"Curation Micro-services: A Pipeline Metaphor for Repositories"

Stephen Abrams †
Patricia Cruse †
John Kunze †
David Minor ‡

† UC Curation Center, California Digital Library, University of California
‡ San Diego Supercomputer Center, University of California

Abstract

The effective long-term curation of digital content requires expert analysis, policy setting, and decision making, and a robust technical infrastructure that can effect and enforce curation policies and implement appropriate curation activities. Since the number, size, and diversity of content under curation management will undoubtedly continue to grow over time, and the state of curation understanding and best practices relative to that content will undergo a similar constant evolution, one of the overarching design goals of a sustainable curation infrastructure is flexibility. In order to provide the necessary flexibility of deployment and configuration in the face of potentially disruptive changes in technology, institutional mission, and user expectation, a useful design metaphor is provided by the Unix pipeline, in which complex behavior is an emergent property of the coordinated action of a number of simple independent components. The decomposition of repository function into a highly granular and orthogonal set of independent but interoperable micro-services is consistent with the principles of prudent engineering practice. Since each micro-service is small and self-contained, they are individually more robust and collectively easier to implement and maintain. By being freely interoperable in various strategic combinations, any number of micro-services-based repositories can be easily constructed to meet specific administrative or technical needs. Importantly, since these repositories are purposefully built from policy neutral and protocol and platform independent components to provide the function minimally necessary for a specific context, they are not constrained to conform to an infrastructural monoculture of prepackaged repository solutions. The University of California Curation Center has developed an open source micro-services infrastructure that is being used to manage the diverse digital collections of the ten campus University system and a number of non-university content partners. This paper provides a review of the conceptual design and technical implementation of this micro-services environment, a case study of initial deployment, and a look at ongoing micro-services developments.

Introduction

Information technology and resources have become integral and indispensable to the pedagogic mission of the University of California, with members of the UC community routinely producing and utilizing a wide variety of digital assets in their teaching, learning, and research activities. These assets represent the intellectual capital of the University; they have inherent enduring value and need to be managed carefully to ensure that they will remain available for use by future scholars. Within the UC system the newly-established UC Curation Center (UC3), one of five programmatic units of the California Digital Library, has a broad mandate to provide innovative solutions that ensure the long-term usability of the University's digital assets. While curation is not solely a technical undertaking – curation success is, for example, highly dependent on important human competencies, analysis, and decision making – a robust infrastructure in which to manage valuable digital content efficiently and effectively is nevertheless a necessary foundation.

As a central system-wide service provider to the ten UC campuses, UC3 is routinely asked to assume custodial stewardship for digital content in ever increasing number, size, and diversity of type. Furthermore, this content is often used and repurposed in novel contexts far removed from the intention of its original creators. Thus, the programmatic imperative of UC3 is to provide a curation environment that is comprehensive in scope, yet flexible with regard to local policies and practices, and the inevitability of disruptive changes in technology and user expectation. To meet these goals the UC3 infrastructure is based on the idea of *micro-services*, the decomposition of repository function into a highly granular and orthogonal set of independent but interoperable components that can be freely composed in strategic combinations towards useful ends. The paradigmatic metaphor for the micro-services approach is the Unix pipeline.

The Pipeline Metaphor

The pipeline concept was first proposed by Douglas McIlroy in 1964 and gained wide visibility through its integration in the Unix operating system in 1973 (Ritchie, 1980). A pipeline chains together a set of independent processes such that the output of a previous process becomes the input to a subsequent process. Although the local function of individual components can be extremely narrowly scoped, sophisticated global behavior is nevertheless an emergent property of the coordinated action. Due to the process coupling at the I/O level, pipelines are highly dependent on the stability of the public interface "contracts" exposed by the component processes.

