

A Peer-to-Peer Approach to Content Dissemination and Search in Collaborative Networks

Ismail Bhana and David Johnson

Advanced Computing and Emerging Technologies Centre,
School of Systems Engineering,
The University of Reading,
Reading, RG6 6AY, United Kingdom

Abstract. There are three key driving forces behind the development of Internet Content Management Systems (CMS) a desire to manage the explosion of content, a desire to provide structure and meaning to content in order to make it accessible, and a desire to work collaboratively to manipulate content in some meaningful way. Yet the traditional CMS has been unable to meet the latter of these requirements, often failing to provide sufficient tools for collaboration in a distributed context. Peer-to-Peer (P2P) systems are networks in which every node is an equal participant (whether transmitting data, exchanging content, or invoking services) and there is an absence of any centralised administrative or coordinating authorities. P2P systems are inherently more scalable than equivalent client-server implementations as they tend to use resources at the edge of the network much more effectively. This paper details the rationale and design of a P2P middleware for collaborative content management.

1 Introduction

There are three key driving forces behind the development of Internet Content Management Systems (CMS) a desire to manage the explosion of information (or content), a desire to provide structure and meaning to content in order to make it accessible, and a desire to work collaboratively to manipulate content in some meaningful way. Yet the traditional CMS has been unable to meet the latter of these requirements, often failing to provide sufficient tools for collaboration in a distributed context. The distributed CMS addresses the need to delegate control of resources and serves as a more natural paradigm for the collaboration in the CMS. However, with the burgeoning mobile market and an increasing need to support a range of end-user devices for content authoring, sharing, and manipulation has lead to a new requirement for meaningful collaborative tools that are able to deal with the complexity and heterogeneity in the network.

Most of current popular open source and commercial CMS implementations (e.g. Zope [1], Cocoon [2], and Magnolia [3]) are based on the client-server model.

The client-server model has many obvious advantages in terms of familiarity (amongst developers, administrators and users), ease of deployment and administration, simplified version control and archiving, manageability in access control, security and data consistency. However, the relative lack of complexity in these systems results in a number of limitations in scalability and reliability, particularly where there is a rapidly fluctuating user base or changing network, as is common in mobile networks. The client-server model is essentially static and does not scale well as the number of clients increases, both because of limitations on the server and limitations in bandwidth around a heavily loaded server (the congestion zone). Server clusters, load balancing, and edge caches (as used in Zope) lessen the problem in some circumstances but are a costly solution and cannot overcome the problem entirely.

In contrast, the P2P approach restores an element of balance to the network. Firstly, whilst servers are still a central element of the network there is no steadfast reliance on a particular set of central provides. P2P systems are thus much more scalable. P2P systems are also in many circumstances much more fault-tolerant (i.e. resources and services have high availability) due to the potential for replication redundancy (for resources that are replicated amongst peers). Moreover, P2P systems can be more efficient in bandwidth utilisation because they tend to spread the load of network traffic more evenly over the network.

These properties are highly significant in relation to the design of a collaborative CMS, particularly in a heterogeneous context (i.e. spanning operating system, network, and mobile boundaries). However, due to increased complexity the P2P approach also presents us with a number of challenges particularly in ensuring consistency, security, access control and accountability. The JXTA CMS [4] the Edutella project [5], and the Hausheer and Stiller approach [6] are attempts to tackle the content problem from a P2P perspective.

Building on the traditional strengths of the CMS, the possible additional elements of fault tolerance, availability, flexibility and a sufficient set of collaborative tools are critical in ensuring the future success of the CMS. The following sections of this paper give details of the rationale and design of a P2P middleware for mobile and adhoc collaborative computing (known as Coco) that includes services to support collaborative content management.

2 Our Approach

Our goal, as described in [7], is to develop a framework that supports collaboration in a way that enables users to self-organise and communicate, share tasks, workloads, and content, and interact across multiple different computing platforms. The rationale for designing a collaborative content system on P2P networks is based on the desire to achieve scalability, enabling a collaborative system to scale with a dynamically changing user base, and resilience. Our goal is also to support self-organisation and dynamic behaviour by developing systems and services that support the organisation of individuals into groups with

shared interests and allowing the formation of dynamic collaborations. As a starting point, our model builds on the general CMS lifecycle depicted in figure 1. This model places collaboration at the heart of content management. The figure illustrates that content management is a continual process of creation, collaboration, and dissemination.

