

Draft

Fedora Commons Technology Roadmap V0.9

Vision

Overview

Fedora Commons was incorporated in May 2007 and startup funding from the Gordon and Betty Moore Foundation was granted in July 2007. Since that time, all business functions have been created and the new organization has been staffed. Fedora Commons has been granted federal and New York status 501(c)3 status. The Fedora software was developed as a joint project of Cornell University and the University of Virginia funded by the Mellon Foundation starting in 2001. The architecture resulted from the pioneering work of Sandy Payette and Carl Lagoze later joined by Thornton Staples and Rosser Wayland. With over ten releases of the software and having developed world-wide adoption, the need to create an organization to foster and develop the Fedora and its related technologies resulted in the formation of Fedora Commons.

This document provides a roadmap for the development of Fedora Commons' technologies. It is the first such document published and the beginning of a community process for governing technology development in Fedora Commons. The roadmap summarizes the strategic vision of Fedora Commons that guides development, documents the themes and priorities of our community summarizing the needs we will address and provides a plan for the software releases we hope will enable those who adopt our technology.

Our Goals

It is now clear that much of the basis of our intellectual, organizational, scientific and cultural heritage is increasingly in digital form and can only be fully utilized in digital form. Our mission is to facilitate organizations and institutions whose own missions depend on technologies and techniques to create, manage, publish, share and preserve the world's information. Unless these organizations and institutions can sustain the world's information, provide durable access and the means to interconnect related information, regardless of the immediate economic value of the information, much of our heritage is at risk and our ability to perform important scientific and humanities research maybe greatly reduced.

- We will do our best to enable the building of information systems to provide access to durable, enduring and re-usable digital content.
- We will do our best to straddle both the Web and enterprise system needs while minimizing complexity to enable widest possible use.
- We will do our best to enable content owners to establish and enforce policies for trust and access to their content, and we will never require that they yield any rights to their content or pay any fee to anyone in order to use the software and practices we provide.

Draft

- We will do our best to enable the use of best practices for the handling and curation of the content contained within information systems incorporating our software.
- We will do our best to provide key technologies in free open source for all our goals.
- We will do our best to ensure that the software we support is modular and can be easily integrated in many configurations to support information systems that satisfy the unique requirements of our community.
- We will do our best to ensure the sustainability of the software and practices we enable, and that we will incorporate an evolutionary approach to them as a key design goal.
- We will do our best to create an ecosystem of cooperating and collaborating projects to help enable making the world's information enduring and to sustain the required infrastructure.

We believe that a holistic approach to understanding the needs of our community and providing a sustainable open community, combined with open standards and open source software, is likely to provide the best platform for long term viability of the world's information and immediate utility for its use. This does not necessarily mean open access to all information since the rights of organizations, institutions and individuals must be respected. However, there must be no proprietary technological barrier that prevents accomplishing our mission and the missions of our community. Additionally, content owners must not be expected to yield their rights simply to use software or practices we provide. While we are a non-profit corporation, however, we believe that constructive engagement with profit making organizations is necessary to reach our goals.

Many of the organizations that Fedora Commons serves have substantial overlap in their requirements for content-related solutions. This provides us with an opportunity to develop technology that satisfies common needs reducing the cost and time it takes to develop content-related information systems. We have also found that many of these organizations have unique elements to their needs or already have investments in applications or infrastructures that they must continue to use. Recent trends in software development architectures and technologies make it practical to create semi-customized solutions from re-useable components and services. Fedora Commons has selected two of the most effective trends to guide development of its technologies: Service Oriented architectures (SOA) and the Web architecture. In addition, Fedora Commons will incorporate semantic technologies and model-driven architectures as these trends become practical. Organizations or solution developers can integrate Fedora Commons-supplied components in different ways, combined with their own locally-developed components and legacy systems, into solutions that meet their unique and individual requirements.

While the approaches described above are more often associated with larger systems, we must also ensure that the barrier to entry is low so that organizations with limited resources are served. One approach we expect to use is pre-integrated solution bundles that provide reasonable utility without substantial development. Solution bundles may be used as-is or be customized within the capabilities of the organization.

Draft

It is interesting when a natural synergy forms between the needs of a community and emerging technology.

