Market Management of Peer to Peer Services

MMAPPS

European Fifth Framework Project IST-2001-34201

MMAPPS Introduction and Overview

The MMAPPS Consortium

BT Research, Ipswich, UK (Coordinator)
Athens University of Economics and Business, EL
Eidgenössische Technische Hochschule, Zürich, CH
Darmstadt University of Technology, D
Mysterian, Glasgow, UK
Telekom Austria, Wien, A
University of Lancaster, UK

For more information on the MMAPPS Project, please see:

www.mmapps.org

or contact:

Peter Loosemore
pp MLBG13, Orion Building
Adastral Park, Martlesham Heath
Ipswich, Suffolk,
IP5 3RE, UK
E-mail: peter.h.loosemore@bt.com

Executive Summary

The MMAPPS project is researching how to use techniques from economics and social science to tackle some of the fundamental difficulties in creating well-founded, sustainable, P2P applications. The project's central approach is to use market management techniques to improve cooperation between peers while enhancing the community-oriented structure of P2P architectures.

Typically, current P2P applications do not provide appropriate incentives to the peers involved. For example, peers wonder why they should contribute (at some cost to themselves) when they can still consume without contributing. In fact, it seems that most existing applications function only on the altruistic contributions of a small minority of peers with all the other peers "free-riding". This inefficiency means that such applications tend to work adequately only when the cost of participation is very low and the value high, a peculiar property of illegal file-sharing.

To allow wider applicability of the P2P approach therefore requires inclusion of techniques to provide better incentives to peers. Given such incentives, rational peers will choose to behave co-operatively and contribute their resources to maximise the efficiency of the community. This will allow applications with dramatically improved utility, efficiency, and robustness and hence enable whole new domains of use.

The typical suggestion for providing incentives has been to use micropayments for transactions. However, payment schemes can often act against the collaborative, community-forming nature of P2P. So MMAPPS is looking to enhance such schemes with innovative, non payment-based accounting schemes such as ratings, where individual members of a particular community receive a rating score based upon their contribution. This rating then affects how other members provide services to that individual. Alternatively, individual members might be required to provide some minimum level of contribution if they wish to remain a member.

The project is drawing upon current economic theory and social studies to develop a generic JXTA-based toolkit that provides a collection of such incentive schemes suitable for supporting a wide range of P2P applications. The toolkit will support application developers in devising a scheme tailored to their own application and community. This scheme can use payment mechanisms where appropriate but these will be alongside other, community-based, accounting, reward and punishment systems. The project is also developing the descriptive and analytical tools that can be used to study the viability and appropriateness of such schemes.

Among the application scenarios being investigated are:

- *Filesharing:* the prototypical P2P application. Investigating this scenario will allow the project's results to be compared with existing work, and thus make the paradigm more immediately attractive to the current generation of P2P application developers.
- *Peering of WLAN domains:* in which local WLAN administrators enable connectivity to roaming nodes in exchange for future connectivity on the roaming nodes' home network.
- *Collaborative team of Tele-Radiologists:* in which consultant radiologists are incentivised to co-operate with semi-skilled radiologists in order to help alleviate the global skill shortage in radiology.

Table of Contents

1 Introduction 5

1.1 Document Purpose 5
1.2 A Short History of MMAPPS 5
1.3 Related Research Initiatives 5
1.4 Document Structure 6

2 The Problem Domain 7

2.1 Approaches 7

3 The MMAPPS Vision 9

3.1 Market Management Approaches 9
3.1.1 Market-Based Pricing 9
3.1.2 Non price-based incentive schemes 9
3.1.3 MMAPPS Market Management Approach 10
3.1.4 Choice of mechanism 11
3.2 Future of P2P Applications 11
3.3 Community Rules and Peer Groups 12
3.4 MMAPPS Approach 13
3.4.1 Completeness and Extensibility 13
3.4.2 Core Assumptions 13
3.5 Project Development Strategy 14
3.6 The Middleware Software Architecture 14
3.6.1 The Use Model 14
3.6.2 Peer Model 15

4 The Economic Architecture 17

4.1 Description of the Economic Architecture 17
4.1.1 The Service Usage layer 18
4.1.2 The Accounting and Distribution layer 18
4.1.3 The Social Control layer 19
4.1.4 Exemplar Accounting Pattern 19
4.2 Classes of Accounting Patterns 20
4.2.1 Transaction Record-Based Accounts 20
4.2.2 Currency token based accounts 21
4.2.3 Other Accounting Patterns 22
4.3 Detailed Case Study 22

5 The Middleware Specification and Engineering-Level Design 23

5.1 The Middleware Specification 23
5.2 Application Development Paradigm 24
5.2.1 Adaptation Tasks 24
5.2.2 Adaptation Roles 25
5.3 Engineering-Level Design 26
5.3.1 Baseline Technology Choice 26
5.3.2 Key Engineering Issues 27
5.3.3 Overlay Networks 28

6 Economic Theory and Modelling 29

6.1 Reputation 30

7 Validation and Trials 32

7.1 Application Development Trial 32
7.1.1 P2P Wireless Network Confederation 32
7.1.2 Community Information Trading Application (CITA) 33
7.1.3 Restaurant Recommendation Application (DineMatch) 33
7.2 End User Trial 33

1 Introduction

1.1 Document Purpose

This document provides a general introduction to the technical ideas and outputs of the EU-funded MMAPPS project.

1.2 A Short History of MMAPPS

The primary goal of MMAPPS is to develop middleware for Peer-to-Peer (P2P) applications that will provide developers with the tools necessary to quickly develop appropriate incentive schemes for a wide range of novel, or improved applications.

The project began in April '02 and is due to complete in September '04. To some degree, it represents a follow-on project from the project team's prior research on the M3I project (which successfully studied the application of market-management techniques to internet connectivity and QoS provision [1]), but the original intention this time was to apply similar techniques to the domain of P2P systems.

The project partners, together, form a balanced team with the necessary set of complementary skills and interests to realise the vision. These include:

- theoretical and socio-economic analysis
- distributed systems and middleware design
- infrastructure and mass-market provision
- P2P application development expertise

Currently the project is focussed on the detailed design and implementation of the middleware itself, and the theoretical validation of our underlying assumptions.

1.3 Related Research Initiatives

Whist there is a long and rich history in applying market-management and economic principles to IT systems, there are important aspects of P2P systems that offer unique challenges (see Section 2: The Problem Domain) and that, partially as a result of many of these systems' failures, are only recently becoming recognised by the wider distributed systems and networks research communities.

As a brief indication of the emerging importance of this domain, we cite:

- the instigation of a major new annual workshop on the Economics of P2P Systems. For really the first time, this seeks to bring researchers from multiple disciplines together in order to discuss the economic characteristics of P2P systems, and the application of economic theories to P2P system design. Three people from MMAPPS (each from different sub-teams) have been invited to attend this workshop.
- recent predictions[1] that 'the next wave of compelling applications…will be in the realm of social software'. Social software is that concerned with the emergent behaviours of groups (ie communities) of individuals. According to Clay Shirky (a

[1] See, for example: http://www.techcentralstation.com/1051/techwrapper.jsp?PID=1051-250&CID=1051-042103C

well-known on-line commentator), early versions of social software (eg internet newsgroups) typically made three assumptions:

- that the groups could be of any size
- that anyone should be able to join them
- and that the freedom of the individual was more important than the group

In the current, truly global internet world, the communities that thrive 'violate most or all of the earlier assumptions'. Most have 'bounded size, or strong limits to growth, non-trivial barriers to joining, and enforceable community norms that constrain individual freedoms'. As will become clear, these themes (albeit under different names) are core concepts within our MMAPPS vision too.

However, despite much good work in this area, MMAPPS remains unique in its attempt to develop middleware that supports a wide range of different incentive schemes. Given the immaturity of this application domain, we believe that it will take many years before the 'best' set of incentive schemes becomes clear, and so a framework to support, and allow real-world experimentation of many alternatives seems highly desirable. Extensibility (in terms of supporting new incentive schemes developed in future) is thus a key requirement on our system.

1.4 Document Structure

This document is structured as follows. Following a short description of the problem domain in Section 2, Sections 3 and 4 outline the core principles of the MMAPPS vision, and the core economic architecture of our solution. Sections 5 and 6 describe how this architecture can be realised in software, and also highlight some of the key engineering-level issues we've faced.

Sections 7 and 8 are complementary, and describe firstly (Section 7) the theoretical modelling work being used to assess the vision, and secondly (Section 8) the broad range of work necessary to validate the middleware platform itself. Included in Section 8 is a description of each of the three, key application scenarios that we have chosen to build on the middleware.

2 The Problem Domain

P2P has become an over-used term with argument about what does and does not qualify. There are several metrics for the "P2P-ness" of a system, the most significant being the social and technical characteristics that are related to the fact that no central entity is involved in transactions between peers. MMAPPS adopts the following approximate definition of P2P intended to capture the aspects of the field most relevant to our work:

> *P2P applications are electronically-mediated communities co-operating for their mutual benefit, but with no centralised control.*

This definition covers conventional file-sharing, for example, but also a wide variety of other applications including the sharing of wireless LAN bandwidth, and technical support newsgroups. We are not dogmatic about the nature or completeness of the decentralisation of our technical mechanisms. We hope that our techniques apply to a wide range of applications with differing characteristics.