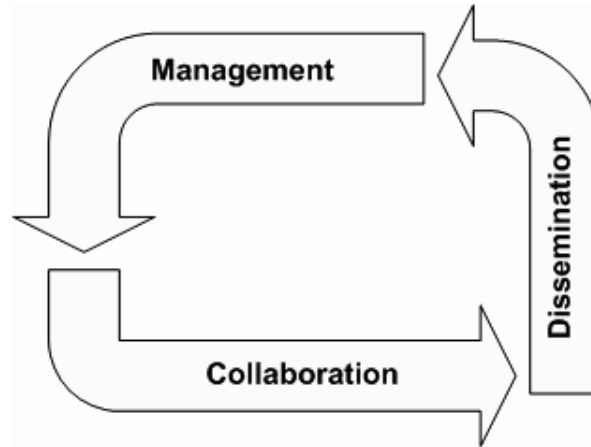


Fig. 1. The CMS lifecycle: a process of content creation, collaboration, and dissemination

The Coco content network can be viewed as a hybrid P2P system built above a pure P2P JXTA network. The network consists of self-regulating regions called provider networks that will typically represent some sort of real world enterprise, such as a university, local education authority, company, or organisation, but may be any arbitrary collection of peers with a shared interest. The provider networks act as a trusted region (or a secure domain of trust). Peers are not limited to the same network, they may be geographically dispersed, or behind firewalls or routers, or may be mobile devices, as illustrated by figure 2.

Whilst peers within provider networks are able to interact freely in a pure P2P manner, each provider network consists of at least one peer (the portal peer) that acts as a gateway and (Web) portal into the provider network for external peers. It is this peer that enables internal peers to interact with external peers and assists in the process of authentication and access control. The portal peer is also able to act as a Web proxy to the P2P network residing within the institution, enabling users to search and retrieve content over the Web (using the company or university website, for instance) without requiring them to install the relevant P2P software. This content is live, meaning that the state of the network is continually changing as peers appear and disappear at will. The system also enables agreements to be formed between provider networks supporting (in future) logging and reporting. For networks to communicate in such

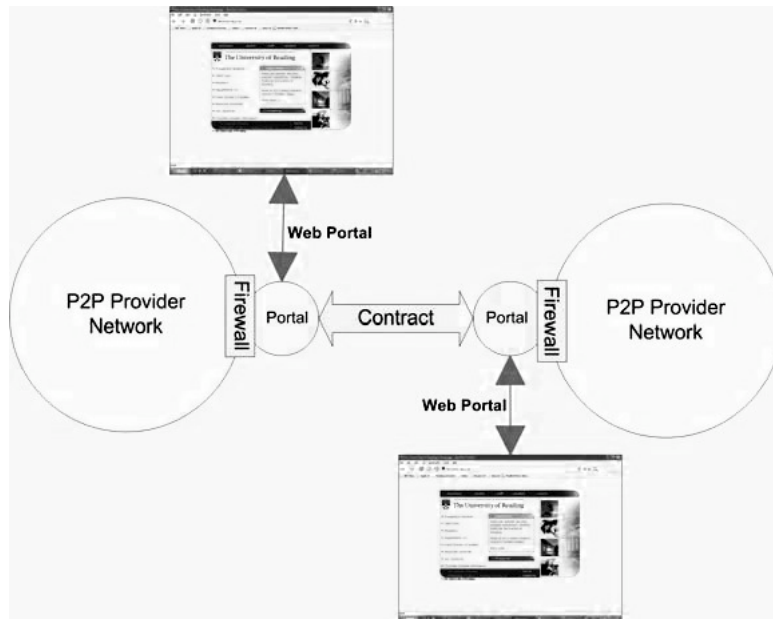


Fig. 2. Accessing Live P2P Content; Provider Networks that will typically represent some sort of real world enterprise, such as a university, local education authority, company, or organisation, but may be any arbitrary collection of peers with a shared interest

a way it is important to define a set of interoperability standards standard ways of representing resources as well as invoking operations on remote peers in the network.

2.1 Content as a Resource

Content, in this context, is defined as the set of resources available for consumption within the network. This definition ranges from the obvious, such as files and documents, to the less intuitive, such as services and computing resources, to things that do not generally have an opaque representation, such people (represented, for example, using VCards [8]). This formulation has much in common with the ethos of the Resource Description Format (RDF) [9] and it is, in fact, RDF that is used in our implementation as the language (rather than framework, as RDF is essentially Web-based) of resource description.

In order for non-Web resources to be described in a P2P context they are represented using a unique Universal Resource Name (URN). The content system uses an URN notation to form a unique content identifier for each unit of content generated using a cryptographic digest. Whilst it is normal to uniquely reference an RDF resource using a URL, there may be many situations in a replicated P2P network in which a given resource is duplicated across many network nodes or

devices and hence a location-dependent representation is inadequate. The given representation allows a resource to be referenced without an a priori knowledge of its location within the network.

