A DIGITAL PUBLICATION AND ARCHIVAL PLATFORM FOR LEGAL DOCUMENTS

The Uniform Electronic Legal Material Act (UELMA) is a uniform act developed by the Uniform Law Commission that provides governments with "an outcomes-based approach to the authentication and preservation of electronic legal material." Uniform Electronic Legal Material Act (July 2011). It mandates that official electronic legal materials be (1) authenticated, (2) preserved, and (3) accessible. Starting with Colorado in 2012, nearly twenty states have enacted some version of UELMA.

As an outcomes-based approach, UELMA does not dictate a method of implementation. And given the complexity of digital storage and authentication, the range of satisfactory implementation methods is quite wide.

The District of Columbia enacted its version of UELMA in 2017. The District utilizes the Open Law Platform to publish its laws and the Code of the District of Columbia in an UELMA-compliant manner. This white paper discusses UELMA compliance as implemented by the Open Law Platform. Using a combination of plaintext XML, the open source distributed version control system called Git, and strong encryption, the Open Law Platform creates a repeatable process for authenticating and preserving electronic legal materials.

It is our hope that this document can highlight important considerations and provide a roadmap for governments wishing to publish their legal materials in compliance with UELMA.

Open Law Library
www.openlawlib.org

March 2018
Vincent Chuang and David Greisen

INTRODUCTION

The Open Law Platform, developed and maintained by the not-for-profit Open Law Library, is a software system created for the purpose of publishing laws, codes, legal interpretations, and any other legal document produced by a government. As part of taking a digital-first approach to legal publishing, the Open Law Platform incorporates UELMA compliance as a core component of the platform.

The Council of the District of Columbia is using the Open Law Platform to publish its laws and code (https://code.dccouncil.us) and provides a case study for replicating key features and processes at other jurisdictions. XML representations of the District's laws and codes can be found at https://github.com/dccouncil/dc-law-xml.

The Platform's version of UELMA compliance is modeled on brick-and-mortar libraries. Lessons about readability over time, information redundancy, version history, and authentication have been learned over centuries in the physical world. And it is useful to apply many of those ideas when considering digital preservation and authentication under UELMA.

TERMINOLOGY

The Council of the District of Columbia is a *Publishing Entity*. As a *Publishing Entity*, the Council is responsible for publishing and authenticating the *Library* of official legal materials relevant to itself. The Council's *Library* contains various *Documents*, including rapidly changing documents, like the entire District of Columbia Code, and static documents, like individual laws. Another *Publishing Entity* could be the Executive Office of the Mayor, and its *Library* could include *Documents* such as the DC Municipal Regulations and the DC Register.

An important difference between a *Library* in the Open Law Platform and a brick-and-mortar library arises in the context of time. The contents of a physical library might change over time, but you can only ever visit the library as it is today. That is to say, if Harvard Law Library throws away its copy of *A Wrinkle in Time*, the library is still the Harvard Law Library, but you can never travel back in time to read *A Wrinkle in Time* there. An Open Law Platform *Library* consists of a snapshot of every version of the library as it has ever existed. For instance, on January 1, 2018, the *Library DC-Law-XML* may have contained one thousand laws. We would refer to that *Library* as *DC-Law-XML* as of January 1, 2018. On January 2, 2018, the Council may pass a new law and add it to *DC-Law-XML*. Unlike a traditional library, you can visit the *Library* as it existed on January 1 or as it existed on January 2.

A *Consumer*, like a citizen of the District, can view *Documents* within a *Library* or download the entire library. And *Hosting Entities*, such as law libraries, can download and host a copy of an entire official *Library*. For instance, if the Harvard Law Library wished to host an authenticatable copy of the Council's *Library* on the Harvard Law Library website, it could do so, just as it could purchase and host an official paper copy of the District of Columbia Code.

CONSIDERATIONS

In addition to UELMA itself, the Open Law Platform was designed with several related and overlapping considerations.

Time

Legal documents have a long history, and that history is itself substantively valuable. As a result, the Open Law Platform is created with the intention that every version of the content it publishes be accessible and authenticatable long into the future. And because legal history is long, this means capturing and maintaining large volumes of documents. The Council's *Library* is only two years old, yet contains more than thirty thousand pages of laws and code, and is growing by over five thousand pages annually. *Libraries* must be manageable, usable, and responsive even while containing orders of magnitude more data than traditional libraries.