These P2P applications consist of a (possibly large) number of independently minded agents (the peers), each running software that implements the same communication protocols and resource sharing techniques. A basic feature is that peers share some form of local resource with the other peers in their community.

Many such P2P applications have difficulties in providing appropriate incentives to the peers involved. Peers ask themselves; "Why should I contribute (at some cost to myself), when I can still consume without it?" – this is the well-known free-riding problem. In fact, it seems that many applications function only on the altruistic contributions of a small minority of peers with all the other peers "free-riding" [2]. A study of Gnutella [3] has shown that 70 % of peers were free riding. However, these file-sharing systems still work adequately because it costs very little for a peer to provide (often illegally) copied files and the value of downloading them is high.

The MMAPPS project intends to extend P2P techniques to a much wider domain of applications including those where the cost of supply is much closer to its value.

2.1 Approaches

Solutions to this problem are emerging in roughly two camps. In one, system designers are making small modifications to the software that runs on peers to cause it to reward contribution. A recent version of the Kazaa [4] application, for example, holds an internal rating that improves as the peer provides files for others; then high ratings are rewarded by other peers (by giving a higher priority for downloading). The implementation of this particular solution is rather insecure, and hacks allowing peers to undeservedly improve their own ratings have quickly become available [5], but in other systems, for example DirectConnect [6], stronger social mechanisms are used to expel peers who are not living up to their obligations[2].

The other camp is dominated by economists who point out that such resource allocation problems can be efficiently solved by using suitable prices for consumption and contribution of resources. In such a scenario, one could imagine peers exchanging

[2] However, note that in DirectConnect these mechanisms operate through the actions of centralised hub operators.

tokens (currency) of a value equal to their consumption of resources. Thus the change in the number of tokens that a peer holds reflects their consumption or contribution.

This is the context in which the MMAPPS project was begun. However, analysis in the early stages of the project showed that P2P systems have three specific characteristics with very significant consequences for such incentive schemes.

Firstly, there is *the presence of externalities.* In a P2P system it is typical for the provision of a local resource to benefit the whole community (eg by increasing the diversity or resilience of provision) even beyond those who directly use the resource. A fundamental feature of a P2P system, then, is the presence of externalities—actions taken by one peer affect other peers; and in the absence of any correcting mechanism, these external effects are not taken into account fully by self-interested peers.

Secondly, there is *the degree of role symmetry.* In a P2P system, each participant is, potentially, both a supplier and consumer of resources. In this sense, there is symmetry of the roles played by members of the system (in contrast to e.g., a client/server system). [But note that role symmetry does not imply that peers are symmetric in other respects, such as the values they attach to making available and using resources].

Finally, *the nodes are distributed*. The objective of the P2P network is to allow resource sharing between peers that are distributed, for example by being geographically dispersed. Even when part of the system is centralized, the key issue is that the resources to be shared are decentralized (since they are located on the computers of individual peers).

These three characteristics taken in combination present a considerable challenge for the design of a well-functioning P2P system. As already described, the *presence of externalities* means that peers will, in general, have inappropriate incentives to join and use the P2P system. Hence, whether the system is being run by a benevolent manager who is interested in maximizing the total benefit generated by the system, or by a profit-maximizing system owner, the behaviour of peers needs to be modified. In other words, **regulation is required**. An important implication of *peer symmetry* is that a P2P system is, typically, **highly dynamic**. In a client/server system, the resources made available by the central server are relatively stable, adjusting slowly over time. In contrast, the resources in P2P systems are spread across the network of peers, and hence are inherently more dynamic, potentially changing rapidly and unpredictably as peers join and leave the system. Finally, the *distributed nature of the system*, combined with its dynamic nature, make it **very difficult to obtain global information about the state of the system**.

These factors had a great influence on the solution approach we subsequently pursued.

3 The MMAPPS Vision

3.1 Market Management Approaches

3.1.1 Market-Based Pricing

It will be evident from the previous section that the core problem we faced was: 'what are the appropriate mechanisms that are needed in order to give peers the correct incentives to contribute to the P2P system'?

As already indicated, the most obvious market-mechanism that might create the right incentives, and the one supported by a large body of theory and practice, is the use of a price-based market to co-ordinate and regulate behaviour. Such a mechanism had also been generally assumed during the Project's inception. In such a scenario, peers are imagined to exchange tokens (currency) of a value equal to their consumption of resources. Thus the change in the number of tokens that a peer holds reflects their consumption or contribution. One early implementation of such a scheme was Mojo Nation [7]. In this system peers earned "mojo" by donating resources to the community and spent them when consuming those resources. Thus, peers needed to maintain a balance between contribution and consumption. Similar economic approaches have been suggested for Grid computing [12].

However, a central issue in these price-based approaches is whether (and how) to regulate prices. A completely free market will achieve social optimality only in the absence of 'market failures' (due to the presence of monopolies, externalities, or asymmetric information between the peers). In practice, P2P systems often exhibit all three of these characteristics and so some price regulation is likely to be needed.

Such (free or regulated) market schemes are often advocated by economists, and in many situations has a number of advantages. A major advantage in standard situations is the degree of decentralization that can be achieved with prices. The only action that a central planner has to take in a classical market is first to announce a price, and then to adjust the price according to whether demand exceeds or falls below supply. Under well-understood conditions, this process leads to an efficient equilibrium.

To pursue such a solution in a P2P context the key challenge becomes how to perform this global regulation, since there is no global information and no central authority with the power to control the prices.

3.1.2 Non price-based incentive schemes

An alternative form of incentive can be provided by the imposition of constraints the consumption and provision of resources. An example of such a constraint might be a relation between the rate of resource availability and resource request actions that are made by a peer (Typically, acts of provision are rewarded by the ability to consume). Another example of such a constraint could be the requirement of a minimum contribution of resources in order for peers to be able to participate in the community.

For example, a recent version of the Kazaa application [4] holds an internal rating ("Participation Level") that improves as the peer provides files for others. High ratings are rewarded by other peers who give a higher priority for downloading. This particular solution, however, still does not give the appropriate incentives and is rather insecure.

Hacks allowing peers to undeservedly improve their own ratings have quickly become available, e.g Kazaa hack [5]. The home page for Kazaa hack itself acknowledges that "In light of the recent abundance of downloaders reaching 'Supreme Being (1000)' status, some users have decided to hastily assume that cheats were used, and immediately cancel their uploads." Clearly, such simple software solutions are inadequate.

In other systems stronger social mechanisms are used to expel peers who are not living up to their obligations. For example, within DirectConnect [6] each peer connects to a hub peer and has to obey to specific hub rules (most of them similar to the minimum contribution constraints described above. E.g., they should offer at least 5GB of music files). File sharing is then conducted with other peers connected to that hub. Cheating (in the form of incorrect accounting) is generally avoided since the owner of that hub has visibility of all service transactions, and can ensure accurate reporting, but this role cannot easily be distributed amongst other users of the hub.

The challenge with this approach in a P2P setting is how to enforce accurate accounting of contribution and reward in a decentralised manner.

If this could be overcome, we anticipate that application-specific constraints could be used to restrict the behaviour of the peers and so influence their decisions in order to achieve more efficient usage of the system.

3.1.3 MMAPPS Market Management Approach

The MMAPPS approach to stimulating cooperation in P2P systems builds on both the above approaches (both regulated market prices, and non-price based schemes). Since each approach is likely to be suitable for a range of P2P applications, we sought to develop a generic P2P middleware that took an inclusive approach.

Our first observation was that the previously-described two challenges (of distributed regulation, and distributed enforcement of constraints) are in fact equivalent – and are both forms of distributed social control. Our common solution was then to use social control from all the peers in the relevant peer community to either control prices, or to apply the constraints.

Specifically, this requires the concepts of:

- Community Rules - to encapsulate social control, and to define the structure of the market within that community. These can be used to either regulate prices, or to directly control participation in the community.

- A Group of Peers ('Peer Group') as a set of peers choosing to operate under a particular set of community rules, and responsible for enforcing the rules on other community members.

The community rules define how peers are expected to behave in P2P communities, and how they can be rewarded or punished for good or poor behaviour. Such rules may, for example, refer to peer identity, group membership, prices charged, reputation, or quality of service. They can, for example, set the maximum size of the group or state that members who do not contribute must be expelled from the group. Examples of rules used in this way in the real world are market rules of stock exchanges, electricity market rules, oil cartels, and environmental regulation.

Dividing the global community into explicit peer groups aids scalability and management. Division of communities into groups is used for example in DirectConnect and Sun's JXTA [9]. Shirky [10] writes about the desirability of groups in the general Internet and his arguments apply equally to groups in P2P systems. He claims that successful groups are those that have a defined maximum size or those that have non-trivial barriers to entry. Furthermore there must be enforceable community norms to limit individual freedoms. These can prevent the group from being taken over by hostile users.