Metadata describing content is cached by peers in the network to ensure high availability and performance of search queries. Each peer is responsible for managing its cache and stale resources may be purged at regular intervals.

2.2 Service Invocation

The CMS is deployed as a P2P Web service using open protocols such as WSDL [10] and SOAP [11]. Search queries are submitted using Web service invocations (although an API is required to deal with P2P interactions, including dynamically discovering peers and peer services). Using open standards such as SOAP provides us with enormous flexibility as it abstracts away the service interfaces from the underlying transport or middleware. Content can therefore be searched and retrieved over the JXTA network as well as over the Web as a standard Web service where the individual peer acts as a web server and is able to tunnel requests through the P2P network (this is essentially what the portal peer does). As figure 3 illustrates the invocation process consists of three steps:

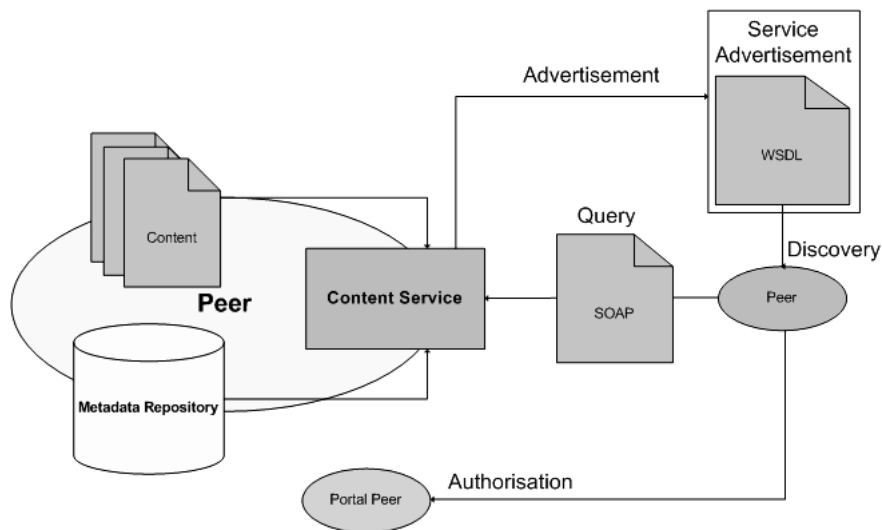


Fig. 3. Invocation of a P2P Web Service: each peer acts as a web server and is able to propagate Web service invocations through the P2P network

- Service Advertisement & Discovery - the service descriptor (WSDL) for the peer hosting an instance of the content service is propagated to peers using the JXTA discovery mechanism. This enables peers to dynamically discover new services as they appear on the network.

- Authentication & Authorisation - the next step (if applicable) is for the consumer peer to authenticate with the relevant authority that will grant access to the peer. This process is optional and allows a peer to delegate the authorisation process as might be desirable in an enterprise or educational context. Fail-over mechanisms will be in place in future if the portal peer is temporarily unavailable. This step may also be used to add additional support for logging, versioning, or charging for the service provided by a particular peer within a provider network.
- Invocation - once the peer has the relevant service description and authorisation it is able to query the service-hosting peer directly. Search queries are normally answered with RDF containing information about resources available on the peer, as well as cached information about resources available on other peers (if specified in the request).

An XML metadata repository (using Xindice [12]) is used to store and retrieve resource entries. The advantage of using an open source XML database is that we don't need to worry about storage and retrieval issues and developing database optimisations. RDF resources are simply added to database and can be retrieved later using XPath [13]. Metadata is normally stored using some form of formal schema, such as Dublin Core [14].

2.3 Enabling Mobile Devices

In the mobile arena, we are building on the Java 2 Platform Micro Edition (J2ME) [15] and JXTA for J2ME (JXME) [16] to allow mobile devices, such as phones and PDAs, to participate in the content transactions. Mobile devices have significant hardware constraints compared to desktop and enterprise machines. These limitations include:

- Significantly less processing power
- Limited runtime memory
- Little or no persistent memory
- Very small screens with limited modes of interaction
- Lower network bandwidth and higher network latencies

By basing software for mobile devices on the J2ME platform, the range of device capabilities in the mobile computing market is accounted for through the use of different J2ME device configurations and profiles. A configuration defines the features of a Java Virtual Machine (JVM) that a device can use, and a profile is the definition of the set of Java APIs that are available to the developer for that class of device. JXME has been implemented for the Connected, Limited Device Configuration [17] and the Mobile Information Device Profile [18] (CLDC/MIDP) that is also the most widely used configuration and profile combination.

