loyola University Maryland

Abstract

With *The Library of Digital Latin Texts* (LDLT), the Digital Latin Library (DLL) is set to publish critical editions as sets of data independent of a presentational format. These datasets are meant to be reused for a variety of purposes and in a variety of presentational formats, including the DLL's own user edition viewer and data visualization applications. It is thus vital to have a way of certifying the data as published in accordance with the DLL's policies and procedures, central to which are its provisions for peer review. Accordingly, to facilitate the use of an edition's data in different settings, and to promote a high degree of confidence in the quality of the data, we have introduced the DLL Review Registry Service. The DLL Review Registry Service offers a way of associating peer reviewed status with the text data rather than within any particularly visualization of that data. Any visualization of text data can consult this service and discover whether or not this data has been reviewed. In this way, we free the "imprimatur" from its association with any particular presentation and allow the indicators of quality to "travel" with the data and be communicated to end users in a plurality of visualizations.

Background and History of the Digital Latin Library

People have been publishing digital versions of Latin texts — in the sense of making them available to the public — for at least as long as there has been an internet. Until recently, however, digital Latin texts did not include any of the features of critical editions, for a variety of reasons. Copyright restrictions have prevented the digitization of anything but the main text of existing editions. The technical challenges of representing a critical apparatus in a digital format are considerable. Perhaps the most significant barrier has been the problem of defining what a digital critical edition is. Then there is the issue of publication itself. Scholars are reluctant to try a new form of publication if they are not certain that their peers will equate it with existing forms. 1

For all of these reasons, the publication of digital critical editions of Latin texts has been the primary objective of the DLL project from its beginning. It is why representatives of three major learned societies, the Society for Classical Studies (SCS), the Medieval Academy of America (MAA), and the Renaissance Society of America (RSA) have always been on the project's advisory board, since these three groups share a common interest in promoting the publication of digital critical editions of Latin texts. Publication has also been the main factor in the project's funding, which has so far come solely from the Scholarly Communications program of the Andrew W. Mellon Foundation.[1] In short, publication is the reason for the project's existence. 2

But what will the DLL publish? The term *digital edition* can be applied to a wide variety of projects, from simple web pages with just a text and some notes to multimedia projects that include transcriptions of multiple manuscripts, digital images of manuscripts and editions, visual representations of textual data, links to other resources, special tools for reading and analysis, and other features.[2] This wide range of possibilities makes it possible to create digital editions that highlight the special characteristics of individual texts or collections. But those same possibilities complicate the implementation of uniform criteria for evaluating projects and developing reliable, stable outlets for their publication. It also frustrates efforts to query, compare, and reuse information from multiple projects, since the data are stored in any number of formats. 3

The key word in that last sentence is *data*. Regardless of the presentational format — a printed book, a multimedia resource, a data file, a visualization — the quality of the information in a critical edition is what matters most. However aesthetically pleasing a critical edition — print or digital — might be, it is not worth much if the text and accompanying materials are based on shoddy scholarship. For that reason, we have made the decision to *decouple* the data of a digital critical edition from any of the presentational formats that it might take. In other words, the DLL is a publisher of critical editions not as books or web pages, but as sets of data. 4

This is not to say that the presentation of an edition's data does not matter, or that encoded data is not itself a type of presentation.[3] Rather, the point of the decoupling that we propose here is to recognize the importance of treating presentation as a field of scholarship and practice in its own right. In other words, the work of textual editors should be editing texts. In addition to doing the work of textual criticism, editors should not be expected to master the fields of Data Visualization and Human Computer Interaction, not to mention the technologies 5

and programming languages required to build and maintain sophisticated applications.[4] By the same argument, designers of interfaces and tools should not be expected to master the field of textual criticism. They should, however, be expected to base their work on texts that meet the highest standards of scholarship.

The DLL itself puts this decoupling into practice by providing its own applications for using LDLT texts. For example, its web-based “DLL Viewer” is an ongoing scholarly project of Hugh Cayless that presents an LDLT edition’s data in a user-friendly, feature-rich online interface based on the JavaScript library CETEIcean.[5] Chris Weaver has also built an experimental application for applying data visualization techniques to critical editions, with a view to encouraging data visualization as a form of philological scholarship[6]. The DLL also provides a service for generating PDF versions of critical editions so that they can be printed out in the format familiar to generations of readers. Since the data for the DLL's critical editions will be available on an open basis, others may devise other presentational formats, which leads to the point of this article. 6

Our aim is to provide a means of independently verifying that a text presented in an interface has undergone scholarly peer review. In this way, textual editors receive credit for their work when it is used in other projects, and interface designers and others can certify that their data come from the authorized, published version of a text. The DLL Review Registry Service fills that need by providing what we are calling a “travelling imprimatur”. 7

The DLL Review Registry Service

The DLL Review Registry Service is one piece of a larger ecosystem that is trying to change the way we think about publication and quality control. In traditional models, publication and the imprimatur of quality control have been tightly coupled. The publisher is responsible not only for printing and distributing a text, but also for providing an imprimatur that indicates that the text in question has reached a certain level of quality. A consequence of this approach is that the imprimatur of a text is bound to a particular presentation of the underlying data. This coupling means that if one wants to access or use the reviewed text, one must do so exclusively within the confines of the particular presentation offered by the publisher. This requirement severely limits our ability to reuse the underlying data for new and unanticipated purposes and it likewise de-incentivizes scholarly interest in creating reusable data of high quality. 8

One of the most exciting possibilities afforded to us by the digital medium is the ability to use and re-use a text for a plurality of purposes and within a plurality of presentations. With a single source document we can create books, websites, databases, and networks tailored to specific research questions. 9

Unfortunately, the dominant publication practices have not kept pace with these changes. While it is now technologically possible to separate the underlying data of a text from its presentational form, the practice of coupling an imprimatur to its publication practically demands that the reviewed text can only be used in a single presentational form, severely limiting the way a peer-reviewed text can be reused. 10

The DLL and its partners aim to offer a new kind of peer review that separates the task of review and quality control from the task of publication and distribution. 11

In our model, each of the partner organizations undertakes the tasks of reviewing the underlying data of a given text rather than any particular presentational form. The DLL Review Registry Service exists to help record these reviews and make them accessible for re-use throughout the web in any publication platform and any client viewer. 12

On this approach, reviews are tied to the reviewed text using a cryptographic hash. [7] A cryptographic hash allows us to identify a text not by its location on a file system, but by its digital fingerprint. Every file reviewed by a given society can be reduced to a unique number. If a single character were added or subtracted from the reviewed text, its fingerprint would change, and any end user would know immediately that the file in question is not identical to the file reviewed and approved by the society. 13

The DLL Review Registry Service is built around these unique fingerprints. Every review certificate (described below) indicates the hash of the file that has been approved. It then associates this hash with a rubric or set of criteria used by the society to make their evaluation. 14

Meanwhile, any publication application, whether print or web, can use the fingerprint of the file to discover any existing reviews. Using the DLL Review Registry Service API, a client application can compute the hash of any file it has. It can then send that hash to the DLL endpoint and discover any review of that precise file. For any request, the DLL endpoint will respond with a machine actionable set of metadata about the review for this file. 15

Included in this response is a link to a society’s *badge* or *icon* that can be used in client applications to show that the text the end-user is currently seeing has been reviewed by one or more of the academic societies working in collaboration with the DLL. Additionally, the DLL Review Registry Service will return a review ID. Clients can use this ID to create a hyperlink back to the original review. In this way, end users can click on the *badge* and be immediately brought back to the review report to verify that the text they are viewing on a client site is the same text that has been reviewed through the DLL. Finally, clients will receive a pointer to an issued certificate, indicating the nature of 16

the review and the document reviewed, that itself has a cryptographic hash and is digitally signed using the reviewing society's private key. This certificate can itself be detached from the DLL Review Registry Service and accessed anywhere on the web. No matter where this signed certificate is discovered, by using the certification verification system described below, clients can offer users a way to verify the authenticity of the certification independent of the DLL Review Registry Service or domain name.