Authentication

The authentication scheme must also be robust against a wide range of factors from the perspective of *Publishing Entities*, *Hosting Entities*, and *Consumers*.

Publishing Entities are governments, and governments vary widely in the number of personnel, institutional capacity, and organizational structure. The authentication process must be usable in these varying environments. It must be possible, for any government, to clearly, easily, and securely convey (1) *when* a document was published, (2) *who* published it, and (3) the *authority* of that person to do so. All three questions can be answered with an appropriately designed cryptographic signing framework.

In order to be robust over time, the framework must be resilient to the loss or compromise of private cryptographic keys. The system must also provide for restoration in the event of a government-scale catastrophe: there must be a mechanism for restoring a *Library* after all encryption keys have been lost. And the system must operate on government time scales. Because published documents are intended to be used over the course of decades, accessibility (by way of readability or cryptographic scheme) must keep pace with changing technology. A *Library* must be accessible and authenticatable long after the *Publishing Entity* has abandoned it

and moved on to other technologies, just as an official paper copy is at a law library even if the government no longer has that particular version.

A *Hosting Provider* should be able to host authenticatable versions of a *Library* for its patrons. For the *Consumer*, a *Library* must function across every use case. In situations in which the delivery network is compromised (such as hackers taking over the *Publishing Entity*'s web server), a *Consumer* must still have confidence that the *Library* being viewed is authentic. As with physical text, a *Library* should be accessible and authenticatable even without an internet connection. Because *Libraries* have a version component, a *Consumer* should be able to ascertain information regarding both authenticity and versioning information, akin to checking publication information inside a book.

Redundancy

The system must also be distributed. Just as Harvard Law Library and USC Law Library may both carry a copy of *A Wrinkle in Time*, a *Consumer* should be able to access a *Library* from a *Hosting Entity* and be able to confirm that the *Library* is the same as one acquired from the original *Publishing Entity*. Even if a *Consumer* can never access the original *Library* from the original *Publishing Entity*, the hosted *Library* should be authenticatable without reference to the original.

THE OPEN LAW PLATFORM SOLUTION

With these various considerations in mind, the Open Law Platform utilizes a combination of technologies, including XML, Git, and strong encryption, to implement a set of authentication techniques.

XML

The Open Law Platform stores almost all documents as plaintext XML. By using plaintext instead of a binary format (e.g., PDF), a *Library* and its *Documents* are virtually guaranteed to be readable for decades to come without relying on legacy software. Plaintext also requires considerably less storage space than binary formats. For the Council, 30,000 pages of XML can be stored in 100 megabytes, while only 10,000 pages of PDFs require fifty times the space when compressed and 500 times the space when uncompressed. This difference means it is feasible to store every version of a plaintext document in less space than a single version in PDF.

XML also has the advantage of being able to store the structure of a document, instead of just presentation information (i.e., how something looks on a screen). This means documents can be converted into any display format in the future and not be tied to any specific software. Together,

these benefits of XML make it possible to satisfy the need for usability over time, ability to store large amounts of historical information, and speed of use.

A common concern with XML-based solutions is that XML can appear complicated and requires a different set of tools than most lawyers are used to using. This has resulted in very few UELMA-compliant XML implementations.

The Open Law Platform solves this problem in several ways. First, the platform focuses on making the XML very clean and simple, using, whenever possible, a jurisdiction's terminology to describe a document and its contents (e.g. Title, Chapter, Subchapter, and Section). The platform also stores metadata logically within the document, again using the same terminology as the jurisdiction.

Good tooling (i.e., software for viewing and editing the XML) also goes a long way to making XML more digestible. The platform provides a mix of custom XML schemas and software to ensure XML accuracy, as well as automatic error detection, and other smart editing capabilities. By focusing on user experience, lawyers familiar with the District's laws and code were able to navigate and understand XML representations of law and code with no training.

Converting documents into XML is itself a process. But again, good tooling can make the process feel seamless. The Open Law Platform includes Open Law Draft, a Microsoft Word plugin that helps drafters conform to their jurisdiction's style guides. Once the document conforms to the style guide, Draft can turn the document into correct XML without user input.