1. To ensure content is enduring, we must make the software sustainable.
2. The emergence of open source software enables communities to sustain their own software. However, the approach a community takes for sustaining software is not clear and has spawned multiple sustainment models which are still being tested. Fedora Commons is still developing its sustainment model but one requirement is clear: the need for a strong community.
3. New software architectures and technologies lend themselves to an evolutionary approach to software development, which is very appealing for sustaining software used in content-related information systems.
4. Sub-dividing the work into evolvable components makes it easier for the community to sustain the software.
5. Using continuous evolution as its guiding principle to create a sustainable software base helps achieve the mission of Fedora Commons and the organizations it serves – enabling organizations to make content enduring.

Roadmap Process

The Fedora Commons Technology Roadmap combines a description of the requirements we plan to support and the release plans for our software development projects. This first generation roadmap was prepared as a result of the first Fedora Architecture Summit held in April 2007, the formation of the Fedora Commons organization through the Moore Foundation grant, and a series of meetings held with our community including the Mellon Foundation and a number of the projects it funds. We are developing a transparent community process for authoring future versions of the roadmap, so this roadmap should be considered a living document whose next version will be prepared by the new process.

For this roadmap, we have adopted a process and terminology similar to that used by Eclipse, and adapted to suit Fedora Commons (http://www.eclipse.org/org/councils/roadmap_v3_0/index.php).

There are three main sections to the roadmap:

1. **Vision** – Information on the Fedora Commons organization and its strategic goals.
2. **Themes and Priorities** – Describes the application areas, strategic use cases, and requirements characterizing the purposes and needs which Fedora Commons is working to satisfy. This section also helps describe the scope and priority for our work.
3. **Release Plans** – Lists Fedora Commons' development projects and their work products including a timeline for their availability.

Over time the activities of Fedora Commons will grow but these activities must be prioritized by considerations of the scope in which Fedora Commons can be successful and the sustainability of the code base. The scale of our efforts is profoundly dependent on having a supportive community ecosystem contributing to the work.

Draft

This roadmap will be documented in several forms. The master and most complete form will be located on the Fedora Commons Web site. Using a Web site permits us to create a set of linked documents that the reader may explore based on interest and depth. The roadmap will also be published in a document form that is more condensed. Finally, the roadmap will be provided as an executive summary.

Themes and Priorities

Introduction

In this section we capture the needs of our community that Fedora Commons will address. These needs are expressed as a set of high level requirements found in the systems which members or organizations in our community are building or intend to build. These system requirements are captured from descriptions of our community's applications, scenarios of use (also called use cases) and experience with their existing systems, combined with imagination of innovations which we can enable. We have collected these requirements into "Themes" as an aid in understanding related sets of requirements. Themes provide a way to classify our community's needs, and may include both specific functionality and general characteristics like performance, scaling and robustness.

For the initial version of this roadmap we will use the Eclipse classification of Themes to help indicate priority. Themes are described in one of four categories (from the Eclipse 2007 Roadmap):

- Active themes are those that are ongoing and changing. From time to time, some Active themes will become Persistent and Pervasive.
- Persistent and Pervasive themes are not time or release specific. Persistent and Pervasive themes are not only a signal of importance, but permanence.
- Deferred Themes are not an indication of priority, but are an indication that there are technical or resource inhibitors preventing them from becoming an Active Theme. Deferred themes are a signal to the ecosystem that help is needed.
- Pending Themes are new and interesting themes that have not yet been properly explored and discussed to become an Active theme.

Active Themes

- **Data curation and data archives**
 - Durable digital objects
 - Preservation enabled archives
- **Re-use and interoperability**
 - Of scientific and scholarly objects
 - Enablement through standards and protocols especially via OAI-ORE
 - Repository interoperability
- **Access and Publication**
 - Integration of datasets with publications
 - Open Access

Draft

- Durable linkage, annotation and citation
- Sharing of historic scientific journals and data in support of improved scholarly/scientific communication
- **Semantic Technology**
 - Innovative uses of semantic technologies for scientific and scholarly collaboration
 - Graph-Oriented
 - Object-Triple Mapping and Query Technology
 - RDF Database (triple-store) Technology
- **Infrastructure, Integration and Deployment Technologies**
 - Transactions, Journaling, Backing Replication
 - Storage sub-system integration (transactional and special purpose, file/bitstream/blob)
 - Ease of deployment, serviceability and manageability of large scale installations
 - Middleware, Messaging and Workflow/Business Process Execution (Enterprise Readiness)
 - Repository and Middleware security
- **Ease of use**
 - Support for simple applications with low barriers to entry (solution bundles)
 - Lightweight and Web interfaces
 - Improved business object generation and persistence
 - Model-driven Content Management