In this regard, the DLL Review Registry Service operates as one of possibly many indexing or discovery services. This service is therefore centralizing only in the sense that it keeps track of review certificates in a single registry. It thereby provides a convenient API for clients to discover these certificates. However, because the authority of the review certificates does not come from its association with a given domain name or API endpoint, but from the certificate's signature, anyone can create a similar indexing service, allowing clients access to alternative discovery APIs via other web domains. 17

In sum: Instead of confining the imprimatur to a particular book or particular website, the DLL Review Registry Service assigns an imprimatur to the underlying data. The service allows that review to be instantly accessed regardless of where or how that text is being published. In this way, the imprimatur "travels" with the text wherever it goes, which is why we refer to it as the "traveling imprimatur". 18

Overview of technical implementation of the DLL Review Registry

The follow is a technical overview of the 1.0.0 DLL Review Registry Service. Given the fast pace of web development and our heavy use and modification of emerging technologies, the following description should not be viewed as documentation of the production system. For up-to-date documentation, the production system itself should be consulted (https://dll-review-registry.digitallatin.org). Instead, we offer here a description of the basic architecture of the review system with sufficient technical detail to make that pattern clear and intelligible. While the specific details of how that pattern is executed will undoubtedly change over time, it is the larger pattern and workflow that we aim to communicate here. 19

There are three main interactions that the DLL Review Registry Service anticipates: 20

- review creation
- review verification
- review retrieval

In order to ensure the long-term survival of these reviews and to avoid the pitfalls of a service with a single point of failure, we have attempted to enact all three aspects in the most decentralized way possible, and it will be the goal of further development of the system to continue moving in this direction as new technologies emerge and stabilize. 21

As mentioned above, this means that while for convenience the service will offer a public-facing site and a centralized index of reviews, the issued review certificates themselves are not dependent on the index for their existence or retrievability. Moreover, reviews will be created and published in such a way that the authenticity of their content is not dependent on the origin from which they are retrieved. That is, one need not infer the legitimacy of the review based on the domain name or service from which it was retrieved. Instead the certificates will carry within themselves sufficient information for new aggregation and new index services to be constructed from verified content. Thus, it will be possible for multiple indices of these reviews to exist at the same time. More importantly, this approach allows subsequent indices to easily replace the existing index service should that become necessary. 22

Review Creation

When a society is ready to submit a review for an approved edition, it can submit that review in one of three ways: via a webform, via a `POST` request to the DLL Review Registry API, or they can construct, sign, and publish the review themselves and then submit a minimal `POST` to register the existence of the review. The latter option may lack the convenience of the former options, but the option exists to emphasize that the creation and verification of these reviews is based on a set of rules and protocols, not on the existence of a particular website. Registration with the DLL is then simply a way of letting particular aggregators know about the existence of the review created according to community standards and protocols described here. The aggregator then functions as a convenient discovery endpoint whereby third party clients can discover the existence of reviews of interest. 23

In its most basic form, a review is a JSON document that takes as its template the OpenBadges specification.[8] The OpenBadges specification is primarily designed for assigning badges to people as recipients. In our case, we want to award badges or certificates not to people, but to documents: documents that can be both identified and retrieved by a unique fingerprint. 24

To do this we make use of another emerging technology called IPFS or the Interplanetary File System. With the help of IPFS we are not only able to generate a unique fingerprint (a SHA256 hash) of the file being reviewed, but via the IPFS network we are actually able to retrieve this file by its hash or content rather than its location.[9] This gives us an immediate level of decentralization and verification. In order to retrieve a specific certificate and to be certain that one is in fact receiving the desired certificate, a user does not need to care about the location (or domain) from which the file is retrieved. Requesting the certificate by its unique hash or fingerprint guarantees that the file received is the file with this fingerprint. Change a bit in the hash and a different file will be received. Change a bit the file and the file is now 25

a new file and will no longer be returned when the hash of the previous version is requested.

Our modified OpenBadges certificate uses the recipient field to identify the IPFS hash of the document or documents being reviewed, rather than a name, email, or url. In addition, the certificate itself should include the public key of the agency of institution that signed the document (see the section of verification below). This public key itself can be hashed and made addressable via the IPFS network as well being stored and addressed via the DLL Review Registry site. 26

A basic certificate can be seen here: 27


```
{
  "@context": "https://w3id.org/openbadges/v2",
  "id": "urn:uuid:2d817e2a-278a-40fe-9080-af73e483699b",
  "type": "Assertion",
  "recipients": [
    {
      "type": "hash",
      "identity": "QmUeCAUYSNDK1gDrEGcwzRhbbJxwXAsAu7VN4D2ppzD1gQ",
      "url":
"https://gateway.digitallatin.org/ipfs/QmUeCAUYSNDK1gDrEGcwzRhbbJxwXAsAu7VN4D2ppzD1gQ"
    },
    {
      "type": "hash",
      "identity": "QmUeCAUYSNDK1gDrEGcwzRhbbJxwXAsAu7VN4D2ppzD1gQ",
      "url":
"https://gateway.digitallatin.org/ipfs/QmUeCAUYSNDK1gDrEGcwzRhbbJxwXAsAu7VN4D2ppzD1gQ"
    }
  ],
  "issuedOn": "2018-07-30 15:47:19 +0000",
  "verification": {
    "type": "signedBadge",
    "publicKey": "QmZ5MFqMdZDunokFFQq5rKHxtURgjTg8e78cmzCypxcadG",
    "publicKey-url":
"https://gateway.digitallatin.org/ipfs/QmZ5MFqMdZDunokFFQq5rKHxtURgjTg8e78cmzCypxcadG"
  },
  "badge": {
    "image": "https://dll-review-registry.scta.info/maa-badge-working.svg",
    "issuer": {
      "id": "https://www.medievalacademy.org/",
      "image": "https://pbs.twimg.com/profile_images/1534408703/maa_symbol_small_400x400.png",
      "type": "Profile",
      "name": "Medieval Academy of America",
      "email": "info@themedievalacademy.org",
      "url": "https://www.medievalacademy.org/"
    },
    "criteria": {"narrative": "Meets the standards of a critical edition; judged
by the MAA as equal in quality to the kinds of editions that appear in the MAA
printed editions series or scholarly articles that appear in the MAA journal
'Speculum'."}
  }
}
```


The construction of this certificate is the main task of creating a review. The resulting document itself can be hashed and pinned to the IPFS network. But in order for this certificate to be discoverable, the hash itself needs to be given to a registry or index, such as the DLL Review Registry Service. 28

This could be done manually or through an API service. However, the more likely case is that an issuer will prefer to use an automated system for hashing the various files, publishing them to the IPFS network, as well as digitally signing the resulting certificate. 29

The DLL Review Registry Service provides this service through a webform, seen in Figure 1. 30

