



Unleashing the Power of Open Source in Document Management

OPTAROS WHITE PAPER

Realize the Benefits of Open Source

Table of Content

About Optaros	3
About the Authors.....	3
Executive Summary	4
About this White Paper.....	5
What is Open Source?	5
Enterprise Content Management and Document Management.....	6
The Need for Document Management.....	7
Document Management Features	8
Create and Store	8
Organizing and Collaboration.....	8
Search & Access	9
Usability.....	10
Problems Around Document Management Faced by Enterprises.....	11
Open and Industry Standards in Document Management.....	12
Why Open Source?	13
The Open Source Document Management Landscape	15
Projects Reviewed.....	15
Alfresco.....	16
Contineo	19
Collaborative Portal Server (CPS)	21
KnowledgeTree.....	24
Owl	26
Plone	29
Open sTeam	32
Comparing Open Source Document Management Solutions.....	35
Conclusion.....	36
Getting More Information.....	37

About Optaros

Optaros is an international consulting and systems integration firm that helps enterprises solve IT business problems by providing services and solutions that maximize the benefits of open source software, open standards and global sourcing.

Bringing together experts in creating enterprise IT solutions and experts in the power of open source and open standards, Optaros plans and builds business solutions that give you better value today and increased control in the future. Applying state-of-the-art methodologies and leveraging open source communities as well near shore and off-shore development capacity, Optaros is able to deploy better solutions in less time and at fraction of the cost of traditional development.

Optaros has offices in Boston, Geneva, Zurich, Atlanta, New York and San Francisco.

ABOUT THE AUTHORS

Seth Gottlieb leads the Content Management and Collaboration Practice at Optaros, based out of the Optaros headquarters in Boston. With over 10 years of IT and technology experience in both software and professional services, Seth has helped numerous companies improve the efficiency and effectiveness of their content management and publishing processes. Seth brings expertise in technical analysis, design and implementation and project management for large scale internet and client-server applications. Seth is a member of the 2006 Board of Directors of Content Management Professionals, a membership organization of practitioners, consultants, and representatives from the software industry that fosters the sharing of content management information, practices, and strategies.

Sebastian Wohlrapp is responsible for the Content Management and Collaboration Practice at Optaros in Europe working from the Zurich office in Switzerland. He has over 8 years of experience in software development, technical team leadership, project management and delivery leadership. He implemented several content management and publishing solutions, developed proprietary CMS solutions, helped maintain a CMS product and built large scale internet and client-server applications for enterprises such as Nestlé, Novartis, the German savings banks, or Deutsche Post.

DISCLAIMER

Optaros makes no representations or warranties with respect to the contents or use of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose.



This work is licensed under a Creative Commons Attribution 2.5 License.

Executive Summary

Document management is a category of solutions that has been around for quite some time. While the focus in the eighties was on reliable and long term storage and archiving, document management software solutions became more and more a turntable for document creation, organization, access and distribution. With the emergence of Enterprise Content Management, a coexistence with Web Content Management was aspired but never really achieved. Key players in the enterprise software market are EMC Documentum, InfoText, Filenet and others. Besides the large international providers, hundreds of more local players, mostly focusing on small and medium enterprises, exist in the document management space.

Most enterprises and organizations haven't resolved their problems around storing, handling and searching documents. Files are still stored on shared file systems or – even worse - local hard disks, many companies have implemented multiple document management applications that don't interact. Searching across repositories is not possible, workflow support is often marginal. There is a clear need to act.

While the open source movement brought up hundreds of new (web) content management solutions, the landscape of open source document management is far less crowded. Document management as a topic is quite well suited to generate good open source offerings, as most companies do have a need for document management and the requirements are similar from company to company. Helpful also is the fact that "open standards" for content/document formats or access and exchange methods have been established and serve as the basis for the development of individual open source components (e.g. repository, search, transformation). Based on these standards and components, new open source solutions such as Alfresco have been developed and matured in less than a year.

Seven open source document management solutions have been compared in this white paper: Alfresco, Plone, CPS, Contineo, KnowledgeTree, Owl and Open sTeam. These applications are based on different technology platforms and each has unique features and uses. Quality and maturity of some of these solutions is more than satisfying even for enterprise usage. The final choice depends on the applicable technology strategies (e.g. Java versus Python), on the requirement priorities (e.g. collaboration versus traditional document management needs), on the need for integration with other solutions as well as on operational needs (e.g. scalability-goals, departmental versus enterprise usage, etc.) but in many situations final contenders will include Alfresco, CPS and Plone. In any case, a proof of concept or pilot involving users is recommended before making final implementation decisions.

About this White Paper

This white paper is aimed at organizations and enterprises that are considering or in the process of implementing a document management (DCM) system as well as such that are rethinking DCM and ECM practices as they refine their content management directions.

This white paper gives an introduction to the state of information chaos in many organizations, providing a view of the major factors such as legislative requirements and information management goals that drive the development of DCM solutions.

The focus of this white paper is on the leading DCM players in the open source space. We have analyzed the leading solutions, leveraging our experience in implementation as well as our understanding what today's larger organizations need and want. The goal is to give some initial help for evaluations as well as application strategies and roadmaps.

WHAT IS OPEN SOURCE?

Open source stands for software components and solutions whose source code is available, can be used, changed and distributed to other users along defined rules. These rules are defined in various license models - recommended reading in this context includes the Optaros whitepaper "Free and Open Source Licenses, Software Development, and Distribution" [1]. Thanks to the openness of the code, everybody can analyze the software and understand how the solution works. Developers can take this solution or components of it, change it or build other applications based on it. Well known open source projects/products include Linux, the Apache web server, the database MySQL or the JBoss application server.

The term "open source" was defined in 1998, however free software and sharing source code is a longstanding concept, especially in research and university environments. Today we know about more than 140'000 open source projects. These projects, whose center is usually one specific solution or component, differ in their state of development and their maturity. Thanks to license model standardization, the use, change and distribution of open source software has become easier over time and follows legally sound principles. On the basis of open source projects, new companies have been created and shaped to make money based on the success of open source technologies. Most of these companies do either distribute closed and open versions of specific software (dual licensing), offer support and maintenance services based on open source software or provide consulting, training and systems integration services around open source software.

Open source has changed and is changing the software industry for three reasons:

1. Low cost and infinitely scalable distribution of the software through the internet
2. Collaborative development process across time zones and geographies by often only loosely connected development teams

3. Transparent access to the source code, allowing other projects to use and change the code and by providing the basis for new services offerings such as support and maintenance independently of owning the intellectual property

Commercial software players have reacted to the open source movement by lowering prices, acquiring open source companies, leveraging open source projects as part of their commercial products, bundling products with open source products and by offering commercial services for open source software.

Open source has gained a strong momentum across the world. Basically every enterprise and organization is using open source in one way or another. According to a recent survey conducted by InformationWeek and Optaros [2], 87% of all enterprises surveyed acknowledged their use of open source.

Enterprise Content Management and Document Management

According to the AIIM international, the worldwide association for enterprise content management “Enterprise Content Management is the technologies used to capture, manage, store, preserve, and deliver content and documents related to organizational processes” [3].

Enterprise content management comprises the management of all types of content to be found in the enterprise. The goal here is to have one tool capable of handling any content regardless of the origin, format and use of the components.

Document management deals with documents that can occur in multiple formats such as PDF, Microsoft Office, etc. but will typically be represented by a file. After the creation of content which will typically happen outside of the document management system through a more or less integrated authoring software like an office application, the focus is on collaboration on these files, organizing the storage, access and the management of the content as well as its meta data to support all these features.

Although many open source and commercial applications that are useful for managing documents are designed for managing structured content and collaboration also, this white paper will focus strictly on document management use cases. For further reading on content management use cases we refer to our white paper “Content Management Problems and Open Source Solutions” [4].