Coco for mobile devices (MicroCoco) is being built on J2ME and JXME. MicroCoco includes services to consume the collaborative content management services provided by Coco. Services that require searching textual data

are ideally suited to J2ME based applications, because the restrictive modes of interaction imposed by mobile computing devices are tailored to text input and J2ME accounts for this limitation in its standard interface components. The mobile device peers will not function as content providers, but only consumers because even though it is possible for devices such as PDAs and mobile phones to host and share content, it is highly unlikely that a user will have a mobile device as their primary computing device. The amount of persistent memory is limited in comparison to that of a desktop machine and we have assumed that users will not want to keep a great number of documents on the mobile device. Many mobile devices also do not have the appropriate third party software to view documents (such as Adobe PDF or Microsoft Word files).

However, a user may wish to search for and record their search results whilst on the move. By having a mobile application that can search the content network for resources, users are given the facility to participate in research whilst on the move. Search results can be stored locally on the mobile device. To facilitate sending search results to a desktop peer that a user may also be running, the mobile peer can be linked with a desktop peer in a similar manner in which Bluetooth devices are paired. The user can then send the search results from the mobile device to a desktop machine where the user can then download the documents, all of which occurs in a purely P2P manner.

3 Conclusion

Our experience indicates that the decentralised (P2P) approach works very well for content distribution. Our rationale for designing a collaborative content system on P2P networks was out of a desire to achieve scalability, as well as to enable a diverse range of devices to participate in collaborative processes. We wanted to provide a framework that supports the interaction of groups or individuals with shared interests. The difficulty in taking the P2P approach is that there is an inevitable increase in the design complexity of a CMS and it makes it difficult to achieve many of the things that traditional CMSs do well. For instance, for version control and archiving, strong consistency is required to ensure that all elements of a version history are always accessible. Workflow is another area that can be complicated with a decentralised model it requires flexible organisational models that can be easily customised, which in turn rely on security and access control mechanisms. Logging and reporting is another key area where flexible mechanisms must be in place to facilitate accountability. Our intention in the near future is to take the development of the Coco content service forward through a series of alpha and beta releases. In future, we intend to make developments in the areas of pricing/charging, replication and versioning, logging, authentication and access control, privacy and accountability, and security.

References

1. Zope: Open Source Application Server for Content Management Systems; Version 2.7.4, <http://www.zope.org/Products/>, (2005)
2. Cocoon: XML publishing framework; Version 2.0.4, Apache Software Foundation, <http://xml.apache.org/cocoon/>, (2005)
3. Magnolia: Open Source Content Management; www.oscom.org/matrix/magnolia.html, 2005.
4. Project JXTA, CMS; <http://cms.jxta.org/servlets/ProjectHome>, (2005)
5. EDUTELLA: A P2P Networking Infrastructure Based on RDF; <http://edutella.jxta.org/>, (2005)
6. Hausheer, D., Stiller, B., Design of a Distributed P2P-based Content Management Middleware; In Proceedings 29th Euromicro Conference, IEEE Computer Society Press, Antalya, Turkey, September 1-6, (2003)
7. Bhana, I., Johnson, D., Alexandrov, V.N., Supporting Ad Hoc Collaborations in Peer-to-Peer Networks; PCDS04, San Francisco, (2004)
8. VCard Overview; <http://www.imc.org/pdi/vcardoverview.html>, (2005)
9. Resource Description Framework (RDF), W3C Semantic Web; <http://www.w3.org/RDF/>, (2005)
10. Web Services Description Language (WSDL) 1.1, <http://www.w3.org/TR/wsdl>, (2005)
11. W3C SOAP Specification; <http://www.w3.org/TR/soap/>, (2005)
12. Apache Xindice; Apache Software Foundation, <http://xml.apache.org/xindice/>, (2005)
13. XML Path Language (XPath), W3C XML Path Language (XPath); Version 1.0, <http://www.w3.org/TR/xpath>, (2005)
14. Dublin Core Metadata Initiative (DCMI), Interoperable online metadata standards; <http://dublincore.org/>, (2005)
15. Java 2 Platform Micro Edition (J2ME), <http://java.sun.com/j2me/>, (2005)
16. JXME: JXTA Platform Project, <http://jxme.jxta.org/proxied.html>, (2005)
17. Connected Limited Device Configuration (CLDC); JSR 30, JSR139, <http://java.sun.com/products/cldc/>, (2005)
18. Mobile Information Device Profile (MIDP); JSR 37, JSR 118, <http://java.sun.com/products/midp/>, (2005)