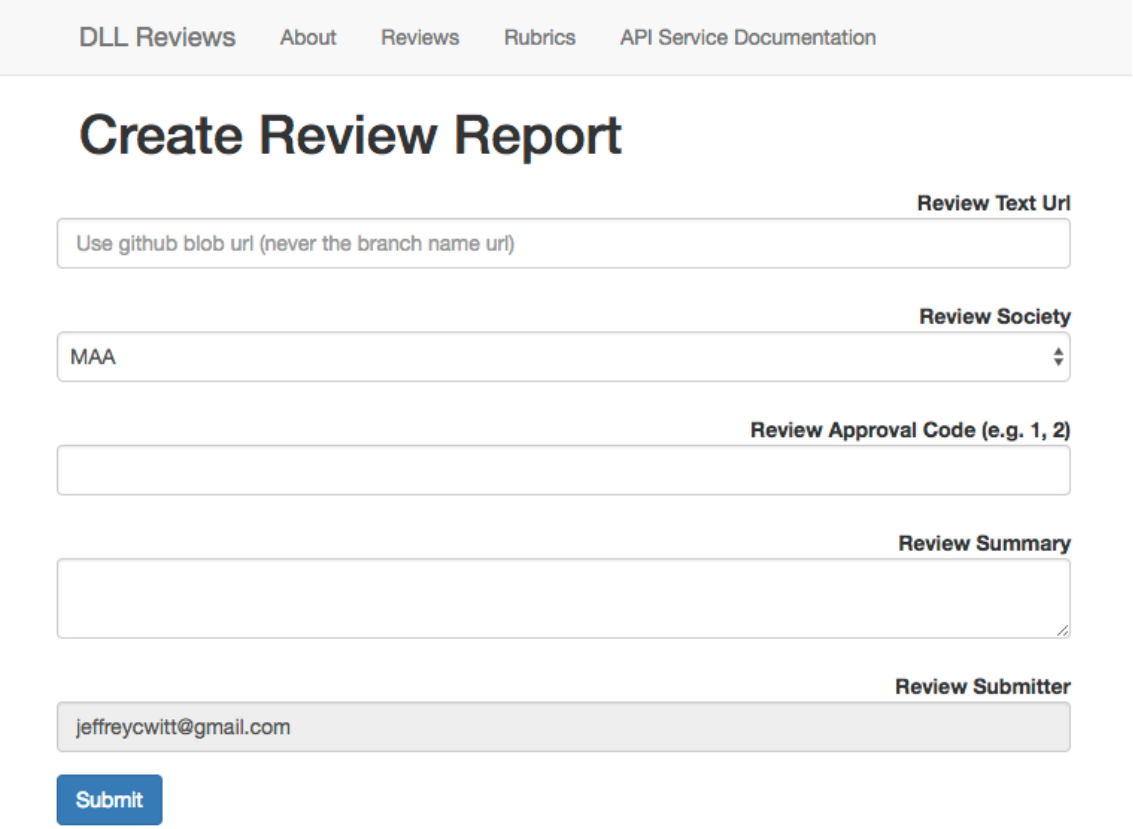

Figure 1.

Submission through the webform takes five basic parameters: 31

1. Review Text Url
2. Review Society
3. Review Approval Code
4. Review Summary
5. Review Submitter

.

The Review Text URL is a URL from which the document or documents being reviewed can be retrieved. If the document has already been hashed and pinned to the IPFS network, then an IPFS gateway address can be used, but any other URL will work as well. This information is not embedded in the certificate, but is used to create the recipient hash embedded in the issued certificate. 32

The Review Society field is the place for authorized users to select the society issuing the certificate. A society listed here means that the DLL Review Registry Service contains and protects a copy of the society's private key and is therefore authorized to issue certificates signed with this key (again, see below). The selection of a society will likewise determine which public key gets inserted into the review certificate. 33

The Review Approval Level is a field for reviewers or institutions to indicate the rubric used for the review. The possibility of indicating approval codes creates a way for reviews to escape the binary of an all or nothing review. A society, if it wishes, can create different kinds of reviews corresponding to different rubrics, allowing them to acknowledge quality work that meets different criteria. The code indicated here will determine which badge is issued in the resulting certificate. 34

An example of such a rubric might look like the rubric seen below in Figure 2. 35

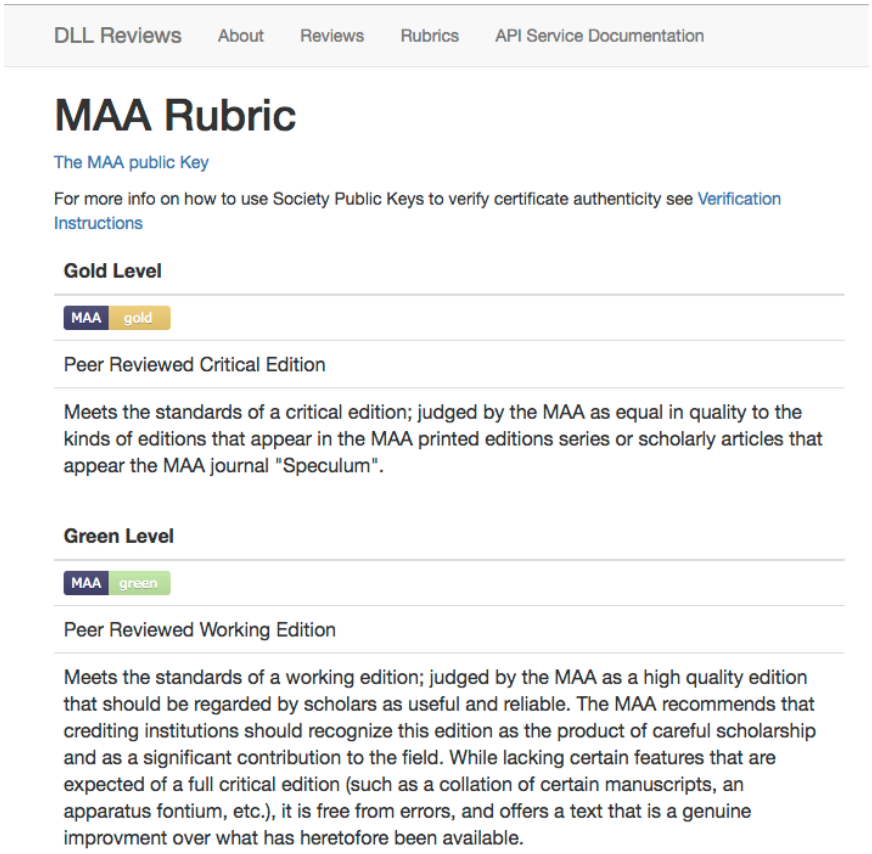

Figure 2.

A Review Summary is a field for leaving any pertinent details about the particular review. It could be inserted into the certificate itself, but at the present it is used merely as a place for a notice about reviews issued through the DLL Review Registry Service. 36

The Review Submitter field is, again, a field used only by the DLL Review Registry Service to keep track of the individual who created the review through the service and on behalf of a particular society. At present it is not used in the creation of a certificate. 37

Upon submitting the above information review, the DLL Review Registry Service performs several actions. 38

First, using the Review Text URL(s), the service retrieves the designated file(s) and pins the file to an IPFS node. At the same time it records the IPFS hash and the plain sha256 hash. (It should be noted that an IPFS hash is also a 256-hash, plus a prefix, encoded in base58, representing the file divided into a series of blocks.[10]) Pinning the file does not just produce a file hash, it also makes the file retrievable via its content hash on the IPFS network. This content then becomes accessible via any computer running an IPFS node. It is also accessible via http on a server running an IPFS gateway. The DLL Review Registry Service itself provides such a gateway (https://gateway.digitallatin.org/ipfs/) and the files remain retrievable here as well as from any other connected gateway.[11] 39

Second, the service constructs the review certificate using the newly minted recipient hash, the review approval code information, and the public-key of the reviewing society. 40

Third, the service then pins the newly generated certificate to the IPFS node generating an IPFS hash for the certificate. This hash is stored in the registry index alongside of the hash of the reviewed documents. 41