Persistent and Pervasive Themes

- **Performance and Scalability**
 - Large scale collections
 - Support for dissemination caching and high speed access architectures
 - Large bitstream handling
- **Evolvability and Extensibility**
- **Accessibility Compliance (US Section 508 and equivalents)**
- **Internationalization and Localization**

Deferred Themes

- **Applications and Solutions**
 - Vertical content authoring and creation applications
 - Digital Asset Management, Media Asset Management, Web Content Management, WIKI, and Blogging
- **Application security**
- **Format identification and validation, format registries, format migration**
- **Authenticity**

Draft

Pending Themes

- **Persistent Identifiers and Alternate Identifiers**
- **Federation**
 - Object Identifier-Resolver systems
 - Update Consistency
 - Replication
 - Locking (single object and graphs)
 - User Identity and Security Management
- **High Availability and Disaster Recovery**
 - Clustering
 - Failover
- **Trusted Archive Enablement**
- **Bulk Ingest**
 - Batch Ingest
 - Pipelined Ingest
- **Innovations in new ways of exposing and accessing content**

Release Plans

Projects

Fedora Commons' software development has been divided into multiple projects, each producing core components or services. We are moving in a direction of Fedora Commons being the home for a set of inter-related open source software projects that produce components or software libraries that fit nicely into solution bundles or for integration into larger information systems. Software projects under the Fedora Commons umbrella must develop components or software libraries for use cases that are consistent with the Fedora Commons mission. Unlike Apache, which has many projects targeted at many different purposes, all Fedora Commons projects are synergistic and are intended to fit together. However, separate projects provide for better management and tracking. Fedora Commons uses software engineering methodologies and appropriate community governance mechanisms to help ensure that the components and libraries may be easily integrated. This is an evolving process as the organization and its software engineering methodologies matures.

Fedora Repository Project

This project is the traditional Fedora distribution which includes the Fedora Repository service and several closely related components and libraries. This software has proven to be an attractive open-source option for organizations building service-oriented platforms for scholarly communication, e-research, advanced digital libraries, and more. From the Web perspective, the Fedora repository service has been shown to be an effective underpinning for Web applications (including traditional Web applications, Web 2.0 style applications, and increasing Semantic Web style applications). Over time this project will be divided into smaller projects to improve integration options, simplify feature

Draft

development and facilitate management flexibility. The Fedora Repository service is a core component of future Fedora Commons' software development. The new Fedora 3.0 is currently in Beta release with production scheduled for April 2008. The current production release is Version 2.2.1. Fedora 2 will be maintained at least through January 2009 since there are a considerable number of organizations that use Fedora 2 as the basis for their production systems. There are no new features planned for Fedora 2 and development will be confined to defect repairs.

Topaz Project

Based loosely on the ORM (Object to Relational Mapping) family of software, Topaz is a powerful object to triple persistence and query service. It has its own software codebase (<http://gandalf.topazproject.org>) and integrates with both the Fedora Repository service and the Mulgara RDF Database (semantic triple-store). Major re-usable software components of Topaz include the OTM – Object Triple Mapping and OQL - Object Query Language. The Topaz Project is working with the Fedora Repository Project on re-architecture efforts that benefit both projects and bring them closer together. One result of this work in the past 6 months is the new Akubra Project, described next.

Akubra Project

The Akubra Project is a new effort developed jointly by the Fedora and Topaz project teams as the first area of joint architecture work. The goal of Akubra is to provide a pluggable storage component that supports transactions on common file systems plus the ability to support multiple customized storage options at the same time. Akubra was a result of the architecture analysis by the Fedora and Topaz developers that concluded that the best opportunity for moving forward on joint architecture was to focus on building open source components to facilitate better integration of the low-level storage of files/bitstreams/blobs, and pluggability of heterogeneous underlying storage systems. We plan to integrate the new Akubra software with both the Topaz OTM and the Fedora Repository service. The primary intent is to create better abstraction and separation of concerns between file/bitstream/blob storage, and the services that Topaz and Fedora provides over them. Once this is done, there will be more flexibility in terms of how Topaz and Fedora can evolve architecturally.