An XML-based solution has many benefits inherent in its format, with the biggest barriers being usability and conversion of existing documents into the format. A focus on user experience and good tooling can overcome these high hurdles. Success on this front reveals the downstream benefits of XML that ultimately outweigh the initial costs.

Git

The Open Law Platform stores XML (and any static PDFs) using the open source Git distributed version control system (https://git-scm.com/). In simplest terms, Git is a piece of software that keeps track of changes to one or more files (each group of one or more files collectively referred to as a "repository"), records the differences between new and old versions of one or more files, and maintains a history of the differences. It does so, in part, by providing the ability to sign each version with a unique cryptographic key (https://git-scm.com/book/id/v2/Git-Tools-Signing-Your-Work). This makes it possible to preserve different versions of documents as they change and creates an immutable chain of authenticatable versions back to the original.

Git makes it easy to copy an entire *Library* from one place to another and then keep the copy up-to-date with the original by just syncing changes. Because every copy of a *Library* has all the historical information and authentication information of the original, it is inherently fraud resistant. In the event a malicious actor attempted to modify the history of the original *Library*, the next time a copy attempted to sync with the now-fraudulently-modified "original", the copy would detect the modification of the history and reject the fraudulent history.

Git is free, open source, and available on virtually every platform. There are also many cloud services that provide Git access. Because of this wide availability, a *Library* that is stored as a Git repository can have all of its historical information hosted on a variety of physical machines located across a large geographic region. And every copy is easily authenticated.

Signing a Library

With XML and Git as the underlying technologies, the Open Law Platform implements specific processes to achieve the needed authentication outcomes.

At the government level, each employee who has authority to publish legal documents receives a smart card (e.g., https://www.yubico.com/products/yubikey-hardware/). A small group of employees (minimum of three, preferably five) or other trusted individuals creates an *Attesting Group*. Each member of the Attesting Group (an *Attestor*) has a smart card that they use to sign *Attestations of Authority*.

Once a threshold of *Attestors* (usually 50%) have attested that a particular person has authority to publish official documents, that person is a *Publisher* (as part of a *Publishing Entity*) and can sign new releases of a *Library*. If a *Publisher* leaves the organization or loses their key, the *Attestors* attest that the old key no longer has authority to publish. If an *Attestor* leaves the organization or loses a key, a majority of the remaining *Attestors* can attest that the old key is no longer valid and can also attest that a new key is a valid attestation key.

Normally, cryptographic signatures are very complicated or very brittle. This system, however, ensures that the system continues to work even if several keys have been lost or compromised. Moreover, encryption keys are stored on physical devices and protected by a password. Even if a jurisdiction's network is compromised, their keys are not.

Authenticating a Library

Attestations of a *Publisher*'s authority are stored in the *Library* itself. Thus, when a *Publisher* signs a *Library*, all the information needed for authentication is available within the *Library*.

This technique combined with the use of Git to create a cryptographically secured history and to create easily replicable repositories results in a robust authentication system for *Libraries*.

While a *Consumer* or *Hosting Entity* can confirm that all signatures and all attestations are valid back to the very first release of a *Library*, they will always require at least one out-of-band authentication (i.e., authentication via something other than the original receiving channel) to confirm the very first release. The design of the Open Law Platform aims to decrease the friction required to obtain out-of-band authentication.

For starters, once a *Consumer* or *Hosting Entity* has performed one out-of-band authentication, usually via a telephone call to the *Publishing Entity*, the use of Git to store a *Library* means any future updates can be confirmed authentic without external verification. Just as law libraries currently provide indirect authentication of paper laws—they buy the laws from the official publisher then represent to their users that these are the official laws—law libraries can download a *Library* from the official *Publishing Entity*, perform the single out-of-band authentication, and then represent to their patrons that these are official laws.

Once a *Library* is hosted by more than one *Hosting Entity*, it becomes possible to perform out-of-band authentication by comparing the various hosted *Libraries*. And this comparison can then be automated for ease of use by *Consumers*.