Fourth, the service digitally signs the generated certificate using GnuPG and the institution's private key. This process is described below. The resulting signatures are themselves hashed and pinned to the IPFS node. These hashes are finally stored in the index alongside the hash of the document and unsigned certificate. 42

Review Verification

While content addressability offers assurance that the document received is the document requested, a means of verifying that a given society really did create this review certificate is still needed. After all, the public key of any reviewing institution is public and therefore anyone could embed this public key into their own certificate. 43

The simplest way of providing this verification is through “hosted” verification. Namely, one can have confidence that a review was issued by a given society because that review was requested from the DLL Review Registry Service’s domain. However, in a distributed system where certificates can travel and are requested not by their location but by their content, we open the possibility for a certificate to be retrieved from any IPFS node or via http through any IPFS gateway. Thus, we need a mechanism of verification independent of the document’s point of origin. GnuPG digital signatures were designed for precisely this purpose.[12] 44

Using the private key of a reviewing society, we can generate a “signature” for a given certificate. This signature is a hash uniquely generated from the private key of the issuer and the data being signed. Importantly, the process can never be reversed. The private key can never be generated from the public key, the signature, the data, or any combination thereof. However, the fact that this signature was generated from the society’s private key can be verified against the combination of the data to be verified and the society’s public key. 45

In other words, a document (in this case, the certificate) can be verified as being issued by a society with a specific public key by putting the data and the signature together and checking it against the society’s public key. The check will confirm that this signature could only have been made by the combination of this society’s private key and the submitted certificate. Change one bit in the certificate (in this case, one would most likely be tempted to change the hash of the “recipient” document) or one bit in the public key and the signature will no longer pass verification. 46

At the present the DLL Review Registry Service creates two kinds of signatures allowing two methods of verification. 47

The first method is a detached signature. The detached signature is a hash stored in a separate file that is the result of the institution’s private key and the issued certificate. The DLL Review Registry Service again pins this resulting signature to the IPFS network and stores its hash as part of the indexed record. 48

Using a detached signature and previously imported public key, the certificate can then be verified as follows: `$ gpg --verify [path/to/signature] [path/to/certificate]` 49

The drawback to using a detached signature is that the signature is detached from the certificate and creates another file that needs a central indexing service to associate with the issued certificate. 50

In order to reduce this reliance on a central indexing service, the DLL Review Registry Service also produces a “clear-signed” version of the original certificate. This means a new version of the certificate is produced that includes both the original content of the certificate as well as a signature. With a clear signed certificate, the verification process can both verify the certificate and return the original un-signed certificate. The certificate can be validated with the following command: `$ gpg --verify [path/to/clear/signed/certificate]`. The original unsigned certificate can be generated from the signed certificate with `$ gpg -d -o original.txt [path/to/clear/signed/certificate]`. This resulting document “original.txt” can be further verified against the “unsigned-certificate.txt” by comparing the hashes of each file or by performing a unix `diff` command against both files, e.g. `$ diff original.txt unsigned-certificate.txt`. 51

From a verified certificate one can have complete confidence that the recipient document in the certificate is the precise document reviewed by the society. This is the case because the certificate does not point to a location but to the precise hash of the reviewed document. Addressing the document by its hash rather than its location via the IPFS network provides assurance that one is retrieving the very same document that was used to make the certificate and verified via its signature. Change one bit in the hash in the request and a different document will be retrieved. Change one bit in the hash in the certificate and the certificate will not longer be verified. 52

These verifications can be done entirely separately from the DLL Review Registry Service website, but the service also offers a web form and API through which certificates can be reviewed, as seen in Figure 3. 53

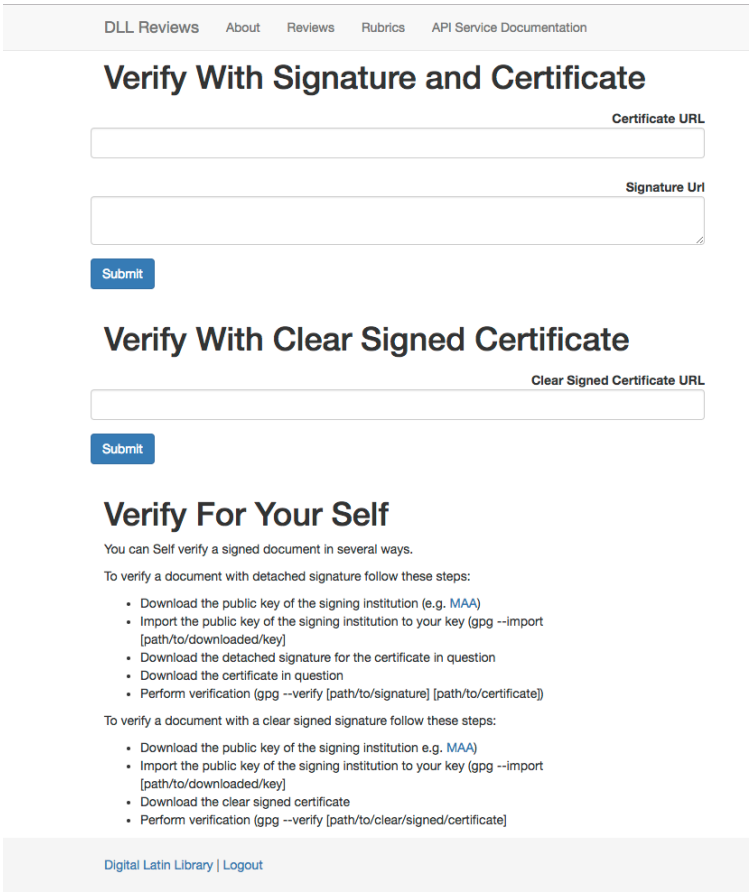

Figure 3.

Review Retrieval

Finally, third party services may want to be able to use the DLL Review Registry Service to discover verified review certificates for documents of interest and to be able to communicate the review status of those documents and verify that status back to end users. 54

Naturally, a human user can use the DLL Review Registry Service to browse reviews, view html renderings of reviews, verify certificates, and read the rubrics used by each society. This, however, is not the primary place an end user is expected to encounter this information. Rather, the primary expected pattern is that a client application will access this information using the DLL Review Registry Service API and offer that information back to the end user within its own interface. (See examples below.) 55

The API for data retrieval in the 1.0.0 version consists of three main routes: 56

1. `api/v1/review/:id` route
2. `api/v1/reviews/:hash` or `?url=document_url route`
3. `api/v1/verify?url=clearsigned_url`

.

The `api/v1/review/:id` route (1) allows a client that knows beforehand the indexed ID of a DLL Review Registry Service review to look up the review report. This is an ID that is unique to the centralized index service. The report in turn returns the information stored in the index including the document hash, certificate hash, detached signature hash, and the clear-signed certificate hash. 57

