

Prio	Requiremen	Description	Fedora	DSpace
ity	t			
Architecture				
***	Flexible and modular System	Flexible and modular architecture that allows to include various extensions in the future that are unknown now.	Fedora consists of a Core Repository Service which exposes its functionality via Web services and a Service Framework (FSF) upon which the other services are built (Modular approach); Object-oriented data model, no predefined hierarchy; True separation between Front and Back End; -- Generic object structure from the user perspective; emphasis on genericity and meta-levels. The application as a whole looks heterogeneous and evolved over time.	A clearly designed data base builds the basis of the system, which supports thorough flexibility of metadata. The application as a whole is well designed and designed for a repository offering typical functions and GUI. It follows a monolithic approach, with a predefined data hierarchy: Repository-Community-Collection-Item-Bitstream
***		Possibility to use repository system as part of a federation	Digital objects (DO) can store content locally in the repository or content can be stored externally and only referenced by the DO	OAI-PMH is supported
***	API for storage engine	Basic operations accessible through well defined, documented and designed API. API: SOAP, REST or comparable. Basic operations: Create new, edit existing, search, query. (needed for LMS-LOR integration and implementation of special operations)	XML-Web Services; SOAP: API-A(ccess) for Consumer and API-M(angement) for Producer, both come with a WSDL; REST: API-A-LITE and API-M-LITE with somehow reduced functionality; A proper RESTful API has been proposed; For search see below	A Java API is provided for extensibility. It is not not trivial to use it, i.e. to implement a SOAP-based API for CRUD functions. A command line interface is available.
***	API for user access rights	User access rights can be assigned programmatically	Object-specific access rights are stored as datastreams and can therefore be modified via the APIs	e-People for administration and submission ADD/REMOVE/READ/WRITE rights on item level
***	API for federation functions	APIs for metadata harvesting and federated search	OAI-PMH can be served with the OAI Provider Service (via FSF)	OAI-PMH is supported
***	Metadata search with heterogeneous schemas	metadata search works with heterogeneous schemas across LOR federation and within a LOR. Applies metadata mapping or another best-effort approach.	RISearch; GSearch (Part of FSF); Arbitrary metadata collections can be added as datastreams. They can be made accessible to the Generic Search Service over XSLT transformations.	provided is DC; SIMILE project for arbitrary metadata (http://simile.mit.edu/). Not entirely clear how to deal with heterogeneous metadata
***	Full-text search	Full text search: in federation, in objects of most common formats: HTML, XML, TXT, PDF, PPT, DOC, IMS-CP, SCORM.	yes - through Lucene or Zebra search engines	LUCENE; is supported for PDF, MS Word, HTML, text documents; configurable
***	Performance	A common single-server installation is able to reasonably deal with at least 50'000 objects	ok according to fedora-commons.org	ok according to dspace.org
***	Scalability	Possibility to support up to 1Mio objetcts, if necessary with clustering etc.	ok according to fedora-commons.org	Problematic (According to evaluation report in NZ, tbc)
***	Security	No intrinsic vulnerabilities. Bullet-proof access rights system.	no obvious flaws, no negative security reports found	no obvious flaws, no negative security reports found
***	Interoperability	OAI-PMH, federated search	OAI can be served with the OAI Provider Service (via FSF)	OAI is supported

Persistent links	Possibility to add persistent link system like OAI, handle, URN	Persistent identifiers down to the datastream level. With the proposed RESTful web service this should be improved further	uses the Handle System for persistent identifiers (www.handle.net), but only to the level of of item, not the bitstreams themselves
Internationalization	Possibility to support at least EN, IT, DE, FR	utf-8 identifiers ok	resources are provided