The Need for Document Management

Despite the ever-expanding Internet and all the attention paid to web content, the information of the enterprise is largely authored and managed within document formats. The knowledge worker has grown to embrace Microsoft Office and other desktop suites that are able to give the author control over the content and layout of the artifact. The explosion of document-based content, and the lack of affordable and usable tools to manage it, has led to a crisis where most of the enterprise's content sits unmanaged on personal hard drives, shared drives and email clients where it is susceptible to arbitrary loss and unnecessary duplication. Companies today are swimming in files but thirst for accessible and reliable information.

There are a number of drivers pushing for document management solutions:

- ◆ **Collaboration and Workflow Support**

Business processes are becoming more and more complex and more and more "virtual" at the same time. To save cost, companies get rid of paper and outsource operational centers to offsite locations. Proper execution of these complex processes can only be ensured by well managed document distribution and storage processes. Collaboration of virtual teams and decentralized workforces need state-of-the-art tools to allow for efficient interaction and management.

- ◆ **Knowledge Management and Knowledge Worker support**

More and more people work as so called knowledge workers. Their product, more often than not, is some form of document. They need the right information at the right time to take good decisions and support internal and external customers in the best possible way. To be useful knowledge needs to be searchable and accessible. In many instances information and knowledge need to be reviewed and approved upon.

- ◆ **Regulatory Compliance**

Compliance today is one of the main reasons to implement document management solutions. The objective is to provide seamless documentation of business processes and transactions to auditors or to be able to do so in case of litigation or audit.

Document management solutions respond to these needs and have been established as a key component in most enterprise architectures.

Document Management Features

The broad set of requirements in the document management space results in a variety of features offered by typical DCM products. The following abstract will give you an understanding of how these features help you solve your issues around documents and the related processes as well as the functional criteria we defined for comparing the tools and measuring their feature completeness.

CREATE AND STORE

Despite the time spent creating and reading documents, most business users do not perceive themselves as content managers. Instead, they focus on the business process that they are responsible for, to which the content, and the management thereof, is simply an aspect. For this reason, most content creators and consumers are impatient with any additional, seemingly unnecessary responsibilities that a document management system would enforce. Hence, save-as and using the “attach” or “send” file functions feel like the most expedient way to share document based information. That transparency has to be balanced with the need for single source storage, good metadata and classification that will increase the reusability and accessibility of the content.

To achieve this balance, the content creator’s need for efficiency and the content consumer’s need for high quality, accessible content, a document management solution should have the ability to be tailored to work seamlessly with the business processes that it supports. Below is a set of base-line characteristics that any document management solution should have.

An author needs to be able to add or update content using **tools** and **processes that are familiar** to him and make use of his usual work environment. For example, being able to save a document from Microsoft Word right into the content repository or being able to drag files into a folder using Windows Explorer.

The system should support configurable **metadata capture and validation**. Where possible, it is preferable that the system be able to automatically extract certain metadata fields based on the properties of the document and/or where it is placed within the directory structure. Care should be taken in determining metadata requirements that capture enough information to make the content accessible and reusable yet do not overburden or frustrate content contributors.

ORGANIZING AND COLLABORATION

Most users find the “folder” metaphor an intuitive and effective way to organize content. Folders are used to store related files and also a way to group files for access control. Most document management systems have adopted the folder metaphor for these reasons. However, deep folder hierarchies can “bury” content making it difficult to find so a flattened view of the repository is also necessary. Search is the most common technique to achieve this flattened view. Some products

support the notion of **“smart” or “dynamic” folders** which are essentially collections of documents based on saved searches.

To avoid duplication, the system should be able to create a **reference to a document** in another location rather than creating a copy. Access control to both the reference and target should be applied in clear and consistent ways.

Workflow helps communicate the state of a document and coordinate collaborative participation in the documents lifecycle. For example, a workflow may ensure notification of and verification by required users that quality requirements for metadata tagging and classification are met. The distribution of relevant information or pointers to this information and the transformation into formats to support internal or external document management processes like PDF conversion can be automated by these workflows. **Versioning and archiving** optimizes the availability of the right information at the right time. **Document lifecycle management** often combines the management of classification, format, security, location and user access for a document from its creation to eventual distribution.

Notification functionality, such as email but also RSS, are useful to coordinate the efforts of multiple people (such as in a workflow) or announce the change of state or availability of a document.

As the system is rolled out to users, it is useful to have key stakeholders monitor the system for how it is being used. An advanced search feature can serve as a basic **reporting** mechanism to identify how many documents are being added to the system, what kind of documents are being added to the system or whether documents have missing or incomplete metadata. Other forms of reporting offer **dashboards** to track the status of a project across departments, and locations.

SEARCH & ACCESS

It is recommended that document management systems be **accessible** through the corporate firewall. Otherwise, sharing documents with external audiences is usually done over email which creates external, unmanaged copies that can fall out of sync from the managed original. A **web based interface** such as WebDAV or a web client is effective for this purpose.

There are a few main methods that users rely on to **find the documents** that they need: navigation, search, and bookmarking.

- ◆ **Navigation** is typically less preferred by users because it is manual and, in larger repositories, can be frustrating as users must constantly drill blindly down into directories hoping to find what they need.
- ◆ **Simple Search** is the most frequently used method to navigate a repository. Users have become accustomed to narrowing and widening their criteria to get a manageable result list (see more about searching below).
- ◆ **Advanced Search** is less frequently used but it can be indispensable to find the documents within large repositories of documents that describe a narrow set of topics. Filtering by metadata attributes such as date, author, in addition to full text can save time paging through

long lists of results. Advanced search often combines full-text searching with filtering on metadata attributes, folder location and document type. It is also used as an ad-hoc reporting tool to understand the contents of a repository such as recently added documents.

- ◆ **Bookmarks**, “Shortcuts,” or “Network Places” are important to be able to quickly return to a document or folder or to send a colleague a reference. Typically, users work on a single document or a group of documents over a period of time. Being able to go directly to that location, rather than navigate from the route of the repository every time, is a critical usability factor. Having bookmarks is also critical to being able to reference one document from another or email a reference to a document rather than the document as an attachment. Some web interfaces work with browser bookmarks better than others. If frames or designs that interfere with bookmarks are used, there needs to be some other way to generate a persistent link to an asset or folder.

All these access features should be bound to a set of **security policies**. Accounts to a document management system should be easy to establish and administer. Integration with the corporate directory (usually LDAP) allows users to manage a single set of credentials. Role and group based authorization makes it easier to apply security policies.

USABILITY

Success or failure of a document management initiative will primarily depend on how the system is configured and rolled out to the user population. Some characteristics facilitate training. Additional features to support collaboration are key to a successful and corporate-wide usage. Being able to embed customized **help content** into the application can complement training and user documentation efforts. Help is most effective if it is targeted to the business task and uses terminology that the user is familiar with.

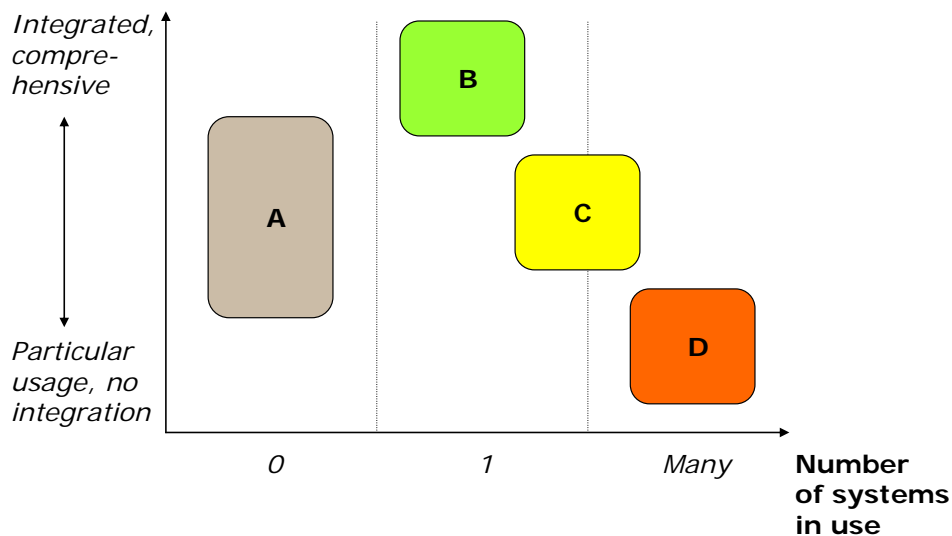
The behavior of the user interface should be consistent. For example, the procedure for adding metadata to a folder should be similar to the procedure for adding metadata to a document. Menus should be clear and use consistent terminology across the application. Depending on business requirements or regulations, compliance to web accessibility standards should also be considered.