3.1.4 Choice of mechanism

Given that we are developing a generic system that allows the application of many different market management schemes, then how should these schemes be chosen? What set of rules (i.e. what economic structure) is best for a given application is not a simple one and we cannot offer a simple rule-of-thumb[3]. This is largely the question addressed by the theoretical work of the project, described in section 6.That theoretical work seeks to identify characteristics of a P2P economy that make it amenable to analysis and hence the deduction of optimal or near-optimal management schemes.

We can also judge the suitability of a scheme by looking at the difficulties in implementing it. For many P2P systems, it seems that price-based incentive schemes are rather ill-suited to some of the key characteristics of P2P systems. In particular:

- the complexity and overheads associated with implementing price mechanisms in large, dynamic and highly distributed P2P system
- the potential impracticality of direct payments between peers (e.g. due to the anonymity of peers and the sheer volume of low value transactions).

Hence, we came to the view that non price-based constraints are the preferred market management approach for simple symmetric P2P systems offering relatively low cost resources. Prices, or some other form of monetary compensation, may be necessary in situations where both high and low value services are exchanged and peers do not have balanced contributions (some peers may be more on the provider side while other peers may be more on the consumer side).

The project is using validation based on experiment and economic modelling to test our general hypotheses and also to provide answers for various specific classes of application. We hope that the resulting descriptive and analytic frameworks will provide sufficient guidance to developers seeking to create new communities and sets of rules. Ultimately such communities will be validated by the marketplace, with the ones that offer the most to their members succeeding, and the weaker ones failing.

3.2 Future of P2P Applications

A second essential requirement on the MMAPPS middleware is that it supports future types of P2P applications. Following the analysis and requirements phase, the project concluded that perhaps the greatest potential was in (future) P2P applications with the following two characteristics:

- Asymmetric peers. It will be common for some peers to act as mainly suppliers and others to act as mainly consumers of resources. Such circumstances require additional incentives to be provided to peers that offer their resources beyond

[3] For confirmation of this difficulty, we cite the lack of agreement amongst politicians and economists about national economies!

merely being compensated by just participation in the community. This is contrary to the current assumption made by most P2P file sharing applications, for example.

- Applications in which services/resources are traded over a P2P network in such a way that the value, together with a Peer Group structure, manages and funds the infrastructure and resources that provide the service. The services themselves are of high value, or have some high cumulative value, while the management and funding of the infrastructure will rely on market rules that make the sharing of group resources reasonably equitable and efficient, and so provide sufficient incentive to participate.

3.3 Community Rules and Peer Groups

One of the main steps in the development of a new application is thus to define the rules that the group (of peers) will use. Our vision is that new peer group creators can easily express these rules and that the middleware will enforce them as automatically as possible.

The middleware must therefore provide services for applications to maintain information on the group of peers who have access to it. [Note that different groups may use similar (or even identical) application software, but may have no particular relationship to each other]. The peer group is thus the level at which community rules are defined.

Developers of a new group create rules that embody their idea of how the group will work, and then the platform encodes (some) of these rules into software that each member of the group runs. The structure of the rules should mean that each member of the group has an incentive to conform to the rules because they expect that the other members are doing so and will consequently encourage or enforce their own correct behaviour.

Different sorts of rules will need different sorts of enforcement. Some rules can be enforced mechanically on a per transaction basis, while others might need third-party checks perhaps on a random policing basis. Others still (involving a somewhat subjective perception of quality, say) might never be explicitly represented within the service, and just arise from a human understanding between the people involved.

Specific examples of Rules that might be applied to a Peer Group might be (when expressed in English):

- Your article will remain available only as long as you continue to provide twice as much disk space as the article takes up and whilst you leave your machine on.
- Your next subscription payment will be reduced by 50% if over the preceding month your average disk space contribution exceeded 10 Mb.
- Your number of uploads must always exceed your number of downloads minus 5.
- Your postings to a group should not include bad language; if a majority of peers feel you have broken this rule you will be expelled from the group.

These core concepts: Rules, Accounting Patterns and Peer Groups are embedded within the full Economic Architecture described in Section 4.

3.4 MMAPPS Approach

This sub-section summarises the approach taken by MMAPPS, following the project's analysis and requirements phase.

3.4.1 Completeness and Extensibility

In the apparent absence of a single, generically-applicable incentive scheme that would be suitable for a wide variety of market-managed P2P systems, MMAPPS decided to adopt an inclusive approach: ie to ensure that a very wide range of price-based and non-price constraint-based mechanisms could be supported. By building such an environment, it was thought that a wide range of (current and future) techniques could be experimented with, and the most successful set of solutions and rules emerge. ie the vision was of MMAPPS middleware being used to stimulate a competitive ecology of Peer Groups (communities), each experimenting with different rule-sets. We believe this is an ambitious and unique approach.

Thus the middleware will support a wide range of schemes, some more conventional than others. The theoretical research effort, however, is focussing on the space between the very simple (essentially un-enforced) rules that are used in existing systems[4], and 'traditional' price based approaches. The project is seeking to develop suitable system rules that are enforceable and actually restrict the behaviour of peers, while achieving an acceptable degree of system efficiency.

3.4.2 Core Assumptions

Our approach is based upon a particular model of P2P applications which we believe applies to a very wide range of P2P applications.

The project considers P2P systems that consist of defined groups of peers who engage in pair-wise acts of communication. Communication between peers will result in flow of value from one peer (the provider) to the other (the consumer)[5]. The providers contribute to the P2P community, while the consumers consume.

Our adoption of Peer Groups requires we assume that peers in the system are not completely anonymous. They are required to adopt one or more un-forgeable identities within the group. This is used to allow peers to prove their membership. It also provides peers with a unique identity against which accounts can be kept. Peers can create new identities at any time but there may be a significant cost of entry (or re-entry) to the group making the change of identity expensive. There could also be group entrance requirements based on stronger assertions of identity.

The project assumes that peers can make external monetary exchanges, should they wish to do so, but these exchanges may be decoupled from the flow of value in the group. The external payment is not required to occur online or to have any particular cost or delay per transaction. In particular, the project does not assume that peers use any particular special hardware (smart cards of hardware wallets), which can be trusted by other peers.

Within these general assumptions, the middleware is designed to support a variety of mechanisms to perform the accounting of contribution and consumption. These

[4] See the mention of Kazaa in Section 2.

[5] Though, as mentioned above, it is common for this flow to also benefit peers other than the consumer.

mechanisms have differing trade-offs between security and performance and make further differing assumptions about the system. One of the most important of these additional assumptions is whether or not we can involve third parties to oversee the execution of the mechanism. Such a third party could be an externally trusted third party service provider. However, they could also be another peer, or a group of other peers executing a joint signing or decision-making algorithm.

3.5 Project Development Strategy

The complete vision adopted for the project is very ambitious (given the relatively short duration of the overall project) and some aspects of it are based upon as yet unsubstantiated assumptions. However, the success of the project is not entirely dependent on these more risky aspects of the vision.

Firstly, the approach taken to building the middleware is that we aim to build a framework that supports the radical aspects of the MMAPPS vision as well as the 'traditional' price-based mechanisms. This means that even if our central hypothesis proves incorrect, much of the remainder will still be valuable. For example, the middleware can be used to implement a straightforward free market in the services that underlie the P2P application. Alternatively it can be used to implement an application equivalent to current applications where peers simply choose to contribute through altruism. Then from either of these baselines it allows application developers to add only those features (eg controlled membership or simple constraints) that are most relevant to their own application.

Secondly, we have adopted a staged approach to the most difficult and long-term parts of the vision. For example, a key research question for the project is the issue of how we expect users to become aware of the Peer Group's rules before they join the group – and how can they be sure that the Rules (as described to them) are indeed those enforced by the software? In order to make good progress on the engineering (and indeed, to help clarify the nature of the problem), we are pursuing an incremental approach. In the baseline case, as operating in the demo and trial applications, the rules and the rules enforcement are simply implemented by code built into the application by using the toolkit of modules provided by the middleware. In the longer term, we will progress to more abstract versions of the rules module which address this aspect of the vision more completely.

3.6 The Middleware Software Architecture

The first step to actualising the concepts and structures described in the previous sections above is to map them into an architecture that talks directly about the peers and the middleware. This is initially done in the Middleware Software Architecture. The full architecture [D5] describes the system in terms of four abstract models of which two are highlighted here.

3.6.1 The Use Model

This model describes the system and the requirements for the system from a user perspective. Here we define the way in which a single user accesses and uses resources on other peers over the network and those peers in turn may use the resources of other peers. Figure 3.1 illustrates the service usage as it is supported within the MMAPPS system. A service is the provisioning of resources or the execution of tasks of one or more (temporarily provider) peers on behalf of one or more (temporarily user) peers. A human user is associated with each peers node, and is assumed to have an objective that can be fulfilled by the services offered by application. On the corresponding peer, it invokes functionality to discover an appropriate service and to determine and contact one or more provider peers offering that particular service. Market management requires that an SLA (Service Level Agreement) be agreed upon before the service can be used or fulfilled. Access to the service will be negotiated between the involved parties or controlled by service-defined rules. In order to fulfil the service, the provider peer can draw on its local resources like hardware, applications, or content. In a more complex scenario, the service may also require to use (and fuse) other services residing on the provider peer. Finally, even the delegation of tasks to or use of resources from further peers is possible.