Mulgara Project

Mulgara (<http://www.mulgara.org>) is one of the premier RDF databases (semantic triple-stores) available in open-source and a key component of the Fedora Repository service and Topaz OTM. Fedora Commons has directly hired one of the lead architects of Mulgara to continue advancing the Mulgara to meeting the needs of Fedora Commons' projects. Fedora Commons and Topaz are hosting the Mulgara project resources (i.e., code repository, collaboration tools). Practical introduction of semantic technologies is important to realizing the full potential of the Fedora Commons mission and ensuring that a production-ready, supported RDF database (triple-store) is available as free, open-source software is a required component for many applications we support.

Fedora Middleware Project

This project is funded by a grant to Cornell University from the Andrew W. Mellon Foundation. The goal of this work is to provide an improved Fedora Repository service and to enable new service integrations in accordance with the principles of Service Oriented Architecture (SOA). This work is motivated by use

Draft

cases that suggest new models of scientific and scholarly communication. The Fedora Middleware Project will demonstrate reference integrations of middleware products provided in open source which are suitable for use in solutions that include services supplied by Fedora software. Middleware products for consideration and use include messaging products such as JMS providers, Enterprise Service Bus products, Choreography Engines, Business Rules Engines, Workflow/Business Process Execution Engines and Distributed Transaction Managers. No middleware products will be built as part of this project though a select group of best-of-breed integrations will be demonstrated. Both simple, lightweight approaches and enterprise-level approaches will be included as part the project. Also, approaches to management of business/mission semantics, message formats and governance issues will be considered.

Summary Release Plan

The summary release plan provides a table for each project giving a high level listing of features and the timeline for their availability. Projects are identified by the codes described in the Legend table below. Codes identify the actions for which each project is responsible: the project developing (creating) the software and projects integrating (using) the software. Usually one project will be responsible for developing the software while many projects will integrate it. These codes help clearly identify who is responsible for the work (and who gets credit for the accomplishment). Projects integrating the software play an important role in the development process by collaborating on requirements, design, critical review, lessons-learned and skill sharing. Sometimes developers must collaborate in order to create a feature resulting in development responsibility being shared between projects. However, we hope that good modularity will minimize this.

It is important to realize that Fedora Commons is a community. We have included the code "O" to mean other independent open source projects. We currently integrate and depend on the work of many open-source developers who are not formally part of Fedora Commons at this time. It is yet to be determined how their work will be incorporated or the degree in which we will collaborate with other independent projects. Regardless, we recognize the Fedora Commons cannot succeed without their works and that having a single column in the tables cannot capture the richness of their contributions. We expect this will be a fertile ground for expansion in future roadmap versions.

Legend	
Code	Description
FR	Fedora Repository Project
A	Akubra Project
T	Topaz Project
M	Mulgara Project
FM	Fedora Middleware Project
O	Other independent open source projects
D	ACTION: Develops Software
I	ACTION: Integrates Software

Draft

Our software development methodology is evolving but it primarily uses lightweight, agile methods focused on adding or improving “Features.” Each of the tables below lists the software or system Features being developed in the leftmost column. The remaining columns provide information about the responsibilities of each project (Actions) for each Feature and the expected timeframe (Availability) for its general availability. We have also included a Notes column for general information. In a future version of the roadmap we will provide a cross-reference between Features and Themes to show how each Feature contributes to Fedora Commons’ goals.

We believe using a Feature-oriented development methodology supported by a consistent roadmap is appropriate for a distributed open-source software community. We hope to create the right balance between the discipline needed to create trusted, durable content systems while facilitating the boldness needed for world-changing innovation.

Draft

Fedora Repository

Feature	Action						Available					Notes
	FR	A	T	M	FM	O	Q108	Q208	Q308	Q408	2009	
Simplified Registry	D		I				X					- Supported by CMA - Increases scalability and ingest rate
Sun ST5800 LLStore Plug-in	I					D	X	X				- Joint work with Sun - Fedora v2,v3
Content Model Architecture	D				I			X				- Includes simple reference CM language
REST API	D		I			D		X				- Contributed by MediaShelf - Experimental in v3.0
Mulgara Plug-in	I		I	D				X				- Support for latest version of Mulgara triplestore
Relationships API	D							X				- Add and remove object relationships without editing the RDF in RELS-EXT datastream
Dynamic Services	D							X				- Via CMA, provides a new way to bind services to objects - Replaces the former "Disseminator" approach
SOLr Support						D		X				- Contributed by GSearch
Atom Object Serialization	D					I		X				- Available via Ingest and Export on Fedora APIs - Serializes single Fedora digital object
ORE Object Serialization	D					I		X				- Available via Ingest and Export on Fedora APIs - Serializes single Fedora digital object
SWORD - Deposit API	D					I			X			- Expose the SWORD API to enable interoperability for digital object deposit into repository
Akubra Plug-in	I	D	I						X			- Deprecates LLStore Plug-in - JTA Backend
Advanced ORE	D								X			- Serializes network of inter-related Fedora digital objects.
Service Façade Framework Refactoring	D								X			- Uniform interface for front-end APIs - Enables repository as a JAR
JTA Compliance	D			I					X			- JTA Front end
Lightweight Batch Ingest	D			I					X			- Includes batch modification

