of describing any kind of performance. Its meaning is simpler for digital records than for analog recordings because digital records already reflect the sampling errors of recording performances that are continuous in time. The archivists expressing difficulty with dynamic digital objects do not express similar uncertainty about analog recordings of music.

Trustworthy Digital Object (TDO) Methodology

The TDO proposal focuses on methods for making the authenticity of preserved digital objects reliably testable and for assuring that eventual users will be able to render or otherwise use their contents. The objectives suggest solution components that can be nearly independently addressed:

- Content servers that store packaged works, and that provide search and access services.
- Replication mechanisms that protect against the loss of the last remaining copy of any work [8].
- Schema for packaging a work together with metadata that includes provenance assertion and reliable linking of related works, ontologies, rendering software, and package pieces with one another.
- Standard bibliographic metadata and topic-specific ontologies defined, standardized, and maintained by professional communities.
- A bit-string encoding scheme to represent each content piece in language insensitive to irrelevant and ephemeral aspects of its current computer environment.

To prepare the TDO that represents a work (see Figure 2), an editor converts each content bit-string into a durably intelligible representation and collects the results, together with standardized metadata, to become the TDO payload. In addition to its payload, each TDO has a protection block into which a human editor loads metadata and records relationships among its parts, and between it and other objects. The final construction step, executed at a human agent's command, is to seal all these pieces within a single bit-string with a *message authentication code*. In a valid TDO representing some version of an object, the bit-string set that represents the version is XML-packaged with registered schema; these bit-strings and metadata are encoded to be platform-independent and durably intelligible. TDO metadata includes identifiers for the version and for the set of versions of the work and the package includes or links reliably to all metadata needed for interpretation and as evidence. All these contents are packaged as a single bit-string sealed using cryptographic certificates based on public key message authentication and each cryptographic certificate is authenticated by a recursive certificate chain.

In the past, wax seals impressed with signet rings were affixed to documents as evidence of their authenticity. A contemporary digital counterpart is a message authentication code firmly bound to each important document. The structure and use of each TDO, emphasizing the metadata portions suggested by Figure 2, is described in [3]. The design includes the following features:

- Each TDO contains its own worldwide eternal and unique identifier and its own provenance metadata, and is cryptographically sealed to prevent undiscoverable changes;
- References to external objects are accompanied by their referents' message authentication codes;
- Certification keys are themselves certified. This recursion is grounded in the published and annually changed public keys of institutions that people trust to be honest witnesses. The stored results of this process chain constitute durable evidence of the TDO's publication date;
- Each person that edits a work being prepared for archival deposit nests or links the version he started with, thereby creating a reliable history;
- Each participant in the creation sequence usually is, or readily can become, acquainted with his predecessor and his successor. Thus the public keys that validate authorized version deliveries can readily be shared without depending on a Public Key Infrastructure (PKI) certificate authority. This arrangement avoids well-known PKI security risks.

Content represented with relatively simple and widely known data formats can be saved more or less "as is." For other data formats, [5] teaches how to encode any kind of content bit-string suggested by Figure 2 to be durably intelligible or useful. Its features include:

- That we enable each information producer to separate irrelevant information, such as operating system details, from information essential to his intentions, encoding only what's essential;
- Rewrite to the code of a Turing-complete virtual machine (extended to handle concurrency and real-time services)—an application of the Church-Turing thesis that any program or rule set producing a finite sequence can be implemented by a simple machine.
- And that such machines can themselves be