The flexibility inherent to a pipeline serves a number of important purposes. By decomposing complex function into a set of simple constituent parts, the development and maintenance of those parts is simplified. This approach is consistent with prudent engineering practice as articulated in forms as varied as the philosophical statement of Occam's Razor ("*entia non sunt multiplicanda praeter necessitate* [entities must not be multiplied beyond necessity]"; Wikipedia 2010a) to the popular culture adage of Murphy's Law ("whatever can go wrong, will go wrong"; Wikipedia 2010b). The design principles underlying the pipeline metaphor have been generalized by UC3 into a preference for the small and simple over the large and complex, the minimally sufficient over the feature laden, the fully configurable over the prescribed, and the proven over the (merely) novel.

The advantages of the micro-services approach to curation infrastructure are manifold. Since each micro-service is small and self-contained, they are individually more robust and collectively easier to implement and maintain. Since the level of resource investment in any given service is small, the level of institutional commitment to that service is concomitantly small, so they are easier to deprecate and replace when they have outlived their usefulness; an important consideration given that curation over archival time-spans is best seen as a relay requiring periodic handoffs between a constantly evolving ecosystem of services and service providers (Janée *et al*. 2008). Since the micro-services are inherently amenable to flexible and strategic recombination, many purpose-built repositories can be easily constructed with the minimally necessary function for a specific administrative or technical purpose.

Design and Implementation

The initial repertoire of micro-services coalesces into four hierarchical levels (see Figure 1). The range of underlying function moves from preservation necessity towards curation sufficiency by maintaining the integrity of content state, managing content context, providing user-facing services, and enabling the enhancement of value.

	Value	**Annotation**	*of content by consumers*
		Notification	*of new content availability*
	Service	**Transformation**	*to create derivatives*
		Search	*of content and metadata*
		Index	*to enable fast search*
Curation		**Ingest**	*of content for curation*
Preservation	*Context*	**Characterization**	*to extract content properties*
		Inventory	*of curated content*
	State	**Replication**	*for safety*
		Fixity	*to verify bit-level integrity*
		Storage	*for long-term retention*
		Identity	*for long-term reference*

Figure 1 – Curation micro-services

The general principles of granularity and orthogonality are applied throughout the architecture, with each micro-service itself built up from smaller components. For example, the Storage service is modeled in terms of five conceptual entities: the *service* itself, which acts as a broker to an arbitrary number of *storage nodes*, each of which manages a storage sub-domain established to meet specific policy, administrative, or technical needs. Nodes manage *digital objects*, which can encapsulate an arbitrary number of *versions*, each of which is a set of *files* representing a discrete state of the object. (As a corollary, any change introduced to object state instantiates a new object version. Previous states are stored as a sequence of reverse deltas to minimize storage utilization yet support the easy re-instantiation of an arbitrary version.) Subsidiary systems and specifications for these entities include Content Access Node (CAN), Pairtree, Dflat, Checkm, and Reverse Directory Deltas (ReDD). (More information is available at <http://www.cdlib.org/services/uc3/curation>.)

All conceptual entities are defined in terms of a set of state properties and behaviors that can manipulate that state. Entity state information follows the Linked Data paradigm in including actionable links to related entities, when relevant (Bizer *et al*. 2007). For example, a version contains a back link to its object and forward links to all of its files. State properties are defined as semantic ontologies and can be reported in various expressions including ANVL (mail header-like name/value pairs), JSON, RDF/Turtle, RDF/XML, XHTML, and XML. Behaviors are first defined as abstract methods that are then mapped to specific interactive modalities. In general, service methods can be invoked through a RESTful API, a command line API, or a procedural interface with various language bindings (currently, either Java or Perl).

The combinatoric power of the micro-services approach is illustrated by the ingest workflow that coordinates the actions of four components: Ingest, Identity, Storage (with subsidiary invocation of encapsulated storage nodes), and Inventory (see Figure 2). The Inventory service manages a triple store-based metadata catalog for all managed content. This catalog is intended as an optimization to support administrative and technical queries, and in general is a duplicative subset of the authoritative metadata that is expressed in files managed by the Storage service. Thus the Inventory catalog can always be fully reinstantiated, if necessary, from the metadata-of-record in the Storage service.