Draft

Feature	Action						Available					Notes
	FR	A	T	M	FM	O	Q108	Q208	Q308	Q408	2009	
Object-centric API	D									X		- Permits write and side-effect operations
Replication Services	D										X	- Between Fedora repositories - Other repositories via ORE
Model-Driven Content Management	D										X	

Draft

Topaz

Feature	Action						Availability					Notes
	FR	A	T	M	FM	O	Q108	Q208	Q308	Q408	2009	
Object Triple Mapper	I		D			I	X					
Object Query Language	I		D			I	X					
Akubra Plug-in		D	I					X				
JTA Compliance	I		D			I		X				
PLoS1 App Support			D				X	X				

Draft

Akubra

Feature	Action						Availability					Notes
	FR	A	T	M	FM	O	Q108	Q208	Q308	Q408	2009	
Pluggable Storage Framework Design	I	D	I				X	X				- Fedora v3.0 for comment
Simple Transactional File System Plug-in	I	D	I					X				
Multiplexed Storage Framework Design	I	D	I					X				
Multiplexed Storage Framework	I		I						X			
Sun ST5800 Plug-in	I		I			D		X				- Joint work with Sun - Fedora v3.1
IA Petabox Plug-in	I	D	I			D			X			- Joint work with Internet Archive
Hierarchical Plug-in	I	D	I			D				X		- Needs one or more partners with HFS products
aDORe Plug-in	I	I	I			D				X		- Joint work with LANL
Advanced Sun ST5800 Plug-in	I	D	I			D				X		- Joint work with Sun - Supports registry in ST5800 - Able to delegate tasks to Storage Beans - Support for call back acknowledgements
Advanced Transactional File System Plug-in	I	D	I			D					X	- Possible joint work with Sun - Supports use of native transactional file systems

Draft

Mulgara

Feature	Action						Availability					Notes
	FR	A	T	M	FM	O	Q108	Q208	Q308	Q408	2009	
JTA Compliance	I		I	D		I		X				
Model Deconflation	I		I	D		I		X				
SPARQL Query Parser	I		I	D		I		X				
SPARQL Query Engine	I		I	D		I		X				
XA-2 Stage 1 (string pool)	I		I	D		I			X			
XA-2 Stage 2 (statement store)	I		I	D		I				X	X	
Reasoning Engine	I			D		I					X	- Planned but not funded
Modular Deployment Support	I			D		I					X	- Planned but not funded

Draft

Fedora Middleware

Feature	Action						Available					Notes
	FR	A	T	M	FM	O	Q108	Q208	Q308	Q408	2009	
Mellon ESB Study	I				D	D	X					- Group study of open source offerings for Enterprise Service Bus (informs design)
RepoMMan Support	I				D	D	X					- Informs design; possible re-use or re-distribution of components or examples - http://www.hull.ac.uk/esig/repomman/
Messaging (JMS) Integration	I				D	I		X				- Repository as publisher of Events in Atom - Services can act as subscribers
GSearch JMS Subscriber	I				D	I		X				- Search service automatically updated via JMS
Proai JMS Subscriber	I		I		D	I			X			- OAI provider service automatically updated via JMS
Lightweight Pipeline Ingest	I		I		D	I				X		- Both attached content and referenced content
Lightweight Workflow (BPE)	I		I		D	I				X		
Middleware Security	I		I		D	D				X		- Includes XACML and other technologies - Likely will use several community partners
JTA Compliance	I	I	I		D	I					X	- Requires Repository and Akubra JTA compliance - Middleware Transaction Manager
Enterprise integration: Workflow/BPE within ESB for Preservation and Archiving	I				D	D					X	- This will be driven by user demand; If lightweight approaches meet requirements, we may re-evaluate.