Beyond that, usability is largely subjective. Users may be attracted to an application that shares the same terminology or metaphors that another familiar application has. Subtle differences may be magnified if users must do the same kind of function repeatedly. The best way to judge usability is to present working software to the users as soon as possible and work with them to choose the best starting point and tune the application over time.

Problems Around Document Management Faced by Enterprises

While document management technologies and solutions have been around since the late 1980s, many companies haven't solved the issues around document and knowledge creation, organization, collaboration and access. Working with many clients in this field, Optaros has often observed the following four scenarios:

Usage and integration



SCENARIO A These companies do not have a document management solution in place and typically store their documents on file shares and the employees' hard disk. This strategy quickly breaks down as the number of documents grows, the organizational structure increases in complexity, and/or regulatory factors come into play.

SCENARIO B Having implemented one document management System in a comprehensive and well integrated way, these companies have done everything what was needed. Very few companies are here, however.

SCENARIO C With the existence of two or maybe three document management solutions these companies have responded to urgent needs from business processes and collaboration need. Still much of the content is unmanaged either because these systems have not been rolled out to most users or because users have rejected the solution. Due to lack of integration and features, efficiency in usage is not optimal. Typically enterprise-wide searching is not possible. New ad hoc systems implemented for good reasons, such as Wikis or Intranets, make things worse.

SCENARIO D Most large companies are operating a whole portfolio of document management applications. Often these applications have been implemented based on an urgent business need in a specific department. Sometimes additional applications were added due to a merger or acquisition. As a result, documents are stored multiple times, cannot be searched seamlessly and there is no control of what is used and distributed.

Scenarios A, C or D are often the starting point for the implementation of a new document management system to consolidate what's there or to provide the basis for better document management tomorrow.

Open and Industry Standards in Document Management

Documents in formats such as Microsoft Word, PowerPoint, Excel, and Adobe PDF are sub-optimal from a pure document management perspective because they do not adequately separate format from structure. The organization of the document is visually represented through formatting, which requires a human eye to reliably parse. Nevertheless, these file formats are entrenched and they are the lingua franca of the "knowledge worker." Such documents also embed metadata attributes within the file, making it important to synchronize these metadata with the attributes associated with the document in the repository. For example, there might be a "title" field in the metadata and also the text of the title of the document in a bold heading on the front page of the document. If the title changes, it needs to be changed in both places. Besides these industry standards, a number of open standards have appeared and are becoming more and more the basis on which document management systems are built. The following table lists the most relevant standards:

Standard	Description	Type
MS Office	Proprietary (native) format in which MS Office documents are stored	Industry
PDF	Portable Document Format: Proprietary file format for representing documents in a device independent/resolution independent format	Industry
CIFS	Proprietary approach to access files (CIFS=Common Internet File System)	Industry
DITA	The Darwin Information Typing Architecture supports the proper construction of specialized DTDs (document type definitions) from any higher-level DTD or schema	Industry
FTP	The File Transfer Protocol is a protocol for exchanging files over any network that supports the TCP/IP protocol (such as the Internet or an intranet).	Open
XML	Extensible Markup Language is a general-purpose markup language for creating special-purpose markup languages	Open
DocBook	XML vocabulary to create documents in a presentation-neutral form capturing the logical structure of the content	Open
ODF	(OASIS) Open Document Format for saving and exchanging editable office documents	Open
WebDav	Web-based Distributed Authoring and Versioning; also HTTP extension	Open
JSR-170	Java Specification Request to define a standard API to access content repositories independently of its implementation	Open

The “open” standards such as XML are more and more influencing the way documents are stored and managed. Emerging XML standards such as DocBook and DITA are showing the direction. However, for now, the content that document management systems typically manage is metadata: structured information about the content and its classification. Metadata allow documents to be more easily organized and found which mitigates issues of redundancy and allows information to flow more freely through the organization.

The value of standards for document management is high. Standards such as JSR-170 separate the document repository from the document producers and consumers and ensure that even if the tool is changed, documents can still be accessed and exported. Standards such as XML, PDF or ODF increase the longer term viability of stored and archived documents.

Why Open Source?

There are hundreds of commercial document management systems, some of them have matured well and cover the broadest requirements. There is a great opportunity for open source alternatives in document management due to the following factors: document management is a “horizontal” solution (basically the same business requirements apply regardless of industry or size of companies); open standards are available; and commercial solutions are often expensive and quite complex.

There are a number of reasons why open source document management solutions should be considered when making decisions on what technology to use:

- ◆ Open source **reduces the cost** of implementing and owning document management solutions because there are no or only small license costs. Support and maintenance costs are often substantially lower compared to commercial alternatives. Modest requirements for hardware, operating systems, databases, etc. allow for smaller investments for the overall implementation. Eliminating license fees also eliminates the disincentive to roll out the solution to more users and achieve wider spread adoption.
- ◆ Open source inherently **avoids vendor dependencies** and lock-in. Open source projects focus on standard capabilities and allow for continuous operation even in the case the company or project behind the solution should decrease. Due to its openness, skills and development resources are often broadly available and come at a lower cost compared to proprietary alternatives.
- ◆ Open source drives for **standards based** solutions. Not only is the reference implementation of new standards in many cases an open source application (e.g. JackRabbit for JSR-170), open source projects also have a strong track record in adopting open standards quickly and adhering to them. Most open source document management applications, for example, support the Open Document Format standardized by OASIS. Organizations often have one or more content/document management solution already in place that have been deployed in a particular department to address very specific needs. As the organization moves towards

having a company-wide DCM strategy, it becomes important for any new solution to be able to interoperate with existing solutions using industry standards.

- ◆ Open source is **proven within the infrastructure layer** and provides a wealth of **reusable components** on all layers of the solution stack. Due to license restrictions, this advantage is frequently only open to solutions distributed under open source licenses. The efficiency of assembling applications from open source components was demonstrated by Alfresco. In only a few months a comprehensive document management solution was developed based on many different open source components (Jboss, Spring, Hibernate, OpenOffice, Lucene, etc.) and matured to a ready-for-production state.

There are clearly many benefits in using open source document management solutions. Besides the ones listed above, the openness of the code also allows for more flexible implementations. Document management solutions can contain features that are not for everyone and that require a certain organizational readiness. However, companies might want to add some of those features throughout a document management implementation process. Open source components are designed, developed, and tested in a highly modular way – the pieces are meant to be put together in assembly fashion. Open source projects anticipate customization, so processes like unit testing are built-in.

The Open Source Document Management Landscape

It is unlikely that the open source document management space will become as populated and as fragmented as the field of open source Web Content Management Solutions. The typical problem solved by document management systems is neither cool nor easy to perceive by the masses of developers. On the other side though, the users of document management all face similar issues and problems preparing the ground for new developments on a common ground. Document management problems are generally expensive to solve. Vendors of commercial products are known to ask for high license fees and not all of the expected benefits of the solutions result in a measurable value. These unsatisfying economic conditions, on the one hand, and the number of users interested in solutions around document management, on the other, seem good enough reasons for a couple of projects to get some traction. Remaining gaps in the existing open source solution portfolio have established additional ground for commercial open source solutions.

PROJECTS REVIEWED

	Version	Architecture	License	Link
Alfresco	1.2	Java	Commercial/MPL	http://www.alfresco.com/
Contineo	2.1 beta	Java	GPL	http://contineo.sourceforge.net/
CPS	3.4	Zope	GPL	http://www.cps-project.org/
KnowledgeTree	3.0	PHP	GPL	http://www.ktdms.com/
Owl	0.8	PHP	GPL	http://owl.sourceforge.net/
Plone	2.1.1	Zope	GPL	http://plone.org/
Open sTeam	1.6.0	Pike/MUD	GPL	http://www.open-steam.org/

On the following pages these seven solutions are analyzed and described in some detail.