Conclusion

In order to facilitate the application of UC Curation Center service offerings to new campus constituencies, and to respond to the increasing number, size, and type diversity of digital content, the underlying curation

infrastructure must be easily adaptable to local needs and practices. An architectural approach based on the principles underlying the pipeline metaphor in which curation function is embodied in a set of granular and orthogonal micro-services best provides the necessary deployment flexibility, while also simplifying development and maintenance effort. Service interoperability is facilitated by strict conformance to the behavioral semantics of well-defined public interfaces. This permits comprehensive curation function to emerge from the strategic combination of individual atomistic services.

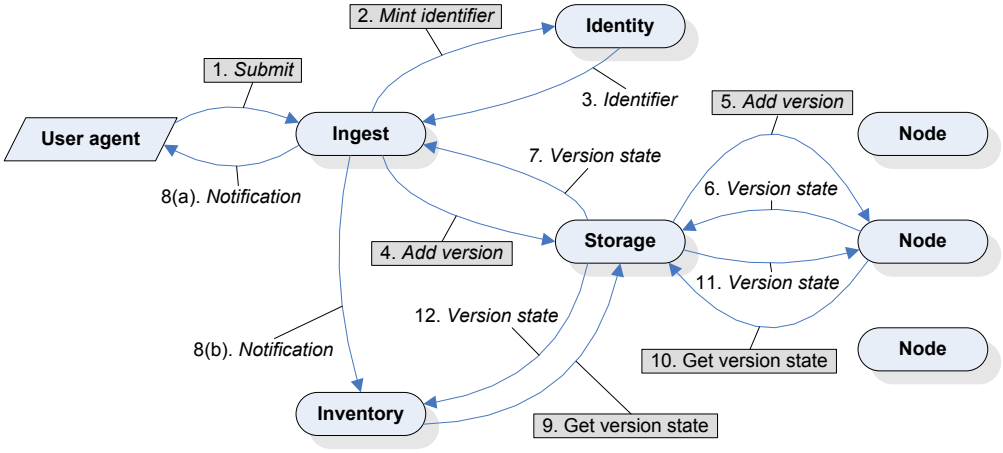

Figure 2 – Ingest workflow

The initial development milestone provided the four foundational repository services: Identity, Storage, Ingest, and Inventory. UC3 is using this infrastructure to build and operate a number of repositories tailored to the diverse needs of the University community, including significant collections of ETDs (including multimedia supplemental material), environmental science data sets, and botanical type specimens. Work for the next milestone is progressing towards implementations of Fixity, Replication, Index, Search, and Characterization. UC3 is also working with campus partners wishing to deploy decentralized micro-services-based repositories in local contexts such as data centers, academic departments, and research groups.

References

Abrams, S., Cruse, P., and Kunze, J. (2009) "Preservation is not a place," *International Journal of Digital Curation*, Vol. 4, No. 1, 8-21. Accessed March 1, 2010, from <http://www.ijdc.net/index.php/ijdc/article/view/98/73>.

Bizer, C., Cyganiak, R., and Heath, T. (2007) *How to Publish Linked Data on the Web*. Accessed March 1, 2010, from <http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial>.

Janée, G., Frew, J., and Moore, T. (2008) "Relay-supporting archives: requirements and progress," *International Journal of Digital Curation*, Vol. 4, No. 1, 57-70. Accessed March 1, 2010, from <http://www.ijdc.net/index.php/ijdc/article/view/102/77>.

Ritchie, D. (1980) "The evolution of the Unix time-sharing system," *Language Design and Programming Methodology*, in J. M. Tobias, ed., *Lecture Notes in Computer Science*, Vol. 79 (London: Springer-Verlag, 1980), 25-36. Accessed March 1, 2010, from <http://cm.bell-labs.com/cm/cs/who/dmr/hist.html#pipes>.

Wikipedia (2010a) *Murphy's law*. Accessed March 1, 2010, from <http://en.wikipedia.org/wiki/Murphy%27s_law>.

Wikipedia (2010b) *Occam's razor*. Accessed March 1, 2010, from <http://en.wikipedia.org/wiki/Occam%27s_razor>.