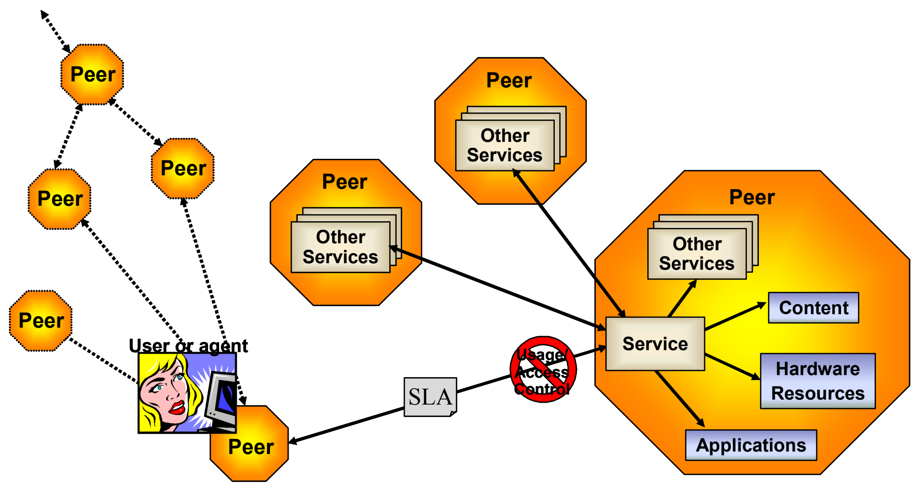

Figure 3.1: The Use Model

3.6.2 Peer Model

This details the description of a peer in the use model with its resources, services, and core functionality. Here we separate the local resources of a peer from the services it offers to the other peers based on those resources. The peer model is depicted in Figure 3.2. Each peer has a number of local resources available at the bottom layer. These resources consist of hardware and software resources and can be used either for providing services to other peers or for support of the core functionality. This functionality includes all mechanisms which are necessary to run the MMAPPS system itself, such as resource management or service discovery. In order to provide some of these mechanisms, it is necessary for the core functionality to communicate with other peers. For instance, service discovery is based on a distributed protocol where peers forward service search requests amongst each other.

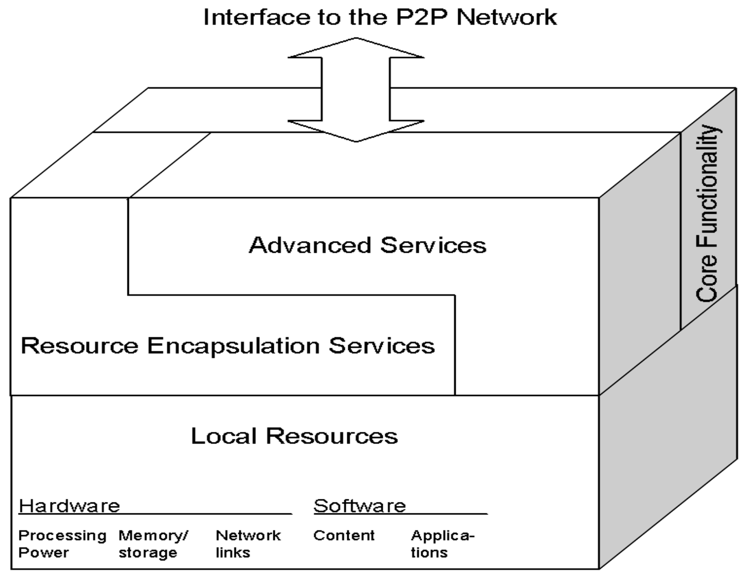

Figure 3.2: The Peer Model

The service layer supports two different types of services. Resource encapsulation services are very simple services that only provide access to local resources, e.g., storage space. The encapsulation is required to enable remote access to these resources over standardized interfaces and to allow market management through access rules and negotiations. Advanced services, on the other hand, contain much more functionality and may offer complete applications. In order to access local resources they can either directly build on them or on resource encapsulation services. The advantage of the second approach is the possibility to later include remote resource encapsulation services, e.g., if the local storage space is getting scarce.

4 The Economic Architecture

Following the development of the MMAPPS Vision described in Section 3, the project conceived and adopted the Economic Architecture shown in Figure 4.1. The architecture decomposes the distributed mechanisms that control consumption and contribution into three conceptual layers.

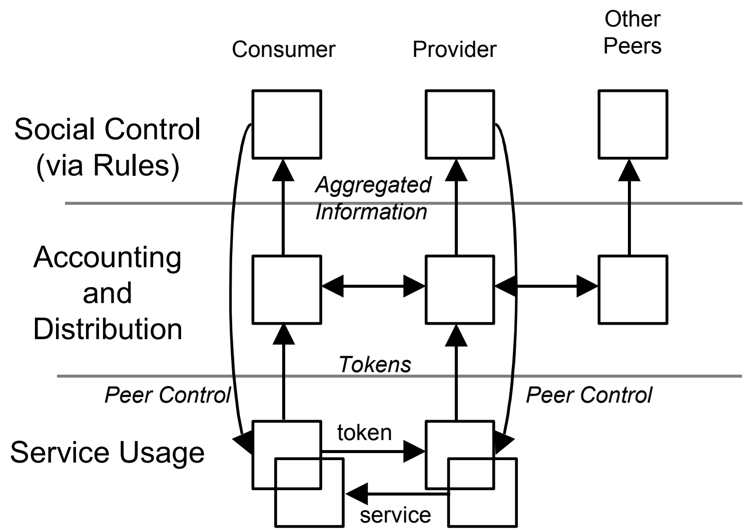

Figure 4.1: The Economic Architecture

4.1 Description of the Economic Architecture

The figure shows the three layers instantiated on three representative peers: a consumer of a service, the provider of that service, and other third-party peers. The figure shows what happens within the system when the first two peers interact, one providing a service to the other. The third peer then represents one other peer (or, in general, a group of other peers) that may get involved in monitoring the provider and consumer. The boxes then represent the behaviour of a peer enabled by the middleware, and the arrows represent the flow of information.

The Service Usage layer of the architecture shows a consumer peer using the services of a provider peer. This fundamental use of the service is accompanied by the flow of tokens from consumer to provider to account for his consumption. Information about the service usage and any token transfers then flows into the next layer up.

In the Accounting and Distribution layer, distributed mechanisms keep track of the service provision information by, for example, passing it to other peers to hold. These mechanisms then provide aggregated information to the top level. A variety of such mechanisms can be used according to tradeoffs between efficiency, security, resilience and the accounting information you choose to store.

The Social Control layer holds the community rules. In this layer peers check the data provided by the layer below and then perform actions (such as reward and punishment) based on community rules.

The layers are described in more detail below.

4.1.1 The Service Usage layer

In the Service Usage layer the act of service provision is combined with flows of accounting-related information such as receipts, invoices or forms of currency token. This information is represented by a token, usually signed by the sender. The standard example is that the consumer sends a signed receipt to the provider. For the purposes of price regulation, a signed invoice may also flow from provider to consumer. For other accounting mechanisms the tokens that flow may be defined by a bank and act as a currency token. The precise protocol for token exchange is defined by the accounting layer above.

In the simplest protocol the provider sends the desired item to the consumer and the consumer sends a signed receipt. However, each can fail to play his part and blame the other. This is the fair exchange problem to which there are technical solutions available (e.g. [11]) under certain assumptions, principally the trading of digital goods and the existence of somewhat trusted third parties. In environments without fair exchange mechanisms, such as eBay (where the goods are usually not digital), simple reputation mechanisms based on votes by the buyers have been proved to be surprisingly effective [12].

Because of the overhead of fair exchange mechanisms and the difficulty of finding an appropriate trusted third party, the architecture will primarily base exchange on a simple core reputation mechanism dedicated to detecting and expelling peers deliberately failing to comply with the exchange protocol. However, the architecture will permit the use of fair exchange protocols where their complexity is justified.

The service provision layer is extensively configurable including, for example, whether it is pre-pay or post-pay (this defines who takes the initial risk), the nature of the tokens exchanged and the protocol used, and how small the granularity of provision should be (ie how often to exchange tokens).

4.1.2 The Accounting and Distribution layer

The Accounting and Distribution layer provides mechanisms for recording and aggregating information about the service usage by peers, and for distributing this information appropriately. The flows of accounting information from peer to peer can be aggregated and delivered deliver aggregated information to those peers who need it for social control purposes while preventing individual peers from falsifying that information (eg to obtain unearned credit).

If tokens are used which can be directly redeemable at a bank then accounting can be very simple: the provider will simply store tokens until they can be efficiently redeemed. However, it is expected that, even here, it will be important to provide information about these exchanges to the social control layer to enable peer group oversight, particularly for price regulation but also for the enforcement of other rules.

Thus it is expected that the accounting system will obtain tokens for each exchange of value. Its job now is to maintain these records so that it can provide information to peers so that they can deduce how to act. It is here that the MMAPPS project offers a "toolbox" of mechanisms that achieve trustworthy accounting and distribution of these records. The mechanism used in this layer needs to match the nature of the token exchange, to support it and to provide appropriate information to the layer above.