ALFRESCO

Version	1.2.1
Architecture	Java
Team/Community	Alfresco Software
Support	Commercial
Link	http://www.alfresco.com

Alfresco is the newest project in the document management space but definitely not the least mature. Most of the mainly UK-based team of Alfresco Software share a Documentum or Interwoven



background and have focused on applying best practices around the management of enterprise content as well as on the architecture that fosters easy and reliable integration with other systems. Alfresco is distributed in three levels or networks: a free community network, an enterprise network, and the small business network, which is a hybrid of the two. In addition to commercial style software support, the Enterprise Network (priced on a per-user or per-CPU basis) gives access to “Shared Source” functionality, which is not available in the free version. The key differentiator is support for high-availability deployments, LDAP support, and support for group based access control (which is very important to any large scale implementation).

ARCHITECTURE

Alfresco is a Java application relying on best of breed open source components. It is based on the Spring framework and Hibernate, includes Lucene for search and indexing, utilizes some of Open Office’s document transformation capabilities and supports common ECM standards such as JSR 168 and JSR 170. The latest version (1.2) introduced support for level 2 of the JSR 170 standard allowing also for update and search in the content repository. Web services have been a well covered topic since the start of the Alfresco project. In addition, the new version offers a web services starter kit. This allows for integration also into heterogeneous or non-Java architectures without having to go the mainstream ECM vision of centralization. The Enterprise Network version of Alfresco supports both LDAP and Microsoft’s Active Directory to ensure integration into an existing security and access infrastructure. Additional features are constantly being added – the aspect-oriented approach will ensure upward compatibility. Alfresco has a distributed architecture based on the Java stack, making it very scalable and highly available with built-in fault-tolerance across multiple machines. Content is stored in the native file systems with attributes stored in a relational database.

CREATE AND STORE

Claiming to be an Enterprise Content Management (ECM) System, it is the document management that Alfresco does particularly well for now.

Alfresco is easy to use due to the multiple ways that files can be written and read from the system. The most intuitive interface is Microsoft's Common Internet File System (CIFS) protocol, which allows a Windows user to mount the Alfresco repository as a normal network file share. Through CIFS, users can open, edit, and move files just like many of us are used to operate on a shared drive without even knowing that they are using a document management system. Behind the scenes, Alfresco handles locking, versioning, security, lifecycles and metadata extraction and can execute rules such as sending emails or adding categories. The more open WebDAV protocol is also supported.

ORGANIZING AND COLLABORATION

Documents inside Alfresco are organized in work spaces following the windows folder metaphor. Access rights can be defined per folder or for every file individually. All advanced functionality and administrative control is done through the Alfresco web client. Here, users with sufficient privileges can define content rules and manage security. The web client also displays metadata and versioning information about content.

Alfresco supports the concept of "Aspects." The general idea is that an Aspect is a set of attributes or capabilities that can be assigned to an object without relying on inheritance through the class hierarchy. In Alfresco, there are aspects like "versionable" or "categorized" which can be applied to any asset within the repository. These concepts allow content types to be very simple and, if desired, users can add attributes to a single instance of a content asset. Aspects can be added to folders as rules with actions through simple wizards, similar to the way non-technical users can add rules in email. The aspect model is also extensible.

Alfresco supports "Smart Spaces." These are similar to "rooms" in collaboration tools where knowledge (an application) on how to manage a project is stored in the form of: folder structures, content lifecycles, auditing, transformations, security and drop zones.

With the current version, Alfresco supports threaded discussions on a document or folder. Projects can be managed through notifications, reporting and dashboards. Notifications can occur through email and RSS.

Documents can be made available in other formats using Alfresco's "Transformers". Retrieving a PDF version of a word document is a typical usage example.

Additional features like BPEL and extended work flow support are on the road map for future versions.

SEARCH & ACCESS

Alfresco's search is powered by the open source search engine Lucene and Open Office, which is able to extract text from many file formats and make them available to the Lucene search engine. Searches can combine content (full-text) with attributes, location (folder), document type and category/taxonomy. Alfresco has extended Lucene to support categories. Support for Microsoft Office and PDF file formats is the strongest. Alfresco can be extended to handle other file formats. Folders, or "spaces" in Alfresco terminology, can have different views or "dashboards," which are defined by creating templates using the FreeMarker templating syntax. The presentation templates as well as being used to develop dashboards can be used to create web pages, custom applications and integrations to other applications.

Security is role-based, offering inheritance between sub-folders. Access as a guest is also supported to grant, for example, general RSS access to a folder. In the Enterprise Network, roles can be granted to user groups. In the Community Network, they only can be assigned to individuals. For large deployments, the Enterprise Network is recommended.

USABILITY

The mentioned file system integration for handling documents inside Alfresco is unique and will be helpful for companies hoping widespread adoption with little or no user training. While there is the ability to automatically extract metadata from a Microsoft Office Document's properties, business processes should be put into place to ensure that the proper level of metadata is applied.

The user interface of Alfresco's web client follows the common rules of file access and management interfaces. A tree metaphor for browsing spaces would help the user to cope with large structures of files and spaces. Novice users will need to rely on a help function for some of the features. This feature hasn't yet been fully documented in the referenced mediawiki on Alfresco's homepage during the writing of this paper. The available documentation and tutorials are being extended and improved every day.

COMMUNITY AND RESSOURCES

Alfresco is a project with development of the core product originating from employees of the Alfresco Software company. Alfresco welcomes community contributions in the form of additions to the core, bug fixes, add-on modules and translations. Some contributions, such as meta-data extraction have already been integrated. The community has also translated the product into over ten different languages. But the Alfresco community is relatively new and it remains to be seen whether a development community will form around the project to contribute more

than bug reports and requirements. Forums on various topics and interest groups are available and well frequented on Alfresco's website. Documentation is another area where an active user and development community would benefit.

CONTINEO

Version	2.1 Beta
Architecture	Java
Team/Community	Small development community
Support	Community
Link	http://contineo.sourceforge.net/index.html

Contineo was started in 2003 by two German developers and has stayed a relatively small project. The latest release of the software, 2.1 (currently in Beta), is designed as a pure document management system with decent support for document management basics. Its simplicity and use of "industry standard" technologies give Contineo potential as a workgroup solution or a simple platform on which to build a custom solution.



ARCHITECTURE

Contineo is a Java-based technology and is distributed as a .war file to be deployed on any J2EE application server or servlet container. Its use of components such as the Struts web application framework, the Torque object relational mapping service, and the Apache Velocity templating engine make the Contineo architecture familiar to many Java-oriented IT departments. These components are mature and reliable but they are not considered cutting edge or trendy within the Java world. Database support for Oracle, SQL Server, Informix, and Sybase as well as open source stalwarts MySQL and PostgreSQL makes Contineo acceptable to the DBA side of the technical organization. Contineo uses the leading open source search engine Lucene.

CREATE AND STORE

Creating content within Contineo is done by uploading files through the web interface. There is no file system or WebDAV interfaces for adding or updating documents. However, convenience features somewhat compensate for this limitation. There is a feature to load an entire zipped folder and have the zip expanded within Contineo. The system also does some keyword extraction from files of recognized types and adds them to the content metadata. Other metadata attributes include source, publication date, coverage, author, type of document, keywords and version description. However, there is no validation other than checking of required fields.

ORGANIZING AND COLLABORATION

Like most document management systems, Contineo offers folder navigation and search. The keyword register, which is basically a meta data filter, is a useful way to locate documents. Contineo indexes most common office file types in its Lucene search engine. Contineo's additional feature "keyword navigation" allows a user to browse through the repository by keywords. Contineo also has a "Similar Documents" feature, which automatically searches the repository for documents containing the same keywords.

Contineo supports emailing documents and links to documents (called download tickets). Users can also post comments on documents in the repository. There is decent support for check-in/check-out and versioning with the concept of minor and major versions.