For a route like the following: `https://dll-review-registry.digitallatin.org/api/v1/review/f4b8dc2f-4d41-478d-877b-843b46e21283` a sample expected output would be: 58


```
{
  "id": "f4b8dc2f-4d41-478d-877b-843b46e21283",
  "review-society": "MAA",
  "date": "2018-07-30 11:16:57 UTC",
  "badge-url": "https://dll-review-registry.digitallatin.org/maa-badge-working.svg",
  "badge-rubric": "http://dll-review-registry.digitallatin.org/rubric/maa#green",
  "review-summary": "Review Summary",
```

```
  "sha-256": [
    "45304964c8bf9fb63737fa54e701b765baea0d950ff396c8fc686dd9bfda0416",
    "bab59377f29af62f1091ed367328db4abe96193a01f3b2c0e8615ba861e63317"
  ],
  "ipfs-hash": [
    "QmdgZhAFTepfXmUsxEGaVampJubZ7XMk4pC42uzHgBEgvy",
    "QmUxAedP9cDvC9dfwJrezwvHJWVD4w5UaJTHzLn37KEhMa"
  ],
  "submitted-url": [
    "https://raw.githubusercontent.com/scta-
texts/hiltalingencommentary/59eae077a22511531b5da383d402d031ca3b26b7/jhb-l1q10/clm26711_jhb-
l1q10.xml",
    "https://raw.githubusercontent.com/scta-
texts/hiltalingencommentary/59eae077a22511531b5da383d402d031ca3b26b7/jhb-l1q9/clm26711_jhb-
l1q9.xml"
  ],
  "submitted-by": "jeffreycwitt@gmail.com",
  "cert-ipfs-hash": "Qmbbotxgr1DBXTWXtDEkT2ZNh18ZPm73bHuAymTvNJQVim",
  "clearsigned-hash": "QmWZDUgZe12ALftn2G5z3GW4KMoLQXM9c4Ysb7yn8cRJTh",
  "detach-sig-hash": "QmYLprZk5uGJsaLjiKib31hiRdyyAQDuYznuZZpqgQfm7i"
}
```


From a decentralized perspective, the clear-signed certificate hash is the only piece of information needed. Every other piece of information can be extracted from it. 59

The clear-signed certificate contains the reference to the public key which can be used to verify the signed certificate and to generate the original certificate. The certificate itself contains the review report as well as the reference to the hash of the reviewed documents. 60

In most cases, however, it is not expected that a client will know the DLL Review Registry Service specific review ID. More likely, a client will be serving a document and it will want to be able to inform its user quickly as to whether or not this precise document has a review. This brings us to the second provided API route. 61

The `api/v1/reviews/:hash` or `?url=document_url` route (2) offers this possibility. 62

In this second, more common case, a client can supply a pre-computed sha-256 hash and receive a list of all reviews for any document with the same fingerprint or hash. Likewise, the client can supply, if it knows it, the IPFS hash and the API will return all reviews for documents having that hash. The IPFS hash and sha-256 hash will always correlate, meaning that a sha-256 hash will always return the same list as its corresponding IPFS hash and vice versa. 63

Finally, if a client does not want to pre-compute the hash of a document, it can simply supply the URL of the document it is serving using the ?url parameter. In this case, the DLL Review Registry Service will use the supplied URL to retrieve this file, compute its hash, and return a list of reviews that correspond to the text. 64

This is perhaps the most likely scenario for a consuming client, as it presumes no prior knowledge on the part of the client except for the location of the text it wants to check. By providing the DLL registry with the location of this text, it can easily check whether or not the precise version of this text has received a review and then receive pertinent metadata about that review. 65

For the following request `https://dll-review-registry.digitallatin.org/api/v1/reviews/QmdgZhAFTepfXmUsxEGaVampJubZ7XMk4pC42uzHgBEgvy` a sample expected response would appear as follows: 66


```
[
  { "id": "52cb5b0a-7ae0-441d-9da5-12534a9a082f",
    "review-society": "MAA",
    "date": "2018-07-30 11:15:11 UTC",
    "badge-url": "http://dll-review-registry.scta.info/maa-badge-working.svg",
    "badge-rubric": "http://dll-review-registry.scta.info/rubric/maa#green",
    "review-summary": "review summary",
    "sha-256": [
      "45304964c8bf9fb63737fa54e701b765baea0d950ff396c8fc686dd9bfda0416",
      "bab59377f29af62f1091ed367328db4abe96193a01f3b2c0e8615ba861e63317"
    ],
    "ipfs-hash": [
      "QmdgZhAFTepfXmUsxEGaVampJubZ7XMk4pC42uzHgBEgvy",
      "QmUxAedP9cDvC9dfwJrezwvHJWVD4w5UaJTHzLn37KEhMa"
    ],
    "submitted-url": [
      "https://raw.githubusercontent.com/scta-
texts/hiltalingencommentary/59eae077a22511531b5da383d402d031ca3b26b7/jhb-l1q10/clm26711_jhb-
```

```
l1q10.xml",
      "https://raw.githubusercontent.com/scta-
texts/hiltalingencommentary/59eae077a22511531b5da383d402d031ca3b26b7/jhb-l1q9/clm26711_jhb-
l1q9.xml"
    ],
    "submitted-by": "jeffreycwitt@gmail.com",
    "cert-ipfs-hash": "QmbhSB9CoRgCA3KBh5JXyavSaS2DymBZQQMRsqTiBSEGpC",
    "clearsigned-hash": "Qmc1165TM3QDsRVFboy56nHSrCF5w2RqkrMDqEoEvvnz3v",
    "detach-sig-hash": "Qmc1165TM3QDsRVFboy56nHSrCF5w2RqkrMDqEoEvvnz3v"
  },
  {
    "id": "f4b8dc2f-4d41-478d-877b-843b46e21283",
    "review-society": "MAA",
    "date": "2018-07-30 11:16:57 UTC",
    "badge-url": "http://dll-review-registry.scta.info/maa-badge-working.svg",
    "badge-rubric": "http://dll-review-registry.scta.info/rubric/maa#green",
    "review-summary": "review summary",
    "sha-256": [
      "45304964c8bf9fb63737fa54e701b765baea0d950ff396c8fc686dd9bfda0416",
      "bab59377f29af62f1091ed367328db4abe96193a01f3b2c0e8615ba861e63317"
    ],
    "ipfs-hash": [
      "QmdgZhAFTepfXmUsxEGaVampJubZ7XMk4pC42uzHgBEgvy",
      "QmUxAedP9cDvC9dfwJrezwvHJWVD4w5UaJTHzLn37KEhMa"
    ],
    "submitted-url": [
      "https://raw.githubusercontent.com/scta-
texts/hiltalingencommentary/59eae077a22511531b5da383d402d031ca3b26b7/jhb-l1q10/clm26711_jhb-
l1q10.xml",
      "https://raw.githubusercontent.com/scta-
texts/hiltalingencommentary/59eae077a22511531b5da383d402d031ca3b26b7/jhb-l1q9/clm26711_jhb-
l1q9.xml"
    ],
    "submitted-by": "jeffreycwitt@gmail.com",
    "cert-ipfs-hash": "QmbbotxgrlDBXTWXtDEkT2ZNh18ZPm73bHuAymTvNJQVim",
    "clearsigned-hash": "QmWZDUgZe12ALftn2G5z3GW4KMoLQXM9c4Ysb7yn8cRJTh",
    "detach-sig-hash": "QmYLprZk5uGJsaLjiKib31hiRdyyAQDuYznuZZpqgQfm7i"
  }
]
```


In the above response one can see that two separate reviews exist for a file with identical content. 67

The API also allows easy filtering of different kinds of reviews. If a user or application only wants to see a review by the MAA (or another specific society) and exclude all other reviews, it can simply add a parameter to the request like so: `http://reviews.digitallatin.org/api/v1/reviews/45304964c8bf9fb63737fa54e701b765baea0d950ff396c8fc686dd9bfda0416?society=MAA` This request will return an array of only those reviews that come from the MAA. 68

Finally, `api/v1/verify?url=clearsigned_url` (3) allows clients to verify clear-signed signature. 69

For the route `api/v1/verify?url=clearsigned_url`, a client can provide the URL of a clear-signed certificate and the API will return a very simple response as follows: 70


```
{
  "verification-message": "gpg: Signature made Mon Jul 30 07:16:58 2018 EDT
using RSA key ID 76160352 gpg: Good signature from \"Medieval Academy of America
(Keys for Medieval Academy of America)\""
}
```