In the case of currency tokens, the accounting mechanism needs to provide the information needed to discover breaches of price regulation. It also may need to find tokens to redeem and to maintain information (such as trustworthiness) about the issuers. For approaches based on keeping more detailed accounts, the mechanism needs to maintain the integrity of these accounts and to provide aggregated summary information (about other peers) upon which to base provision decisions.

4.1.3 The Social Control layer

Community rules are the key mechanism operating in the Social Control layer. MMAPPS provides developers with the tools to design rules that are appropriate to the particular application and peer-group that will use it. These community rules take information obtained from the accounting layer and make decisions on whether to reward or punish other peers. It is these community rules that are the key controlling mechanism in this social control layer.

Developers of a new group create community rules that embody their idea of how the group will work. Different sorts of rules will need different sorts of enforcement. Some rules can be enforced mechanically on a per transaction basis, while others might need third-party checks perhaps on a random policing basis. Others still (involving a somewhat subjective perception of quality, say) might never be explicitly represented within the service, and just arise from a human understanding between the people involved.

Rules can be purely about resources, or link underlying resources to high-level service provision, or be meta-rules (eg about what happens if you break the rules). Rules specify that something will happen under some conditions (eg if you break the rules you'll get a penalty), or they are requirements (like pre-conditions) on some requested action (eg you must be recommended before you can join the group).

The tools of reward include preferential service and real money while punishment can be financial, against entitlements held by the group or, ultimately, expulsion from the group, based on consensus-forming mechanisms such as voting.

4.1.4 Exemplar Accounting Pattern

A simple example of an Accounting Pattern conforming to this architecture might be a 'reputation-based' accounting pattern. Here, the 'tokens' correspond to ratings information (generated by the consumer peer), and the 'aggregation and distribution' layer may correspond to 'gossiping' between peers about the ratings information, and its subsequent aggregation into 'reputation' values for specific peers. In reputation-based systems, the `social control layer' may be enacted by users, who interpret the reputation values they obtain from the lower layer, and make their decisions on that basis. In other cases, the 'social control layer' is combined with a reputation-based policy, e.g. restraining peers with reputation values below a certain threshold from being customers to certain services.

The following section lists a range of other Accounting Patterns that conform to this model, and that we intend to support within the middleware.

4.2 Classes of Accounting Patterns

The key new abstraction in theMMAPPS architecture is the Accounting and Distribution Layer. Within this layer, we anticipate a rich and diverse set of accounting mechanisms to be possible, and these represent one of the most interesting and novel aspects of the middleware. Some of these mechanisms, which we call *accounting patterns*, are already well-known (or obvious), a few others have recently been invented (by ourselves and others), and many more (we believe) remain to be devised. The intention is to provide application developers with a wide choice of such patterns, and so allow them to be flexible in the construction of their P2P systems. A generic interface is provided to all the accounting patterns so that the newly-created patterns can easily be incorporated into the basic middleware (see Section 6).

The primary purpose of all the accounting patterns is to keep track of the behaviour of peers within the community, which is necessary to ensure that they follow the community rules. The accounting patterns typically have differing trade-offs between security, efficiency, quantity of information stored and performance. The correct execution of these mechanisms may also require monitoring by third parties – which could be an externally trusted third party service provider, or alternatively, another peer, or group of peers, executing joint signing or decision-making algorithms (for example).

The MMAPPS project is currently implementing a number of these alternative accounting patterns and testing their operation with the rules. The patterns involved may, for example, keep the information locally, and provide public access to other peers. Alternatively third parties may keep the information. Other mechanisms use non-forgeable cryptographic currency tokens as a core part of the mechanism that prevents cheating.

Further mechanisms have been suggested to allow peers to gather group consensus information about each other, such as reputation systems based on gossip or query and also voting mechanisms. Such mechanisms are formally similar to these accounting mechanisms, though they may not be based directly on the accounting tokens in quite the same way.

A basic classification of the accounting mechanisms is described below. Firstly we present three different methods of storing the data as a collection of account transactions where no currency is used. Then we present a number of ways to hold the data as a collection of currency tokens. Finally we discuss other forms of accounting that can be implemented as accounting patterns but which have different functional properties.

4.2.1 Transaction Record-Based Accounts

Local accounts

The simplest accounting mechanism is that each peer keeps a record of their own behaviour. This mechanism is insecure. It gives consuming peers no incentive to keep accurate records; this is exactly the problem that faces Kazaa.

Public accounts

The public accounts mechanism requires peers to keep complete public accounts of their transactions. Peers cannot delete consumption transactions because the corresponding provision transaction in the provider means they are always at risk of discovery. Peers can regularly audit the accounts of other peers to discover false accounts. The mechanism allows history to be eventually rolled up into a carried-forward balance. This can be achieved by asking a group of peers to sign any proposed balance. Once a balance is adequately attested (say, by a minimum number of other peers) it can be added to the account and the history deleted.

A similar, but less general, system is described by Wallach [13].

Third party account holders

The third party account holders mechanism provides an alternative remedy for the insecurity of local recording of the peer's account. For example, if a Distributed Hash Table such as Pastry [14] is used, the peer addressing mechanism can designate one or more other peers to hold the account. The provider sends each signed receipt to its account holders and then from them on to the account holders for the consumer. The account holders need to hold these records only for a short time at most and then they can roll them up into the aggregated balance.

4.2.2 Currency token based accounts

Instead of keeping an account of all transactions, a mechanism like digital cash can be used in which each peer holds an account represented by a number of cryptographic tokens that they cannot forge. The exchange of these tokens is managed by the lower layer of the architecture. The accounting layer can arrange for the issue and redemption of tokens when necessary.

Cryptographic tokens need to be issued by someone. Furthermore, to control the possibility of a peer double-spending a single token, they must eventually return to the issuer (in the absence of trusted hardware at the peer). The project has identified three mechanisms based on the identity of that issuer:

Tokens issued by an external bank

In the usual digital cash case, the issuer is a trusted bank external to the peer group. However, the cost of using such a bank is not negligible. Furthermore, the use of an external bank imposes restrictions on the peers (eg that they use a common banking system) that are somewhat contrary to the co-operative and ad-hoc nature of P2P applications.

Tokens issued by an internal distributed bank

It is also possible for a sub-group of the peers to create a distributed bank that can mint its own currency. Threshold cryptography [15] can be used to allow the whole sub-group to create tokens together but to prevent any smaller sub-group doing the same. However, the necessary intra-bank communications seems to be a heavy overhead. This mechanism has been implemented within the project [16].

Tokens issued by each peer

A particularly appealing approach is to allow each peer to mint their own currency. This means that providing peers will rapidly collect coins from a variety of banks. To control double spending, peers will need to regularly redeem coins at the issuer. However, the issuer may have no service to provide the peers to whom they are in debt. They may be able to redeem their own coins against coins from another issuer. But these too, will need to return for checking against double spending. It is not yet clear to us if this exchange process is manageable; further research is planned.

4.2.3 Other Accounting Patterns

Other mechanisms have been suggested to allow peers to gather group consensus information about each other, such as reputation systems based on gossip or query and also voting mechanisms. Such mechanisms are formally similar to these accounting mechanisms, though they may not be based directly on the accounting tokens in quite the same way. The project plans to provide some of these mechanisms for the use of the rules in the layer above.

4.3 Detailed Case Study

We have implemented a version of the second in which the tokens are signed by a group of peers, representing an internal distributed bank.

Specifically it implements a token based account whereby tokens are issued by an internal distributed bank [16]. Here each peer holds an account with a specific number of tokens. Each token has been issued by a group of "super-peers" chosen randomly who use threshold cryptography [15]. A peer spends a token by sending it to another peer. When a peer provides a service it collects tokens from other peers, but cannot spend these foreign tokens. Foreign tokens are exchanged for own tokens via a token aggregation process performed by a group of super-peers at which point tokens spent more than once will be detected. Super-peers are elected from the peer-group and their membership changes slowly. The approach assumes that at least some of these super-peers will be trustworthy. Then the threshold cryptography ensures that the non-trustworthy peers cannot abuse the token system.

5 The Middleware Specification and Engineering-Level Design

5.1 The Middleware Specification

The MMAPPS Middleware Specification describes a software design for the Economic Architecture that supports P2P application developers in adding powerful incentive mechanisms into their applications with the minimum of difficulty. It describes, in a language-neutral manner, the necessary supporting infrastructure to be realised in software modules. Detailed documentation for the specification and design can be found in [D10]. The outline specification for a single peer is shown in Figure 5.1.