SEARCH & ACCESS

Contineo has a group-based access model where groups are granted read and write access to documents and folders. Unlike many other document systems, access rights in Contineo are not inherited from enclosing folders. Whenever a document is added to the system, the user selects the groups that will have read and write privileges to it. After initial upload, additional groups can be granted access.

The frame-based user interface makes bookmarking and copying URLs somewhat problematic. The best way to share a direct URL is through the send a download ticket feature.

USABILITY

Contineo has a spartan user interface that is easy to navigate. Most of the functionality is driven by right-click menus that are not noticeable unless the user knows that they are there.

Implementations with deep directory structures will be difficult to navigate because there is no quick way to return to commonly used files and directories. Keyword navigation and search help flatten out these hierarchies and mitigate this issue somewhat.

COMMUNITY AND RESOURCES

As mentioned earlier, the Contineo community is very small. Documentation is sparse consisting only installation instructions, Javadoc, and some online user help. Until someone steps forward and is committed to writing up documentation, the primary information resource will continue to be the online forum and mail lists where most of the questions are answered by a small core of developers.

Developers interested in extending Contineo will benefit from the abundance of documentation, professional training, and published books on the underlying technologies. For example, there are more than ten

books on the Struts project alone.

COLLABORATIVE PORTAL SERVER (CPS)

Version	3.4
Architecture	Zope
Team/Community	Nuxeo Software and Community
Support	Community/Commercial
Link	http://www.cps-project.org/

CPS (Collaborative Portal Server) is developed by French open source software company Nuxeo SAS and an external community of developers. The software is distributed under the GPL. CPS was first developed in 2003 to fill a gap between what could be done on a basic platform of Zope and the Content Management Framework (CMF) and true “Enterprise Content Management” solutions such as what was being offered in the proprietary software market. The CPS application, along with other Nuxeo products, is based on the CPS Platform which has been architected as a comprehensive ECM framework. From a functional perspective, CPS is somewhat similar to Plone. Like Plone, CPS is designed to handle documents and web based formats such as pages and news items and has a strong role based access framework. However, the CPS Platform adds a layer of infrastructure that Plone does not have.



ARCHITECTURE

CPS is a newer project than Plone and, while it lacks the benefits of Plone’s huge development and user communities (extensive documentation, active mailing lists, and an extremely polished UI), CPS has some key architectural differentiators and is more closely aligned with the cutting edge of Zope development including Zope 3. New add-on products are being constructed using the Five Project, which allows new Zope 3 features to be used on a Zope 2.x instance.

One key differentiator is the way that CPS stores content. Rather than use the Zope Object Database (ZODB) object hierarchy to organize the site, CPS separates the storage and placement of objects. Users place objects on the site by reference (called “proxies”) while the objects themselves are stored transparently behind the scenes. This architecture allows more flexibility for features like versioning, single sourcing, localization, and multiple workflow states of assets. The repository abstraction layer has potential to enable alternative storage mechanisms such as external databases.

Another key architectural differentiator is that CPS has implemented an event-driven framework to trigger functionality and integrate

components. This allows for looser coupling than standard Zope architectures, which typically depend on acquisition or inheritance to share functionality.

CPS also benefits from several infrastructure components that are provided by the CPS framework. CPSTDirectory creates a virtual directory that can be backed by disparate sources such as an LDAP directory and a relational database. CPSSchemas and CPSDocument work together to implement an MVC (Model View Controller) architectural pattern which is commonly used in Java web applications. CPSRelation manages relationships between objects in the repository.

CREATE AND STORE

The primary interface for creating file-based content within CPS is by uploading files through the web interface. CPS does not support WebDAV primarily because of its unique way of storing content in the ZODB (see architecture section). However, CPS does support Zope's ExternalEditor product, which gives the user the experience of working on a local file while the server copy is being updated with every save.

Basic metadata support is built into most content types and additional attributes can be added through CPS' CPSTypeMaker GUI. CPSTypeMaker allows fields to be defined with different validation, including what is required and valid values. CPSTypeMaker supports field level ACL and a selection of different storage engines. A content taxonomy can be applied using CPS' vocabularies framework, which allows administrators to define controlled lists of words that can be used for lookups in forms. Vocabularies can be designated for specific locations within the system, allowing for distributed controlled vocabulary management which is often necessary in large organizations.

COLLABORATION

CPS is built on top of a robust versioning system. Every time an asset is checked out, a new version, called a revision, is created within the repository. Previous revisions can be viewed (called "consulted") or reverted to through the user interface. Specific revisions of a document can be published to different sections.

The CPS workflow framework CPSWorkflow is more elaborate than DCWorkflow which comes embedded in Zope's Content Management Framework and is used by Plone. Like DCWorkflow, CPSWorkflow consists of a collection of states and transitions. However, CPSWorkflow allows assets to be assigned to a workflow based on their location (rather than just their content type) and allows multiple assets to be processed as a group. Workflow "Stacks" represent collections of tasks that can be

dynamically assigned to a workflow job. The “proxy” architecture of the CPS repository allows an asset to exist in multiple states which enables advanced workflow patterns such as branch and merge.

CPS’ event-driven architecture enables powerful subscription system where users can subscribe to be notified of various events in the system such as adding, deleting, submitting, accepting, or rejecting content.

ORGANIZING

While The CPS Framework can be modeled to achieve any strategy for organizing content, out of the box CPS is organized into Workspaces and Sections. For most content types, content is created and refined in a Workspace and then submitted for publication to one or more Sections which could be accessible to the un-authenticated or restricted based on access control rules. By default, users are given their own private workspaces. Additional workspaces can be established for teams and controlled by CPS’ role based access control framework. Sections are managed by section reviewers and managers which have permission to accept or reject content that has been submitted to their section. This allows content to be easily placed in multiple locations rather than creating duplicates of the same assets.

SEARCH & ACCESS

Like most Zope-based systems, an asset’s URL is composed of its placement within the folder hierarchy. This makes URLs readable, navigable and allows for easy bookmarking. Both Sections and Workspaces are subject to CPS’ role-based access control framework. Sections can be designated as viewable by anonymous visitors to support public access. By default, Microsoft, OpenOffice, and PDF file formats are searchable through a full-text index.

USABILITY

Users looking for a simple folder-based document management system might be a little overwhelmed when first introduced to CPS because it has so many features beyond simple document management. The Workspace/Section paradigm may not be appropriate for all business uses. However, that can be changed with some configuration.

The feature that CPS users may miss the most is the ability to browse the repository with familiar file management tools such as Windows Explorer. ExternalEditor integration will close this gap to some extent by allowing familiar editors to interface directly with the repository.

COMMUNITY AND RESOURCES

The CPS user and developer community is small but helpful. Email list traffic is moderate (at least on the English language list) but the response rate is excellent thanks to dedicated Nuxeo employees monitoring the list. Documentation and other resources are useful but not extensive. There

are thorough PDF documents for managing and using CPS. But developer documentation is somewhat thin. For development at the Zope and Python layer, CPS benefits greatly from those communities. Google search results for CPS specific problems are somewhat limited but that is made up for by the responsive mail lists.

KNOWLEDGETREE

Version	3.0
Architecture	PHP
Team/Community	JamWarehouse
Support	Community/Commercial
Link	http://ktdms.com/

KnowledgeTree is a pure-play, PHP based, documented management system developed by South African Jam Warehouse and distributed under the GPL. Originally developed for the South African Medical



Research Council, KnowledgeTree was open sourced and has enjoyed a good deal of popularity with 125,000 downloads to date. KnowledgeTree, along with Owl (described later in this whitepaper), is one of the few PHP-based DMS available. Because of its strict document management focus, the KnowledgeTree UI is simple and easy to learn. Recently KnowledgeTree has started to focus on support for compliance with Sarbanes-Oxley regulations. KnowledgeTree is frequently bundled with Jam Warehouse's proprietary product Baobab which provides WebDAV support, a client side application to navigate the repository, and toolbar plug-ins for Microsoft Office applications.