The expected pattern is that a client could offer its users access to the review certificate for a given document by first sending the URL of the document to the Review Registry Service. In the response, the client would find the hash of the unsigned and clear-signed signature. The client can then offer these certificates to end users. If end users want to verify the authenticity of the certificate, clients can use the retrieved url of the clear-signed certificate and the DLL Review Registry `/api/v1/verify` route to offer end users a verification. 71

A high-level illustration of anticipated interactions between the DLL Review Registry and Client systems can be visualized below in Figure 4. 72

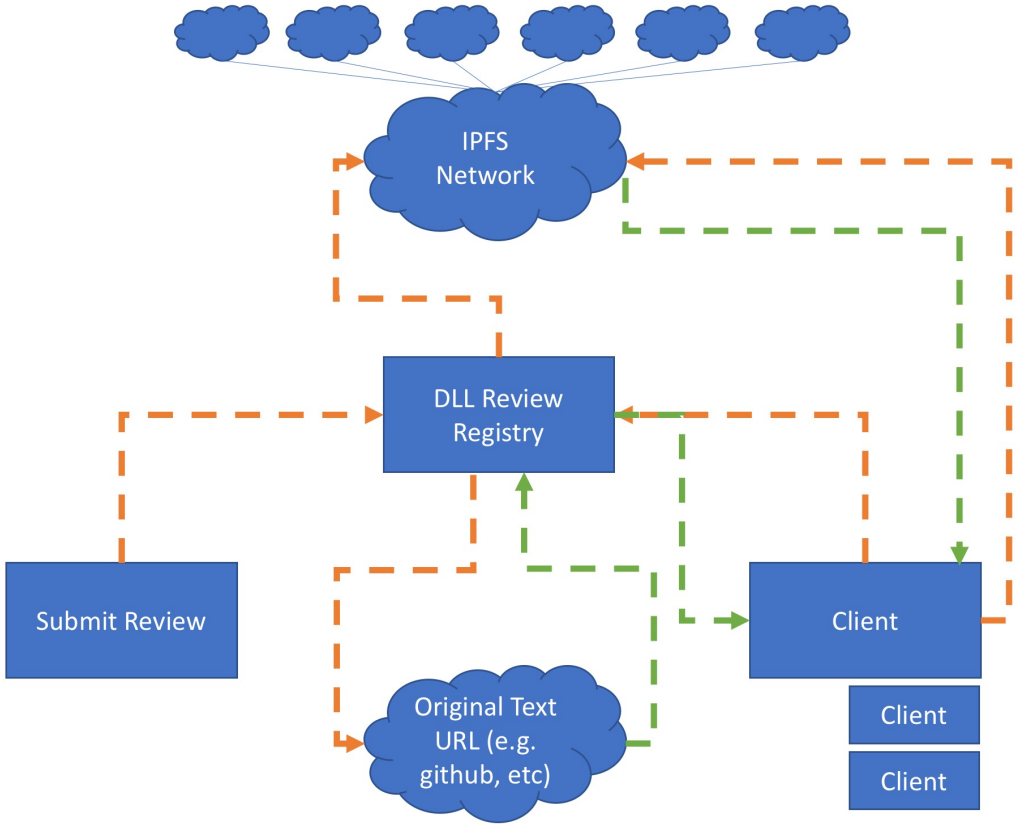

Figure 4.

Below are three brief examples of third party clients using the DLL Review Registry Service. 73

Figure 5 shows the LombardPress' LombardPress-Web (Lbp-Web) application displaying a critical edition. Lbp-Web does not contain any text on its server, but uses the Scholastic Commentaries and Texts Archive to locate data sources based on various query parameters, then it makes a further request for that data, which can be distributed anywhere on the web, and finally it displays that data to the end user. In this case, Lbp-Web has discovered a data source, an XML file for “Lectio 1” of Peter Plaoul’s *Commentary on the Sentences*. After retrieving this XML file, it wants to alert the end user to whether or not the data source has been peer reviewed. To do this, it asynchronously sends the location of that same file to the DLL Review Registry Service API and learns two things. It learns the IPFS hash of the data source and is able to provide that hash to users, so that users can always cite the precise data source they are viewing and retrieve that data source from any node on the IPFS network. Second, it learns whether or not a review certificate exists for which this data source is a recipient. If so, it crawls the certificate, discovers the imprimatur granted by the MAA, and displays the corresponding insignium of that imprimatur to the end user, along with a link back to the certificate and the DLL Review Registry Service review record. 74

Figure 5.

Figure 6 provides a second example of basically the same process. What is important here is the emphasis on the way the same imprimatur can travel with the data despite the highly varied use of the data in question. In the Ad Fontes application — an application designed to allow users to explore individual quotations throughout the Scholastic corpus — when a user requests to see the context paragraph of a given quotation, the application knows that this paragraph has been pulled from a larger data source (a larger XML file). At the same time that the application is showing the end user the context paragraph, it can also send an asynchronous request to the DLL Review Registry Service to check if there have been any reviews for the data source from which this context paragraph has been extracted. If so, it can display the review badge and data source hash alongside the context paragraph, as seen in the bottom right corner of Figure 6.

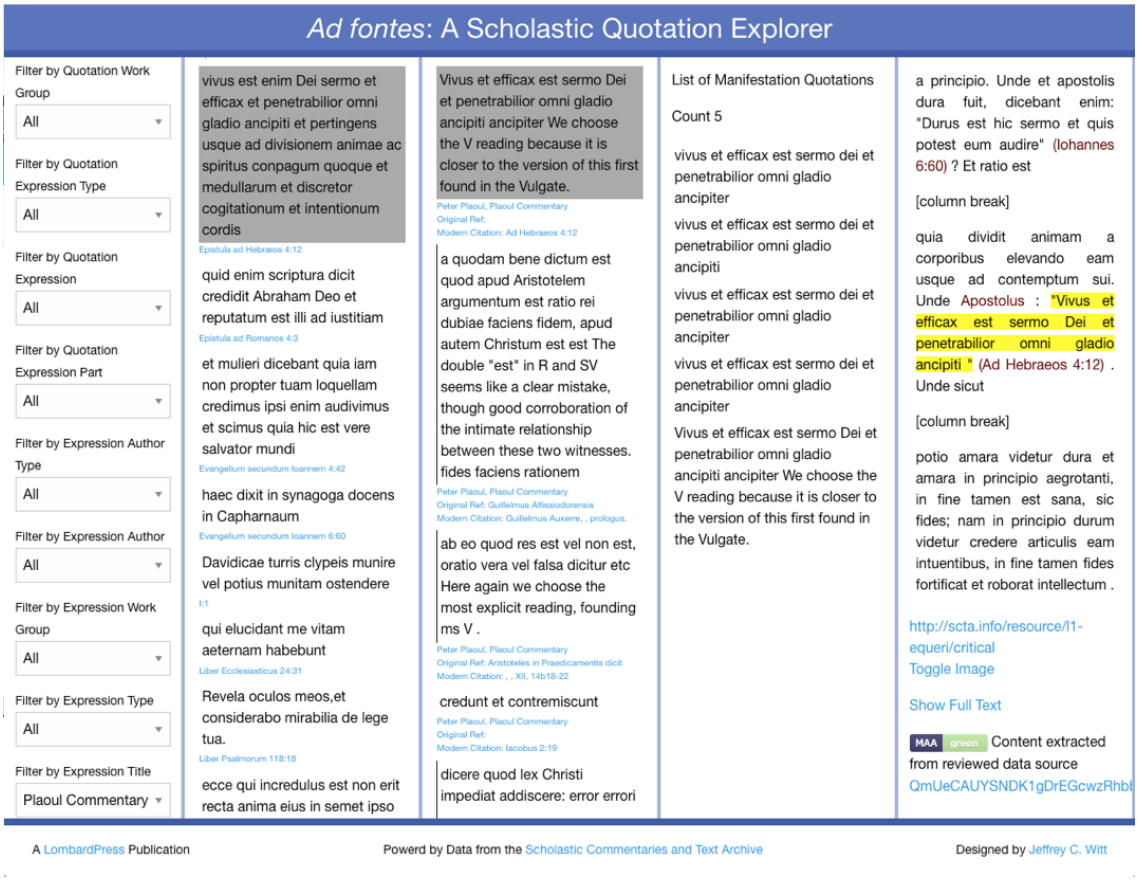

Figure 6.