Metadata

*** Minimal metadata schema	Possibility to define a minimal metadata schema with overall mandatory items	DC as minimal schema.	Metadata is provided to be changed and adapted. Therefore it is possible to start with a smaller set of data as well.
* Predefined sets of metadata	Possibility to pre-define common metadata schemas (IMS, MPEG7, DC, ...) that can be used when necessary	The following formats are supported: FOXML (Fedora specific, simple) and Fedora METS; In upcoming versions other formats (METS1.4 and MPEG21/DIDL) will be available.	It can be defined for a collection, which metadata is already filled in.
*** Customizable metadata schema	Institutions, groups of interest or individuals (?) can extend the minimal metadata schema according to their needs	metadata schema is extensible without restrictions	custom metadata schema restricted to a collection
** Metadata mapping for metadata search	Possibility to map metadata items between schemas. For example "contributor" of mandatory schema to "author" of IMS schema	schema mapping through XSLT transformations in generic search service	not supported
*** Unicode support		Database in UTF8; No problems were encountered	Database in UTF8; No problems were encountered
** Social tagging	Possibility to support dynamically defined social tags	not available	not available

Graphical User Interface

*** Complete standard UI	Complete standard UI available that covers all important functions for administrators and end-users	GUIs are not part of Fedora, they are separated; FEZ is a powerful Web-Interface written in PHP, but it depends on a second database where a lot of additional information gets stored; ELATED and others (http://fedora.info/tools/)	Provided; Tight integration between GUI and Backend (one product)
*** Extensible standard UI	The standard UI can be customized and extended	All the UI-projects are open source and can be adapted, but there are the usual problems when doing a fork	possible, but complex.
** Multiple standard UIs	Multiple standard UIs can be configured to run on the same repository system (needed to support multiple institutions - "Mandantenfähigkeit")	As Fedora provides its services via SOAP/REST, several different UIs can use these services;	problematic because of missing web services or other API.
*** Custom UIs	Possibility to add special purpose UIs as needed, like stripped-down query interface, specialized video portal, etc.	Possible using the Web Service interfaces	problematic because of missing web services or other API.
*** AAI authentication	Possibility to add AAI authentication system	MAMS project in AU is shibbolizing Fedora (http://www.melcoe.mq.edu.au/projects/MAMS)	- could probably be easily implemented as others have already done it, and it works fine with apache and tomcat

**	Associate copyright license	pre-defined copyright licences (like creative commons) can be easily associated	dcRights	- is provided
***	Direct distribution	objects can be directly accessed through a URL	-> persistent links	-> persistent links
**	Direct streaming	video/audio objects can be directly streamed	possible through persistent links	possible through persistent links
*	Alternative protocols for data upload	Access through https, WebDAV, (s)ftp, ...	no	no

Storage

***	Object can be of any format		ok	There exist already various kinds of repositories, which store different types of data; moreover format can be easily added.
*	Multi-part objects	One object may consist of multiple elements (a CD consists of audio tracks and scanned booklet images)	A Fedora Digital Object can consist of one or many datastreams which contain the actual content	ok
***	Access rights	Possibility to define read and write access rights on 4 levels: world, institution, self-defined group, private	Authorization module build upon Sun XACML engine plus a Policy Enforcement module; Can be used to control web services, digital objects, datastreams and disseminations; - Fedora Servlet Security filters (are these user rights?); Fez provides FezACML (as XSD)	provided on item level
***	Hierarchical organization	objects can be stored in a self-defined hierarchical structure like Institution - Domain - Department - Teacher	There are no predefined, static hierarchies; With the DO Relationships (using RDF) any kind of hierarchy can be mapped; A few common relationships are predefined in the Fedora Relationship Ontology	hierarchies are supported
**	Property and metadata inheritance	object properties (access rights) and metadata items (author name, institution) can be inherited within hierarchy. Support for pre-defined properties and possibility to override it.	not available right away. needs to be programmed in the workflow	per collection
*	Versioning system	Re-uploading an object with the same ID does not overwrite the original, but create a new version	Advanced versioning system	no
***	Large objects	support for large objects like videos of several GBytes	ok - tested with large video objects.	can be configured

Other

***	Strength of development community		Development team has acquired important funding in 2007	google summer of code
***	Strength of users community		Smaller users community. Website has been improved by relaunch in Summer 07.	large users community and active discussion forums. more running installations than fedora
***	Code quality		complex -- there are uncommented code parts	complex - well documented source code

*** Documentati
on quality

A lot of documentation is available but it is not always structured in a logical way. well documented and clear

** ease of
installation

difficult to install and set-up, because of unstructured documentation and complex design. Difficult to make precise Google-searches because of the name confusion with fedora linux distribution. very straight forward, quickly getting a running version