ARCHITECTURE

KnowledgeTree is built on the web scripting language PHP (4.x) and MySQL. Performance of PHP and MySQL is generally very good but the architecture does not support clustering for load balancing and failover, although this probably will not pose a problem for most organizations. High availability and scalability facilities as well as clustering are on the roadmap. WebDAV support is only available through the Jam Warehouse's proprietary product Baobab. Without Baobab, KnowledgeTree, suffers from lack of standards support making it difficult to integrate into an enterprise architecture. However, as a point solution, this may not pose a problem.

By default, document files are stored in the file system in a directory structure that maps to the folder structure shown in the UI while metadata and the search index are stored in the database. Using MySQL to hold the search index may pose a slight performance problem for very large repositories as a relational database is not optimized for this use.

The available StorageProvider plugin architecture allows for other storage locations if desired.

CREATE AND STORE

Documents added through the web interface are assigned a document type which determines what metadata fields are available and required for the asset. The document types are defined through the administrative interface. The basic install comes with a set of default document types, which can be easily extended to other types like the Dublin Core field set. File upload can be made less tedious through the use of the bulk upload feature which allows you to upload a zipped archive of files. However, all the files must be of the same type and will share the same metadata values until they are individually changed.

ORGANIZING AND COLLABORATION

KnowledgeTree's workflow model is used for document routing and can be triggered on a per-folder or per-document basis and is based on roles. Users assigned to a reviewer role are notified of items that require their attention through email and messages on the dashboard. Users are able to subscribe to folders so that they are notified of any changes.

KnowledgeTree supports check-in/check-out and document versioning. There are comments fields for the reason for checkout and a description of what edits were made. The versioning feature displays metadata differences. Earlier version can be downloaded through the user interface and used to overwrite the most recent version in order to perform a revert operation. Users can engage in threaded discussions on any asset and create references to documents in other folders within the repository.

An archiving feature allows collections of documents to be placed in an archived state but then restored later to some other location within the folder structure.

SEARCH & ACCESS

KnowledgeTree is able to full-text index most file formats including Microsoft Office, Open Office and PDF. Custom extensions can be written for other file formats. An advanced search feature allows a user to construct complex Boolean queries on most content attributes. Administrators are able to save searches and make them available to users. The repository can be browsed by document type and certain metadata values in addition to the traditional folder based navigation.

A robust role based access control (RBAC) model allows users to assign access policies to roles and groups. Permissions set on folders are inherited by subfolders and their containing documents (unless overridden). Users are organized into a hierarchical structure of organization, unit, and group and roles can be assigned to users and

levels of the organizational hierarchy. The system can be configured to authenticate users against either the user tables of KnowledgeTree's MySQL database or against an external LDAP repository or ActiveDirectory.

URLs to individual assets can be bookmarked but they are based on query strings and IDs so they do not indicate the location of the asset within the folder hierarchy.

USABILITY

Because KnowledgeTree is totally focused on document management, the user interface is fast and efficient to use. Most business users will quickly feel proficient with the UI which is intuitive and looks professional. The dashboard view gives quick access to pending tasks and checked out documents. The context-sensitive Document Actions section on the left of the screen makes it clear what options are available within the current context.

COMMUNITY AND RESOURCES

Although the recently released version 3.0 has major user interface enhancements, most of the documentation seems to be written on the 2.x release. The user forums site is also new (launched in late 2005) and has not yet to attain a critical mass where most questions can be answered by searching the archive. The KnowledgeTree community has designated moderators and the responses are generally prompt and helpful.

OWL

Version	0.8
Architecture	PHP
Team/Community	Community
Support	Community
Link	http://owl.sourceforge.net/

OWL is a totally community-driven project unlike KnowledgeTree which used OWL as a base. In general, OWL looks much less sophisticated than its PHP peer with a somewhat rough user interface and less functionality.



The user interface appears to be designed for more technical users who are comfortable with a relatively steep learning curve of becoming familiar with icons and terminology.

ARCHITECTURE

OWL runs on PHP (4.1+) and has options of MySQL, PostgreSQL, and Oracle for a database. Documents are either stored in the file system or in a MySQL database while metadata is stored in the database (except if in the case OWL-Lite which only uses the file system). There is a backup feature which ensures that a backup contains an archive of the database

and the file system in a matching state.

CREATE AND STORE

When uploading documents through the web client, users have the option of entering basic meta including title, keywords, and description. There is an option to integrate virus checking for all uploaded documents. By clicking a checkbox, entered keywords are automatically added to a remembered list to be used in later document uploads. Users can also set major and minor revision numbers. After setting the initial values the revision numbers are auto-incremented according to the editor's selection of minor or major revision. The metadata that is entered is extremely important because, as mentioned later, the search engine depends on it.

Users can upload a directory of files at a time in a Zip archive. Metadata values (including the title) will be the same for all the documents uploaded in this way. However, it is simple to go back and change these values after upload. The administrative interface has an initial load feature which allows an empty repository to be propagated with files placed on the server's file system. Files uploaded to the OWL directory structure will be added to the database when browsing the archive.

ORGANIZING AND COLLABORATION

Documents are stored in a folder hierarchy that is easy to reorganize using the move option that is available on both folders and documents. There is basic check-in/check-out functionality which locks the file from editing by others. OWL supports a comments feature that allows users to add messages about a document. Users can also "monitor" folders and documents so that they are notified whenever there is a change. A review feature allows for documents not being published before a set of defined users have approved on the documents content.

Prior versions of documents are stored and can be accessed through the user interface. The versioning system has a simple design of renaming a prior version by adding a version suffix (before to the file type extension) to the file name. For this reason, whenever a file is over-written, it must have the same file name as the original file. Files and folders deleted by users are put into a "Trash Can" that is visible to the administrator who can delete them permanently or restore them.

There is a simple but nice persistent dashboard on the user interface that presents useful statistics including how many documents have been added recently, how many documents the current user has checked out, and how many have been updated. Drill down links display the documents that are behind these numbers.

Users can email documents individually or in bulk. There is an option to

attach the file or just send a reference to the document. The attachment option is useful when emailing to individuals who do not have an account on the system.

SEARCH & ACCESS

OWL has a simplistic permission system where a document owner can select from several policy options including: "Everyone can read/download," "Only you can read/download/write," and "The selected group can read/write/download (NO DELETE)." Users can also associate a password with a file or folder to further limit access. The next version of OWL might allow for more granular security as it introduces Access Control Lists (ACL).

Users can download or preview individual documents and can also download a folder at a time as a zip archive. However, files should be updated individually.

OWL's search engine relies heavily on document metadata rather than full text which is only done for a select set of file types (PDF and MS Word). Search result ranking ascribes different weights to keyword matches in the title, file path and name, keywords, description and comments. There is no advanced search, but the basic search has options of whether or not to consider the full text of the document and whether to restrict the search to the current folder. There is also a "find related documents" feature from that finds documents with similar metadata entries to the current document.

USABILITY

Non-technical users may feel intimidated by the user interface. Most of the operations are represented by small cryptic icons that take a while to get familiar with. A quick reference card that explains what the icons mean should be kept screen-side as the user gets acclimated. The limited search could be critical limitation.

COMMUNITY AND RESOURCES

Documentation is sparse but the forum is generally active and responsive. Being a totally community driven project, development on OWL goes in fits and starts. The community is dynamic. For example, recently an external developer created an alternative user interface for OWL and made it freely available (after attempting to charge \$10 per download and then being informed of OWL's GPL licensing restrictions).

PLONE

Version	2.1.1
Architecture	Zope
Team/Community	Community
Support	Community/Commercial/Consulting
Link	http://plone.org/

Plone is the most popular of the Zope-based content management systems and enjoys a large community of developers and users.



While Plone is used for many purposes (everything from commercial identity websites to team workspaces) it has decent capabilities as a document management system. The one key limitation of Plone as a pure document management system is its weakness in versioning. For many use cases, this limitation may rule out Plone. However, in cases where versioning is not critical, Plone's other strengths may compensate. The Plone community puts a great deal of emphasis on the design of the user interface. Most new releases have features that enhance usability. Still, as a pure document management system, Plone might seem too feature rich and complicated. Unless these other features will be leveraged, a simpler solution that has versioning might be preferable.