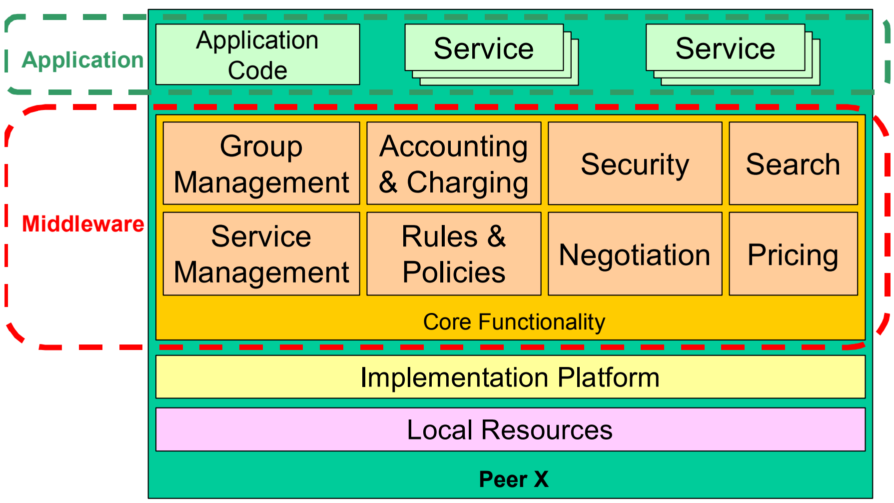

Figure 5.1: Outline of the Middleware Specification

The two upper layers of the economic architecture are represented in this specification by the following two modules within the middleware:

- The *Accounting & Charging* module encapsulates both the Accounting and Distribution (middle) layer of the Economic Architecture (see Figure 4.1), together with the token-exchange part of the Service Usage (lower) layer.
- The *Rules & Policies* module fully encapsulates the Social Control layer of that architecture (the community rules) but also includes local rules, 'policies', stipulated by that peer. Usually we expect a peer to specify its policies to be consistent with the community rules[6] (e.g. the community rules may allow peers to discriminate over who they provide services to, and so a policy may express that discrimination).

[6] Peers are free to implement policies that conflict with the community rules but they then risk being observed by other peers in the community and so being subject to the punishments described in the community rules.

Further middleware modules then provide the essential supporting infrastructure within the middleware:

- The *Group Management* module explicitly supports the key MMAPPS concept of a peer group (i.e. the fact that peers operate within well-defined communities). When a particular peer joins, leaves, or is evicted from a Peer Group, this change in membership status is managed through this module.
- The *Search* module supports mechanisms that allow the discovery of the services offered by other (remote) peers. ie the process of service discovery.
- The *Negotiation* module encapsulates the functionality that supports the bilateral negotiation of service provision contracts between peers. This may include negotiation of price, service QoS, payment periods, and pre- or post-payment. In the future this could also include multi-party negotiation (e.g. support for auctions).
- The *Service Management* module is responsible for controlling the execution of a service, including starting and configuring the service.
- The main purpose of the *Security* module is to provide access control functionality. In order to provide this functionality, the Security module internally must also provide and make use of key management and encryption mechanisms. Some of these mechanisms, especially encryption, are also offered to other modules, e.g., in the form of secure communication channels.
- The *Pricing* module is responsible for setting appropriate prices for services (if the community uses price-based incentives), which allow the service provider to be competitive in the market and create revenues which generate profit for him.

Finally, the specification shows (possibly multiple) Service modules, and the Application module code itself:

- The *Service* module essentially represents the service(s) provided by one peer to another, and whose activity is accounted for by the Accounting & Charging module[7].
- The *Application Code* module then contains all of the residual application-specific code – including that required in order for the user to interact with and control the middleware and services.

5.2 Application Development Paradigm

5.2.1 Adaptation Tasks

P2P application development using the MMAPPS middleware is a very open process, that requires several tasks of adaptation of the middleware:

1. Abstract middleware implementations of the Rules & Policies module and the Service modules need to be configured or specialised into application-specific versions.

[7] In MMAPPS, only the activities of the Service module can be accounted for. We assume that the activity of (say) searching (performed by the Search module) is a trivial overhead in comparison to the utility offered by participation in the community.

2. The Application Code module code itself is able/required to directly invoke standard API calls on some of the core middleware modules.
3. The Accounting & Charging module must be configured and adapted to support the needs of the application, e.g., accounting and charging by the data volume transferred by its services.

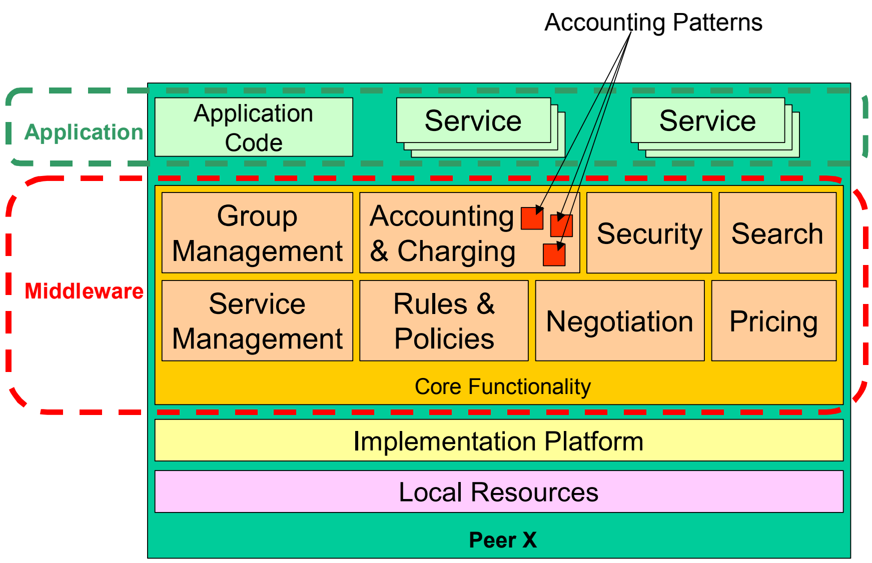

Figure 5.2: More Detailed Middleware Specification

The last step especially demonstrates the high degree of extensibility in the middleware. The Accounting & Charging module is designed so that it can support a diverse (and expanding) range of accounting patterns[8], only some of which are provided as a standard part of the middleware code-base. This scheme (internal to the Accounting & Charging module) thus provides a means for 3rd party accounting pattern developers to extend the middleware as shown in the figure above and have their patterns made available to a wider development community. The Accounting and Charging core implements a simple container for the accounting patterns used in the application. This container is configured by the Rules to create instances of the patterns needed. It then receives events of contribution and consumption from the Services module and distributes them to the accounting patterns that are interested.

5.2.2 Adaptation Roles

The middleware adaptation described above involves multiple roles: service developers, account-pattern developers, application developers, and the peer groups themselves.

Service developers write core pieces of service provision and usage functionality (eg the core ability to upload, or download a file; or the provision of WLAN Internet

[8] Some of these were described in Section 4.2

connectivity). Separately, whilst the middleware provides a basic set of accounting patterns, third party account-pattern developers will be able to develop new mechanisms and integrate them into the middleware.

Application developers use both the services and the middleware (and the set of available accounting patterns) in order to develop the application. Many different groups may adopt this application, each choosing to specify somewhat different sets of community rules.

As an example of the process, consider file-sharing. The core pieces of functionality (eg file upload and download) are written as services. Accounting for the usage (uploaded and downloaded files) is achieved by using one of the accounting patterns provided or writing a new accounting pattern if none suit. Another pattern may be used to track the reputation of each peer for providing good quality content. The middleware itself will be used to negotiate service provision, manage group membership, etc. The application will provide a GUI and control the middleware, calling the appropriate services and support functions. Each group can use the application for its own ends. For example, one group may use it to trade their own computer written music while another will trade pictures of aeroplanes taken by individual users at air shows. Each group is free to choose (or write) their own set of rules. One might have a joining fee, another might not. One may require one file upload for every one downloaded while the other does not.

5.3 Engineering-Level Design

This section outlines the key problems faced, and decisions made in designing a concrete implementation of the Middleware Specification.

5.3.1 Baseline Technology Choice

To begin building a concrete implementation of the MMAPPS middleware, we needed to select an implementation language or baseline distributed computing technology. Several strong candidates were considered, including:

- Microsoft's .NET
- Sun's SunONE
- Project JXTA/Java

The combination of JXTA and Java is the most suitable for both technical and strategic reasons:

Technically:

- JXTA is a Java-based distributed computing technology specifically designed to support P2P systems, and so potentially offered more suitable primitives on which our layer(s) of middleware could be developed.

Strategically:

- JXTA and Java have large, established developer communities – ultimately providing a clear exploitation and testing environment once the middleware passes a basic validation.
- JXTA is an open-source initiative, giving greater opportunity for integration and refinement of previously-developed code.

5.3.2 Key Engineering Issues

Having made the choice of JXTA/Java as our baseline technology on which the MMAPPS middleware could be built, we were faced with five significant engineering issues:

Mapping MMAPPS To JXTA

Since JXTA itself is a toolkit to support the development of P2P systems, there is inevitably some overlap of functionality between itself and (at least the original) MMAPPS design. The main areas affected were: Group Management and Search, where JXTA techniques have so far proved adequate. Adopting JXTA techniques wherever possible is largely motivated by our desire not to reinvent the wheel, so to speak, but to concentrate on those areas where MMAPPS can make a real difference. Also, using JXTA will make the transition to adopting MMAPPS as straightforward as possible for those developers already using JXTA.

Inter-module Communications Mechanism

Our applications will be inherently multi-threaded. Any particular module can expect to be called by more than one thread at the same time. Java does not provide facilities to automatically encapsulate threads at interface boundaries. Thus, the interfaces need to (at least implicitly) define their concurrency structure; that they are thread-safe; how they block with respect to other threads in the same module, and so on.