Finally, Figures 7 and 8 illustrate the way the travelling imprimatur can be implemented in an even wider array of applications to provide users confidence when merging content in a Linked Data world. Figure 7 illustrates how Linked Data Notifications (LDN) have been used to make announcements about related but distributed content. In this case the Scholastic Commentaries and Texts Archive has published a transcription via the IIIF Open Annotation Model about a manuscript owned by the Bayerische Staatsbibliothek and made accessible via a IIIF Manifest. Through the use of this Linked Data, LDN, and the Mirador-LDN-Plugin[13], this related content can be brought together at runtime. However, librarians and end users alike want to have confidence in the content they bring into their research environment. With the traveling imprimatur, the LDN notification can either report the existence of the review certificate or the Mirador-LDN-Plugin can be modified to perform the check against the DLL Review Registry Service itself. In Figure 7, one can see that a user is given the option of importing two different pieces of related content into the workspace, only the second of which shows a review badge indicating that the data comes from a reviewed data source. Here administrative gate-keeping is removed; all available content can be announced and shown to the end user. The end user can then use indicators of quality control (like the DLL traveling imprimatur) to decide which kind of content she wants to bring into her workspace.

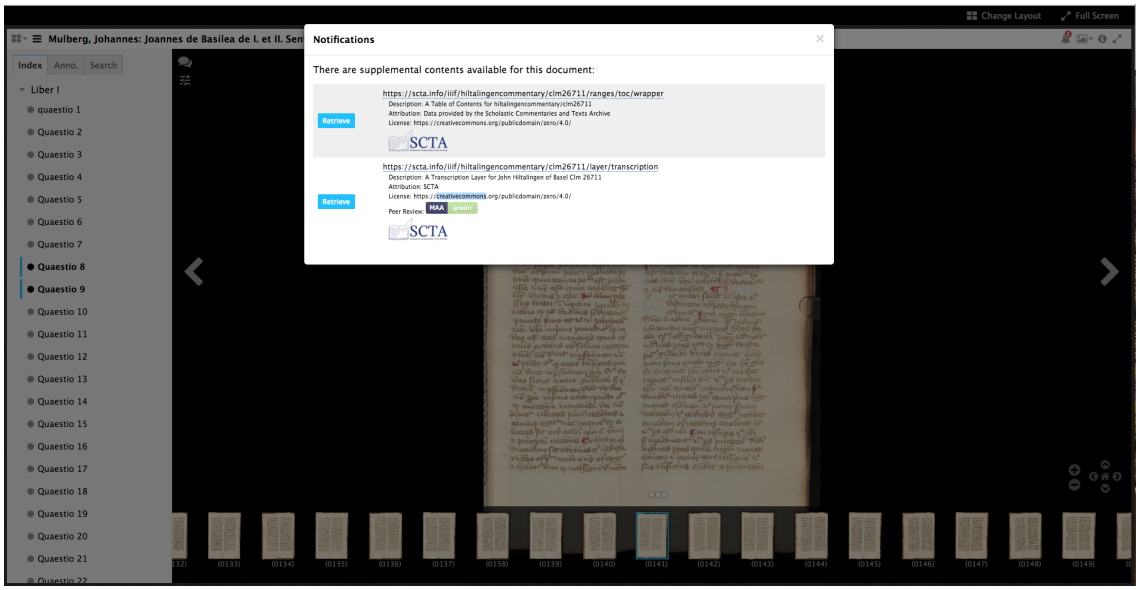

Figure 7.

Figure 8 offers a second example of this use of the traveling imprimatur. This time the additional content for the British Library Manuscript has already been imported. The visual imprimatur and accompanying IPFS hash for the data source from which this transcription was extracted are provided for the end user. The end user can click on the imprimatur to learn more about the review and the rubric used to make the review or can use the IPFS hash to access the raw data source directly. 77

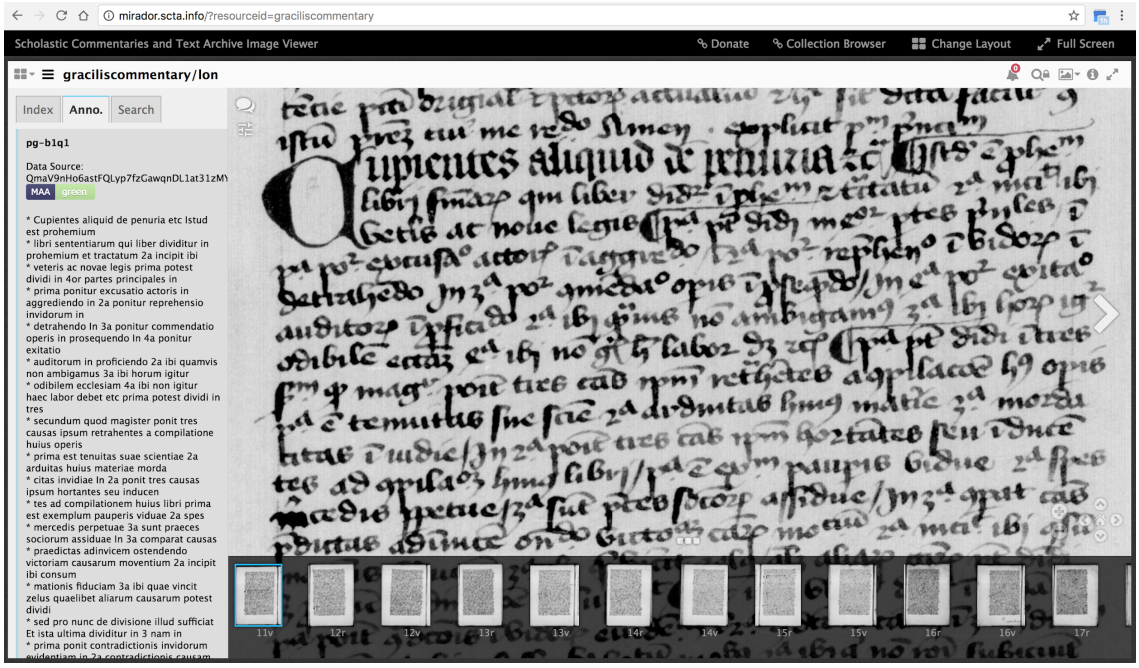

Figure 8.