ARCHITECTURE

Plone is built on the Zope Content Management Framework (CMF) platform, which provides many of the core document features such as storage, access control, and workflow. The Plone layer provides mostly user interface oriented functionality and a framework for integrating additional add-on modules. All content is stored in the ZODB unless an add-on component such as ExternalStorage is used. Otherwise, large files tend to bloat the ZODB and make it unnecessarily large. While the ZODB is relatively closed to external applications, Zope supports several ways to access content through the application layer: WebDAV, FTP, and the ability to implement XML-RPC based APIs.

Following the latest trend in Plone development, they are starting to leverage new architectural concepts introduced in Zope 3 - a total re-write of the Version 2 which Plone was written on. The Five Project allows Zope 3's new features to be available on a Zope 2 server and several Plone add-ons are starting to take advantage of their availability.

CREATE AND STORE

Content can be added to a Plone site via the web client, FTP, or WebDAV. Plone comes with a component called ExternalEditor which allows any locally installed editor (MS Word, Adobe InDesign, etc.) to send updates to an asset in Plone on every save. However, this feature requires the installation of the Python runtime and a small program on the workstation. Enfold Systems (employer of Plone co-founder Alan Runyan

and several other key Plone developers) also produces a commercial product called Plone Desktop, which gives a file explorer-like interface to a Plone server and enables features like drag and drop and access to Plone metadata. The Microsoft Office Suite's strong support for WebDAV makes it easy to open and update a document directly through WebDAV.

By default, there is strong metadata support and new metadata attributes and validation rules can easily be added. Thanks to Zope's acquisition model, which causes assets to acquire information and capabilities of parent folders, metadata assigned at higher levels can trickle down reducing the need for metadata entry and access control assignment.

Versioning is not supported by the core Plone application and it is not projected to be included in any new releases in the immediate future. Right now, the best alternative is an add-on product called CMFEditions, which handles the basics adequately but has a limited user interface that requires the users to manually create a new version rather than automatically create one every time a new version is uploaded into the system. CMFEditions is still in Alpha and not recommended for production deployments although several production systems do use it.

ORGANIZING AND COLLABORATION

While Plone content is organized in folders, Plone allows the creation of query based collections called "Smart Folders." Smart folders create the opportunity for assets to appear in multiple places on the site and eliminating the need for redundancy. Add-on products can also utilize Smart Folders to deliver functionality such as rule based RSS feeds. Direct references can be achieved through the "link" content type.

Because of its roots as a web content management system, Plone has a robust workflow system that manages an asset's state and can trigger events when the state changes. Out of the box, Plone is set up with states for private (where only the owner can view it), public draft (where only authenticated users can see it) and public (readable to anonymous users). States and visibility rules can be configured through the Zope Management Interface (ZMI).

Plone has a commenting feature that allows users to post notes about a document in the system. Commenting can be globally enabled/disabled and overridden at the asset level. Plone's lack of a strong event model makes adding functionality like automatically emailing an individual or group when an asset is added to a folder difficult to do. However, workflow transitions can trigger this kind of logic.

SEARCH & ACCESS

Zope's folder based architecture ensures that all assets within a Plone

sites have readable and navigable URLs that reflect the organization of the site. With the addition of a couple of add-on modules, Plone can full text index most popular file types including all of the Microsoft Office Suite, Open Office, and PDF. With the latest version, Plone added an AJAX based LiveSearch feature which is similar to Google Suggest. WebDAV and FTP support make the Plone repository through familiar tools such as Windows Explorer.

Access is controlled through Plone's role based security model, which can be applied at the individual document or folder level. Acquisition allows access control policies to trickle down to sub folders unless overridden. Plone comes with roles for Owner, Manager, Reviewer, and Member and more can be added through the administrative interface. Users can share folders and content directly with other users and groups.

The Plone project is committed to supporting accessibility standards and passes ADA Section 508 and the W3C AA standards. Plone also is valid against the XHTML 1.0 specification.

USABILITY

Most users find the out-of-the-box Plone interface attractive and intuitive. The Plone interface works elegantly with multiple content types (documents, pages, news items, etc.) but also works for just documents (or files in Plone's terminology).

The new LiveSearch feature is helpful and there are plans to add more AJAX based functionality (as long as they do not violate accessibility standards) in future releases.

COMMUNITY AND RESSOURCES

The Plone community is extremely large and prolific in the development of the core application, add-on modules, and documentation. A couple of times a year, the Plone community organizes "sprints" where Plone developers from all over the world convene for an intensive 3-5 day programming effort to add new functionality. There are several excellent books on using and extending Plone and the mailing lists get heavy traffic with questions and answers.

OPEN STEAM

Version	1.6.0
Architecture	MUD/Pike
Team/Community	German universities (mainly University of Paderborn), community
Support	Community
Link	http://www.open-steam.org/

Open sTeam is an infrastructure for computer-based cooperative learning. The project was started in 2000 and was supported by Germany's National Research and Education Network (DFN) until 2003. Today it is mainly run by the department "Computers and Society" at the Heinz Nixdorf Institute of the University of Paderborn in Germany. The initial idea was to develop an innovative system architecture for structuring information through an open source development community. Today open sTeam is part of the Debian Linux distribution. It is used to support eLearning, content management and knowledge management initiatives.

**ARCHITECTURE**

Open sTeam was developed in Pike, a dynamic programming language with a syntax similar to Java and C aiming to be "the first scripting language for the Semantic Web" [5].

The open sTeam server is based on MUD (Multi User Dimension) technology, similar to what can be found in multi-player computer games. But the event-driven environment is based on a subscriber/publisher paradigm rather than on callbacks like in most gaming-oriented MUDs.

The sTeam infrastructure is open to integration into heterogeneous environments. The Pike server supports numerous protocols like http(s), ftp, irc, smtp, imap and jabber. APIs for Java, C++ and PHP ensure easy integration of external services as well as the integration of the sTeam system into existing infrastructures.

sTeam objects are stored in a MySQL database accessed through a persistence layer. Using MySQL to hold the search index may pose a slight performance problem for very large repositories as a relational database is not optimized for this use. An object-relational database like PostgreSQL could help support the systems architecture without have to do a relational mapping.

CREATE AND STORE

Uploading of documents is handled through the web client, through FTP or WebDAV or by sending an email to the server. The uploaded documents are organized in knowledge areas and virtual rooms in which the collaboration and content advancement is organized. sTeam allows for the

creation of links to external object. E.g. you can add a PDF to the sTeam repository which is located on a server somewhere on the internet.

Metadata is not extracted from documents but so called keywords can be defined. Additionally one can add annotations to the uploaded assets. Annotations can be plain text as well as HTML and file attachments. The user is free to choose an icon to better distinguish different documents within rooms as well as a publication path and a style sheet used for inline display of the document.

ORGANIZING AND COLLABORATION

The virtual rooms represent central meeting points within areas. Rooms can be connected through gates. Folders help to further structure content below this level. Within those entities users can collaborate by maintaining a document base as well as chatting through an integrated Java chat tool or communicate through freely definable message boards. A calendar module supports the organization of schedules; a shared whiteboard is available for visual online communication.

Check in/out is not available but a function called "rucksack" allows users to take documents or copies of documents with them to other rooms or areas. Changes to documents result in a new version. Versions can be compared and restored.

Areas and containers can be monitored. Users will receive a message in case of certain modifications. This is particularly useful for areas and containers containing a message board. As soon as a new contribution is entered, users get notified through their sTeam mailbox or if desired by email.

SEARCH & ACCESS

Access rights can be defined per user and group for read, write, execute, move, insert, annotate and sanction. An object may acquire its access rights from its environment. If acquisition is activated, the access rights for a particular user or group are composed from those explicitly set for this object and from those of the environment. By default, acquisition is activated in order to simplify the rights management. The new version 2.0 (still beta) will allow for LDAP integration.