In principle, we could simply define this information at the interface. However, this description typically damages the encapsulation of the interface. For example, for each method on a module we need to know the possible sequences of locks it can acquire (including in any other modules it calls) to be sure that it cannot deadlock with any other method (called by another thread) on any other module.

Instead, we have developed a runtime infrastructure that isolates threads within a single module and prevents module code needing to worry about synchronization or deadlock.

In this, all calls between modules are asynchronous. The call will return immediately and the processing of that call will be done at the convenience of the called module. The most important restriction this implies on the interfaces is that all methods must return void, which means that many modules need to support callback interfaces too.

Inter-Peer Communication

All inter-peer communication will be between modules of the same type, e.g the negotiation module can only talk to another negotiation module in another peer. This decision means that module developers can work on modules independently.

Inter-peer communication will use JXTA advertisements and pipes.

The Application Development Paradigm

Of great importance too is the detail of how the middleware is presented to application developers. Standard options are to present middleware as either a toolkit, platform, or development framework, though complex functionality (such as is provided by MMAPPS) is usually some combination of these paradigms. The major trade-offs are between simplicity and flexibility, and currently our middleware has a predominantly 'framework' flavour to it. Additionally, a recurring pattern is that some middleware modules are *configured* by the application developer, which we regard as a somewhat simpler and less error-prone task than expecting them to *specialise* abstract (provided) versions of those modules.

The Account-Pattern Development Paradigm

A key objective in designing the middleware was to allow it to support new accounting patterns in the future – and for these new patterns to then become available (via MMAPPS) to application-developers. Although the number of people developing new accounting patterns will be far fewer than those developing new applications, increased experimentation and development of these patterns is something we strongly wish to encourage – and so we are keen to design this as well as possible.

These engineering issues constitute a significant part of the effort for the entire project, and the quality with which they are solved will strongly influence the degree to which the MMAPPS concepts are accepted by the wider community.

5.3.3 Overlay Networks

The formation of the overlay networks provides a vital abstraction, which allows the development of complex communication paradigms among the peers. Though the adaptation of JXTA makes its default overlay construction mechanism as the default in the MMAPPS middleware as well, it can be replaced by more efficient approaches.

The design process of the P2P overlay networks is a complex procedure, where a large number of (often) conflicting requirements should be met. For example, providing complete autonomy in the operations of the peers requires higher communication among them in order to perform these operations (e.g. searching and updating advertised items), which increases the maintenance cost and lowers the effectiveness of the system.

Within MMAPPS, the requirements of designing overlay networks have been extensively investigated and a number of designs have been explored in depth. AGILE and SHARK are systems that take advantage of DHTs and groups of interests to provide efficient solutions for lookup and search services, respectively. Omicron is a hybrid approach that aims at the effective adaptation of the overlay to the heterogeneous physical capabilities of the peers (which can range from PDAs to clusters) and the behaviour of the users, taking into account important requirements such as the large scale and the dynamics of the network.

6 Economic Theory and Modelling

The modelling work is focused on the important theoretical issues underlying the MMAPPS vision. Its main goal is to validate the main assumptions made about the provision of incentives, propose suitable mechanisms for the efficient resource allocation of resources into a p2p community and give directions for the selection of the critical parameters of the proposed mechanisms (e.g. the efficient design of rules). Many application-specific issues are also investigated in order to provide useful examples where this theory can be applied.

We model P2P resource allocation as a public good problem. In the case of file sharing for example, the copying of a file by one peer does not prevent another peer also from copying it (we consider the possibility of peer exclusion later). Content request, on the other hand, is a public bad: a request rate by an agent increases the load on all other agents equally (in an appropriate statistical sense). The standard problem with public goods is that the (Nash) equilibrium in which agents determine their contribution levels to maximize their own utility is typically inefficient relative to the social optimum, in which contributions are set to maximize the sum of all utilities. The main reason is free-riding.

The existence of externalities rules out as inefficient the adoption of a free-market approach using market-defined prices for controlling such a system. This fact in conjunction with the implementation difficulties of micropayments in P2P systems motivated the use of system specific rules relating consumption and contribution of goods by peers as the main incentive mechanism of the MMAPPS system. Such rules must be designed by a system manager and can adapt to the various states of the system.

We are pursuing two different modelling directions for computing optimal rules. The one assumes complete information of peers' valuations and costs (their type). In this case we compute the optimal prices and rules which, if set by a social planner, would lead the system to the most efficient equilibrium. However, the assumption of complete information is rather strong but can be relaxed by assuming that we know only the statistics of the number of the different peer types.

The other modelling direction deals with incomplete information. It is based on new and existing results that indicate that as the number of participants in a public good provision problem becomes large, the solution of the provisioning problem, when exclusions are possible, can be approximated by solving a simpler problem with a policy based on entrance fees [17].

However, there is still a considerable computational burden to calculating the minimum contribution with the mechanism proposed in [17]. Rather than undertaking that task, we have shown that for a large class of models including the one used in MMAPPS, a simple fixed fee policy which is independent of the declarations of the peers, is enough to get us within $o(N)$ of the second-best . In many interesting cases, this fixed fee can be easily computed by solving a simple optimization problem (see [18] [19]).

In [17] we have introduced a general model that can be used to calculate the efficiency of both the complete and the incomplete information schemes so that they can be compared. Numerical analysis indicates that these schemes converge to a fixed proportion of the full information optimal as the number of peers in the network becomes large. This result means that it is not necessary to collect large amounts of information, or to undertake complicated calculations, in order to implement the correct incentives in a large peer-to-peer network.

Furthermore, we have specifically studied the use of reciprocity rules, which relate consumption and provision of resources [D8], and extended the basic model to include the notion of QoS (in terms of the blocking probability offered for a specific resource, e.g. a file download or access to a WLAN) [18].

In this model, the optimal prices and rules need to be personalized, and on-going work is investigating the effects on efficiency of the use of uniform prices and rules when the number of peers is large, and how much system information would be needed to compute them. For the use of rules, a stability result is proved.

Finally, we have developed a "free market" economic model for p2p content distribution [19], in cases where the system is not symmetric and high-value content is traded in a p2p community.

6.1 Reputation

A further major area of investigation for the modelling is in designing reputation-based mechanisms, suitable for our chosen MMAPPS case studies, for facing information asymmetry and improving the achievable efficiency for peers. Information asymmetry is an inherent characteristic of P2P environments and reputation can be an appropriate means for revealing hidden information or type of peers.

Working in this area, the project has performed an analysis of the appropriateness of different reputation system designs with respect to the various P2P search architectures: pure P2P, P2P with centralized index, and P2P with distributed index (in super-peers or clusters). The possibility for introducing randomisation in order to decrease the complexity in the reputation derivation was also explored. Randomised aggregation with super-peers mechanism has been identified as having the smallest communication overhead, and whether small fractions of feedback information were enough for calculating accurate reputation values was investigated [22].

A simple P2P reputation model has been developed that captures most of the relevant key issues. According to this model, there are two different types of peers, which succeed with different probabilities in providing services. Client peers provide binary feedback based on the outcome of their transactions. If the different outcome in transactions can only be affected by the hidden offering quality of peers, then we refer to this reputation metric as "Reputation for QoS", while if the different outcome in transactions also depends on hidden actions by providing peers, then we refer to this reputation metric as "Reputation for Performance". It has been demonstrated by simulation experiments that the straightforward use of reputation (select the providing peer with the highest reputation and be indifferent when serving peers) results in inefficiencies for high performing peers. Thus, we argue [22] that reputation-based policies have to be employed in an incentive compatible way in P2P systems. We defined two orthogonal dimensions: "provider selection" and "contention resolution" . We showed using simulations that both these dimensions have a great impact to the achieved efficiency of the system. We analysed the cycle that is formed by the reputation-based policies regarding reputation calculation and its exploitation. Also, we dealt with the efficient aggregation of the ratings feedback in this distributed environment. We proposed that the efficient aggregation of the ratings can be accomplished by aggregating only a rather small fraction of the feedback randomly selected. Simulation experiments indicated that this is enough information for the fast and accurate calculation of the reputation values even if the peer-to-peer population is renewed with a high rate.

MMAPPS also deals with the provision of a proper reputation-based mechanism for having peers eliciting honest feedback; we refer to this as Reputation for Trustworthiness (Reputation for Credibility in other contexts [23]). Reputation for Trustworthiness has been clearly separated from Reputation for QoS (or Reputation for Performance) and a mechanism providing suitable incentives for credible behaviour has been specified. According to that, each peer is assigned a value expressing his credibility in reporting. Any inconsistency in accounting reports between peers results in (credibility dependent) punishments for both inconsistent peers. This mechanism has been game-theoretically justified to provide the right incentives for peers to be sincere. Furthermore, the stability of the mechanism against various liars' strategies has been theoretically justified using a Markovian model. Specifically, it has been proved that, using the proposed credibility mechanism, the number of unpunished sincere peers in the P2P system is almost always larger than the number of unpunished liars, when the total number of sincere peers is larger than the number of liar ones. Simulation experiments have been conducted against various liars' strategies and the total efficiency achieved by the credibility mechanism is close to the case of having no liars in the system. Furthermore, a centralized and a distributed approach for implementing the credibility mechanism have been proposed. It has also been proposed that the ratings for reputation for performance are weighted by the credibility of the raters. Using reputation-based policies along with the credibility mechanism, the total efficiency of the P2P system approaches that of having all hidden information about performance and credibility revealed to all peers, as shown by simulation experiments.