Conclusion

It is important to acknowledge that web technologies are evolving at a rapid pace. This includes the web technologies described above. New possibilities are emerging every day. Accordingly, it needs to be reiterated that the central import of the present article is to describe the actualization of an idea and the general architecture of that achievement rather than the particular details of its achievement. Above we have tried to show how we have built a system that moves the idea of the “Traveling Imprimatur” to a reality in production. Beyond providing a production service, this actualization is also meant to show that social rather than technical obstacles are causing delays in de-coupling the distinct and separable functions of publication and quality control. 78

Today, the need to demonstrate that this idea can be actualized is urgent because of the persistent temptation to view the construction of a digital edition as something equivalent to or on par with a printed book, where the edition itself is bound to and identified with a particular 79

visual presentation of that data.[14] This temptation invites us to blur the distinction between data and presentation, in effect siloing and embedding our data in technology stacks that are difficult to maintain and more importantly preventing us from re-using this data for new and unanticipated purposes. Part of this temptation naturally comes from within, from our familiarity with printed books, and the understandable tendency to reproduce what is familiar within a new medium. But another part of this temptation comes from the political structures that surround our editorial work, the social apparatuses in which we live and breath. Like a fish unaware of the water that surrounds it, we frequently and uncritically adopt the pathways that are suggested by the surrounding social structures, which shape the methods of research we adopt. If, despite a great personal desire to exploit the possibilities of the new digital medium, the social structures that evaluate and reward our work continue to reinforce a paradigm that is no longer suited to the new medium, if these social structures continue to treat visual websites as editions themselves and fail to recognize the distinction between data and presentation, then the tendency of the field will be, by and large, to continue creating siloed and unconnected digital editions, despite our awareness that there is a better way.

Notes

[1] Grant numbers 11200693, 21400643, and 21500706.

[2] For an overview of the protean nature of the term *digital edition* see [Sahle 2016a]. See also [Robinson 2002], [Ducourtieux 2004], [Shillingsburg 2006, 80–102], [Gabler 2010], and the essays collected in [Driscol and Pierazzo 2016].

[3] See the essays collected in [Bleier et al. 2018] for many views on the relationship between edition and interface. As for encoded data being a presentational form, the fact that a goal for XML was that it should be “human-legible and reasonably clear” ([Bray et al. 2008]) demonstrates that the encoding is a way of presenting data directly to human readers. Consider also that the Text Encoding Initiative defines *encoding* as “any means of making explicit an interpretation of a text.” ([TEI 2019]) Thus, XML presents data to machines and humans; additionally, TEI XML presents an interpretation of data to machines and humans.

[4] Cf. [Shillingsburg 2006] 94: “Creating an electronic edition is not a one-person operation; it requires skills rarely if ever found in any one person. Scholarly editors are first and foremost textual critics. They are also bibliographers and they know how to conduct literary and historical research. But they are usually not also librarians, typesetters, printers, publishers, book designers, programmers, web-masters, or systems analysts.”

[5] See [Cayless 2018] for a description. The code for the viewer is available at https://github.com/DigitalLatin/viewer. On CETEIcean, see [Cayless and Viglianti 2018].

[6] For more on this research, see https://digitallatin.org/library-digital-latin-texts/data-visualization

[7] For an introduction to cryptographic hashes see [Paar Pelzl 2009, 293–317].

[8] See also https://www.imsglobal.org/sites/default/files/Badges/OBv2p0/index.html

[9] See for example, [Benet 2014].

[10] See [Benet 2014, section 3.1] for a description of the hash pattern. According to https://github.com/richardschneider/net-ipfs-core IPFS uses the multihash project https://github.com/multiformats/multihash for the production of IPFS hashes.

[11] For example, https://ipfs.github.io/public-gateway-checker/ is a list of known IPFS gateways.

[12] We’re also looking into the use of JSON Web Signatures (JWSs) as recommended by the OpenBadges specification. At present the use of JSON Web Signatures does not appear nearly as stable or well documented as the GnuPG standard which was introduced in 1997 and has a strong history of use and documentation. In the future, it would not be difficult to issue certificates signed with JSON Web Signatures alongside certificates signed with GnuPG.

[13] For more on the use of Linked Data Notifications and IIIF and the RERUM LDN Inbox, see https://centerfordigitalhumanities.github.io/inbox-docs/#!/ and https://centerfordigitalhumanities.github.io/inbox-docs/#!/specifications

[14] For a good description of this complaint among many, namely the temptation and tendency of digital editions to repeat the paradigm of the printed book in digital form, see [Zundert 2016] and [Sahle 2016b].

Works Cited

Benet 2014 Benet, Juan. “IPFS — Content Addressed, Versioned, P2P File System” (2014) https://arxiv.org/pdf/1407.3561.pdf.

Bleier et al. 2018 Bleier, Roman, Martina Bürgermeister, Helmust W. Klug, Frederike Neuber, and Gerlinde Scheider. *Digital Scholarly Editions as Interfaces. Schriften des Instituts für Dokumentologie und Editorik* 12 (2018).

Bray et al. 2008 Bray, Tim and Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, François Yergeau. *Extensible Markup Language (XML) 1.0 (Fifth Edition).* W3C Recommendation 26 November 2008. https://www.w3.org/TR/REC-xml/.

Cayless 2018 Cayless, Hugh. “Critical Edition and the Data Model as Interface.” In Roman Bleier, Martina Bürgermeister, Helmust W. Klug, Frederike Neuber, and Gerlinde Scheider (eds.), *Digital Scholarly Editions as Interfaces. Schriften des Instituts für Dokumentologie und Editorik*, pp. 249-263. *Schriften des Instituts für Dokumentologie und Editorik* 12 (2018).

Cayless and Viglianti 2018 Cayless, Hugh, and Raffaele Viglianti. “CETEIcean: TEI in the Browser.” In *Proceedings of Balisage: The Markup*

Conference 2018. Balisage Series on Markup Technologies, vol. 21 (2018). https://doi.org/10.4242/BalisageVol21.Cayless01.

Driscol and Pierazzo 2016 Driscoll, Matthew James and Elena Pierazzo, eds. *Digital Scholarly Editing: Theories and Practices*. Cambridge: OpenBook Publishers (2016). https://www.openbookpublishers.com/product/483/digital-scholarly-editing--theories-and-practices.

Ducourtieux 2004 Ducourtieux, Christine. “L’èdition électronique en quête de définitions(s).” *Le médiéviste et l’ordinateur, Histoire médiévale, informatique et nouvelles technologies* 43 (2004). http://lemo.irht.cnrs.fr/43/43-02.htm.

Gabler 2010 Gabler, Hans Walter. “Theorizing the Digital Scholarly Edition.” *Literature Compass* 7.2 (2010): 43–56.

Paar Pelzl 2009 Paar, Christof and Jan Pelzl. *Understanding Cryptography: A Textbook for Students and Practitioners.* Springer (2009).

Robinson 2002 Robinson, Peter. “What is a Critical Digital Edition?” *Variants: The Journal of the European Society for Textual Scholarship* 1 (2002): 43–62.

Sahle 2016a Sahle, Patrick. “What is a Scholarly Digital Edition?” In Matthew James Driscoll and Elena Pierazzo (eds.), *Digital Scholarly Editing: Theories and Practices*, pp. 19-40. Cambridge: OpenBook Publishers (2016).

Sahle 2016b Sahle, Patrick. “Zwischen Mediengebundenheit Und Transmedialisierung”. *Issues* 30 (2016). https://doi.org/10.1515/9783110223163.0.23.

Shillingsburg 2006 Shillingsburg, Peter L. *From Gutenberg to Google: Electronic Representations of Literary Texts.* Cambridge : Cambridge University Press (2006).

TEI 2019 TEI Consortium, eds. “v. A Gentle Introduction to XML.” *TEI P5: Guidelines for Electronic Text Encoding and Interchange.* Version 3.6.0. July 2019. https://www.tei-c.org/release/doc/tei-p5-doc/en/html/SG.html (date of access: 2019-09-18).

Zundert 2016 Zundert, Joris van. “Barely Beyond the Book?” In Matthew James Driscoll and Elena Pierazzo (eds.), *Digital Scholarly Editing: Theories and Practices*, pp. 83-106. Cambridge: OpenBook Publishers (2016).