Workspaces can be accessed via ftp which allows for easy browsing and bulk download. Objects may be published in order to be freely accessible. A virtual path for publication may be specified at which the object will be published. Anybody may then access the object by specifying the server name, port, and virtual path in their browser. By exporting, the area or container including all or selected contents may be downloaded. The server creates a file with that content in compressed form (in tar format)

and presents it for download.

Search allows filtering and full-text search covering the documents inline content, metadata and annotations. The search in the demo version was rather slow.

USABILITY

The metaphors used for the sTeam processes as well as the user interface are easy to understand and use. Some of them are not common for document management environments, features like the rucksack or gates have commoditized pendants in the document management and collaboration space but novices to this topic in general will still benefit from easy to understand functions.

The context-sensitive help has a well elaborated content.

COMMUNITY AND RESSOURCES

The usage and community around sTeam is mainly in German universities. Around 20 diploma theses have been written around this project dealing with various topics around organization of distributed learning environments. The results of this scientific work have all contributed to the feature set of sTeam in one or the other way. The website offers a lot of documentation and a well-maintained demo of the software.

Comparing Open Source Document Management Solutions

Comparing solutions with different objectives, different backgrounds and roots is difficult. The following table takes an “**Enterprise perspective**”. The relevance of individual criteria depends on the specific needs in a given situation.

	Strengths/Weaknesses	Standards Support	Community	Resources available
Alfresco	<ul style="list-style-type: none"> + Strong versioning + Standards support + Content rules - Lack of Documentation 	<ul style="list-style-type: none"> CIFS WebDAV FTP JCR LDAP (commercial only) 	The Alfresco project has only grown over a good year yet, so the community is small. Alfresco Inc. has the resources to provide the support until a strong community develops.	Documentation is available through the Alfresco website.
Contineo	<ul style="list-style-type: none"> + Simplicity + Familiar Java technologies - Lack of Documentation - Small community 		Very small development community consisting of only a few programmers.	Very little documentation is available. The only source of information is the SourceForge forum and mail list.
CPS	<ul style="list-style-type: none"> + Strong versioning + Strong workflow - Lack of WebDAV support - Closed ZODB repository 	LDAP	Nuxeo drives development of CPS and monitors mailing lists to ensure that questions are answered.	Documentation is useful. The CPS mailing list is very responsive and helpful.
KnowledgeTree	<ul style="list-style-type: none"> + Simplicity - Lack of standards support (free version only) 	LDAP	Jam Warehouse is responsible for most of the development but there is a growing user community.	Documentation is useful but has not been updated to the new 3.0 release.
Owl	<ul style="list-style-type: none"> + Simplicity - Unattractive UI - Weak search - Lack of standards support 		The Owl community is relatively active for a small project. There are several developers integrating Owl into other applications.	Documentation is extremely thin.
Plone	<ul style="list-style-type: none"> + Nice UI + Good search - Poor versioning - Closed ZODB 	<ul style="list-style-type: none"> WebDAV FTP Section 508 W3C AA LDAP 	<ul style="list-style-type: none"> Thriving user and developer community Active mailing lists and blogs 	Excellent user community contributed documentation and several professionally published books.
Open sTeam	<ul style="list-style-type: none"> + Strong collaboration support - Closed DB - Weak search 	<ul style="list-style-type: none"> WebDAV FTP Java and C++ API 	Community and usage mainly around German universities.	Well documented project. Several user and technical manuals are available in addition to an online demo.

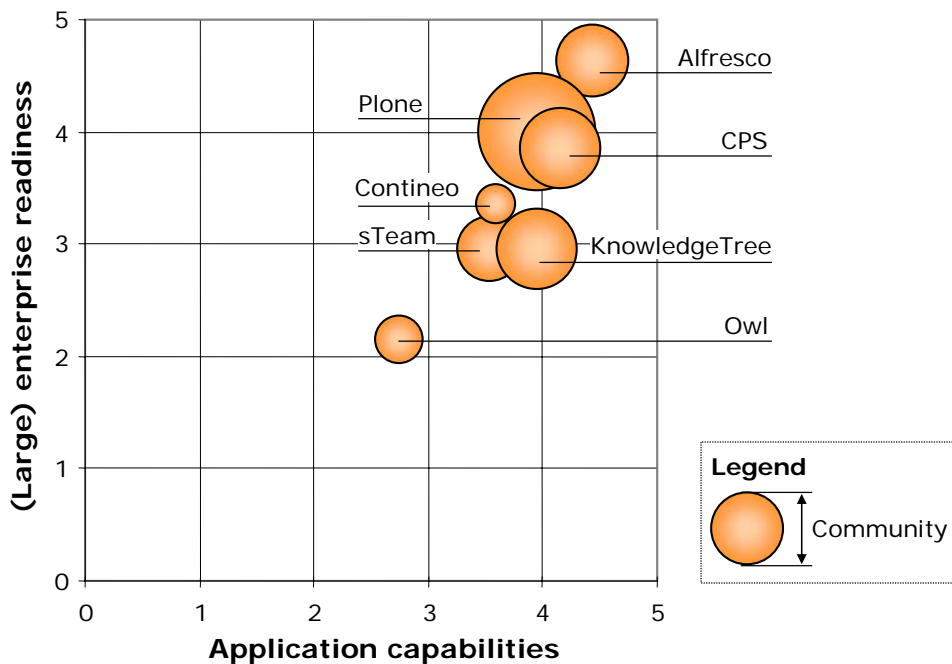
Conclusion

All the solutions documented in this white paper have unique features, strengths and also weaknesses. To help companies defining a first short list, the following chart compares existing open source document management solutions along **three axes**.

“Application capabilities” is measuring the availability of critical functions and features, the number of supported relevant document types, ease of configuration and adaptation to specific needs, the availability of workflow support and transformations as well as interfaces and APIs.

With **“Enterprise readiness”** architectures and design patterns are ranked, technologies assessed and mapped against typical enterprise requirements. Integration is key for larger organizations; therefore, the support of open standards and commonly-used interface standards is key here, too. Important is also the availability of professional release management, support and documentation.

The **“community”** is an important indicator for the level of adoption and the assistance and the access to existing experiences you get when implementing the system. It can also be regarded as insurance for future development and support. It is represented by the size of the bubbles below.



Depending on the situation, different solutions might be the right choice. Often the existing platforms around the document management solution will influence the selection of a new technology in a dominant way. Companies with a clear Java policy will most probably go for Alfresco or Contineo, for example.

Doing a proof-of-concept as part of the evaluation is strongly recommended. Open source makes these additional tests easy as no contracts and signatures are usually needed to download the software.

Getting More Information

The following resources provide excellent sources of additional information on some of the topics addressed in this paper.

- [1] **"Free and Open Source Licenses, Software Development, and Distribution"**
by Stephen Walli, VP open source Development Strategy at Optaros, Inc.
- [2] **"The Growth and Value of Open Source in Enterprises and Organizations"**
Optaros White Paper, by Stephen Walli, Bruno von Rotz and Dave Gynn
- [3] **"About Enterprise Content Management (ECM)"**
<http://www.aiim.org/about-ecm.asp>
- [4] **"Content Management Problems and Open Source Solutions"**
by Seth Gottlieb, Content Management Practice Lead at Optaros, Inc.
- [5] **"pike – first for the semantic web"**
<http://pike.ida.liu.se/>

JOBS

As we grow, we are adding technical consultants, marketing team members, project management, and sales to our team. To apply, please email your résumé to work-us@optaros.com for employment in the US and work-europe@optaros.com for employment in Europe including the job title in the subject line.

CONTACTS

Seth Gottlieb

sgottlieb@optaros.com

contentthere.blogspot.com

Sebastian Wohlrapp

swohrapp@optaros.com

blog.wohrapp.com

Brian Otis

botis@optaros.com

60 Canal Street; 4th Floor

Boston, MA 02114

+1 617 227 1855

Heinz Rudin

hрудin@optaros.com

Weinbergstrasse 147

CH-8006 **Zurich**

+41 (0) 44 362 11 20

Kay Flieger

kflieger@optaros.com

12 , Avenue des Morgines

CH-1213 Petit-Lancy, **Geneva**

+41 (0) 44 362 11 19