Importantly, this system works without relying on a public root certificate (like those underlying HTTPS) or a web service maintained by the *Publishing Entity*. If the web service goes down, or the *Publishing Entity* stops supporting the web service, the *Library* will still be fully available and authenticatable through the constellation of *Hosting Entities*. In root certificate based systems, compromising the root certificate means compromising all historical documents signed by the certificate. While it may seem unlikely that a root cert will be compromised, this is surprisingly common. Symantec, until recently one of the most trusted root certificate authorities, was forced by Google and Mozilla to divest itself of its root certificate in 2017 because of major systemic security violations. An authentication system premised on a public root certificate system is too fragile to provide authentication over decades. Instead, by intimately tying the authentication mechanism to the preservation mechanism, preserving the documents automatically preserves the authentication.

The discussion up to this point has been regarding *Archival Authentication*, i.e., downloading and authenticating an entire *Library* (along with all historical versions). Most users, such as lawyers and judicial staff will be performing *Transient Authentication* of particular versions of individual *Documents*. For these purposes, a web-based authentication service is ideal, as it makes it trivial for users to authenticate.

For most use cases, a *Consumer* that uses a *Library* by accessing an HTTPS-protected website or application programming interface (API) of the *Publishing Entity* can be generally certain of the authenticity of the *Library* being accessed. This method of use and authentication serves as the base case provided by the Open Law Platform.

For more advanced *Transient Authentication*, the Open Law Platform is designed to provide an authentication service through a website, an API, and plugins for all major browsers. The authentication service can assess authenticity by comparing a hash of a *Document* (e.g., as published by a *Hosting Entity*) against the hash of the same *Document* from a known-authentic *Library*. This method of authentication is particularly relevant when a *Consumer* accesses a *Document* hosted by a *Hosting Entity* as opposed to the original *Publishing Entity.* Additionally, because the authentication service can access the hashes of all versions of all authentic *Documents*, the authentication service can not only tell the user if a *Document* is authentic, but also tell the user when the version in question was created and if/when it was superseded by a newer version. Unlike other web authentication services, the Open Law Platform optionally provides the full cryptographic audit chain so an individual can confirm for themselves against a full copy of the *Library* that the *Document* in question is authentic. Further, the *Transient Authentication* service can be bootstrapped from any authentic library, allowing *Transient Authentication* even if a jurisdiction ceases hosting their own *Transient Authentication* Service.

Redundancy

Redundancy is built into the system because of the way repositories are stored using Git and because of the authentication process.

With respect to redundancy of information, the wide adoption of Git and the various commercially available Git hosting solutions means that anyone at any time can easily retrieve and host their own copy of a *Library*. This replicability means that *Libraries* can be quickly distributed across large geographic areas and can help recover from data loss. Moreover, each copy of a *Library* is cryptographically signed in a way that permits for corruption detection.

No less important and considerably more complex is the redundancy of authentication. If many *Hosting Entities* are constantly pulling down updates of fully authenticated laws, the constellation of entities can help a *Publishing Entity* recover from catastrophic losses (such as a natural disaster). If all *Attester* keys are lost in, say, a flood, a group of *Hosting Entities* can represent that a new set of *Attester* keys are official keys, helping to rapidly bootstrap a *Publishing Entity* back to an authenticatable state. Moreover, the presence of verifiably authentic copies held by *Hosting Entities* means that any new copies can be authenticated against those copies even if the original *Publishing* entity no longer exists.

IMPLEMENTATION COST AND OVERALL ASSESSMENT

UELMA compliance is a core feature of the Open Law Platform. There is no additional cost to implement UELMA for a jurisdiction already using the platform.

The initial cost of developing the Open Law Platform was significant, but it is now a fully generalized legal publishing platform that is available for any jurisdiction to use. Free Git repository hosting is available from several well-established commercial providers including GitHub, Bitbucket, and GitLab. A law library can set up its own archival copy of all *Libraries* published on the Open Law Platform using a five-year-old computer in one afternoon.

As of February 2018, version 1.0 of the Open Law Platform is complete and running for the District of Columbia. Documents published using the Open Law Platform can be found at https://code.dccouncil.us, and XML representations are available at https://github.com/dccouncil/dc-law-xml. Initial work on *Archival Authentication* is complete and is being rolled out to the Council; *Transient Authentication* is expected June 2018.