Another interesting issue under research is that of reputation about SLA consistency in SLA markets. According to this work, a peer's willingness to pay for an offered service is weighted by the reputation of one of the transacting parties. Of course the price each peer is willing to pay depends on how he is related to risk. Social welfare maximization is the objective in this context and this subject will be further investigated. This subject will be investigated both theoretically and by simulation experiments.

7 Validation and Trials

There are many aspects of MMAPPS that need to be validated, some of which are fundamental, and some of which are supplementary/peripheral, and each of which calls for a different validation strategy. For example, the fundamental economic robustness of Rules is being validated directly through our own theoretical modelling work; whilst the validity of our vision of a highly-abstract (configuration-only) Rules module is likely to remain a long-term, multi-project objective.

However, the development of a small number of carefully-selected applications is an especially important way to validate several key aspects of our system. These include:

- the effectiveness of incentive schemes at improving P2P efficiency
- the ability of our middleware to support a broad-range of accounting patterns
- the ease with which an MMAPPS incentive scheme can be added to existing P2P services
- the ease with which new accounting patterns can be added into the MMAPPS middleware
- the breadth of P2P applications for which MMAPPS is suitable

The first of these necessitates some form of real-world trial (as indeed was planned from the outset), whilst (say) the third of these can only be truly validated when MMAPPS middleware is released to third party developers (as tentatively planned, via the JXTA community). Two of the partners especially, Mysterian and Telekom Austria, have a strong application focus and thus a critical role in helping validate the middleware.

The project has defined two formal trialling tasks intended to validate different aspects of the middleware.

7.1 Application Development Trial

Within this task we are performing the development of three widely differing applications, by application developers who are somewhat removed from the core middleware team. We are recording the nature of this application development process by using a bug-tracking tool to structure the interactions between the application developers and the middleware team, and also by asking the application developers to record their ongoing experiences of the development process.

The three applications we are developing are as follows:

7.1.1 P2P Wireless Network Confederation

The *P2P Wireless Network Confederation* (P2PWNC) [20] [21] is a community of WLAN administrative domains that offer network access (e.g. Internet connectivity) to each other's registered users. The great benefit of near-ubiquitous access that these roaming users could enjoy compensates for their domain's cost of providing access to visitors. Existing roaming schemes utilize central authorities or bilateral contracts to control the parties' behaviour. In contrast, a P2PWNC forms a P2P community in which participating domains are autonomous entities and they make independent decisions concerning the amount of resources (access bandwidth) they contribute. As a result, similarly to existing P2P systems, a P2PWNC would ordinarily suffer from free-riding if no incentive mechanisms exist to ensure that domains offer the amount of resources that is economically justified.

This application has been successfully prototyped using the MMAPPS middleware with rules written to require reciprocity amongst the peers in order to regulate (and incentivise) an individual administrator's access provision policy.

7.1.2 Community Information Trading Application (CITA)

This is a general application for trading information between a group of peers. It could take the form of, say, a local area multimedia newspaper for local people (or a Reuters-style news agency formed from free-lance reporters), or it could be a community of interest such as plane spotters.

We anticipate there being specific membership rules in order for peers to join a group, with peers taking on different roles concerning production and consumption. Since the producers of the information typically would not consume corresponding services from other peers, there must be some form of monetary compensation in order that it be beneficial for them to participate in the P2P community. The key Rule in the group could be that a specific membership fee is set, and that the amount of money thus gathered could later be shared between the producers according to their reputation. An appropriate accounting pattern would need to be chosen for this measure of reputation.

A variant of the CITA scenario, of strong interest to one of the project partners, is to apply such a scheme to the domain of tele-medicine. Here, the intention is to develop a collaborative application that creates incentives for all parties to participate, and so increases the total capacity for radiological analysis (the is a global skills shortage in this domain).

The idea is to utilise the 'spare resources' at the edge (itself a very P2P-type concept) – in this case *semi-skilled, but unqualified radiologists*. Whilst well-established clinical practice requires that only qualified consultant radiologists are responsible for a patient's analysis, that still allows semi-skilled workers to work with the consultants in order to increase throughput and efficiency. The role of MMAPPS in this scenario is to appropriately stimulate both the semi-skilled and consultant radiologists in order to form a sustainable co-operative group.

7.1.3 Restaurant Recommendation Application (DineMatch)

This is an application which stores reviews and recommendations by a community of restaurant goers and then searches for and provides these reviews to the peers when needed. The economic problem here is that plainly peers have little reason to provide reviews to the system unless they are somehow rewarded for that contribution. We intend to use the MMAPPS middleware to provide that reward.

This application is being developed by a university student outside of the MMAPPS project.

7.2 End User Trial

The primary objective of the MMAPPS project is to research and develop the middleware necessary to develop better P2P applications. Thus the primary trial is of how this middleware supports its users, ie P2P application developers. However, the trials would not be complete without also checking that the type of application that results from this development process is credible; that users can use it successfully and that they can, at least in some cases, find benefits from its use.

We are trialling these latter aspects by developing a P2P application (modelled on a pre-existing Telekom Austria service) using the middleware and deploying that application to a selected group of users for use and evaluation.

The application to be used in this end user trial will be based on Telekom Austria's MP3 swapping service (http://mp3.aon.at/). We will develop a P2P version of this service based on the file-sharing application already developed.

We will then trial this application in a school setting for a fixed period.

[D14] and [D17] contain further information on the applications and trials described in this section:

References

[1] M3I Project, http://www.m3i.org/

[2] G. Marwell and R. Ames, "Experiments in the provision of public goods: I. resources, interest, group size, and the free-rider problem", *American Journal of Sociology*, no. 84, 1979.

[3] E. Adar and B. Huberman, "Free Riding on Gnutella", *First Monday*, vol. 5, October 2000

[4] The Kazaa P2P application, http://www.kazaa.com.

[5] Kazaa Hack, http://www.kazaahack.net/home.html.

[6] The DirectConnect P2P application, http://www.neo-modus.com.

[7] J. McCoy, "Mojo Nation Responds", http://www.openp2p.com/pub/a/p2p/2001/01/11/mojo.html.

[8] R. Buyya, H. Stockinger, J. Giddy, and D. Abrams, "Economic models for management of resources in grid computing", in *Commercial Applications for High-Performance Computin*g, 2001.

[9] http://www.jxta.org/

[10] C. Shirky, "Social Software and the Politics of Groups", http://www.shirky.com/writings/group politics.html/

[11] N. Asokan, "Fairness in electronic commerce", Tech. Rep. RZ 3027 (#93073), 1998.

[12] C. Dellarocas, "Analyzing the economic efficiency of ebay-like online reputation reporting mechanisms", 3rd ACM Conference on Electronic Commerce, 2001.

[13] Tsuen-Wan "Johnny" Ngan, Dan S. Wallach, and Peter Druschel, Enforcing Fair Sharing of Peer-to-Peer Resources, *2nd International Workshop on Peer-to-Peer Systems (IPTPS '03)* (Berkeley, California), February 2003.

[14] Rowstron and Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems", IFIP/ACM International Conf. On Distributed Systems Platforms (Middleware), pages 329-350, 2001.

[15] Y. Desmedt and Y. Frankel. "Threshold cryptosystems", In G. Brassard, editor, *Advances in Cryptology*, Crypto '89 (Lecture Notes in Computer Science 435), pages 307{315, Santa Barbara, California, U.S.A., August 1990. Springer-Verlag.

[16] David Hausheer, Nicolas C. Liebau, Andreas Mauthe, Ralf Steinmetz, Burkhard Stiller: Token-based Accounting and Distributed Pricing to Introduce Market Mechanisms in a Peer-to-Peer File Sharing Scenario. In Proc. 3rd IEEE International Conference on Peer-to-Peer Computing, Linkoping, Sweden, September 1-3, 2003.

[17] P. Norman, Efficient Mechanisms for Public Goods with Use Exclusions, Review of Economic Studies, forthcoming.

[18] C. Courcoubetis and R. Weber , Asymptotics for Provisioning Problems with a Large Number of Participants,' AUEB Technical Memorandum, 2003. Available at http://nes.aueb.gr/research/.

[19] C. Courcoubetis and R. Weber, Asymptotics for Provisioning Problems of Peering Wireless LANs with a Large Number of Participants. To appear at WiOpt'04 workshop, University of Cambridge, UK, 2004.

[17] P. Antoniadis, C. Courcoubetis and R. Mason. "Comparing Economic Incentives in Peer-to-Peer Networks", *to appear in* Special Issue on Internet Economics, Computer Networks.

[18] P. Antoniadis, C. Courcoubetis, E. Efstathiou, G. Polyzos, and B. Strulo, Peer-to-Peer Wireless LAN Consortia: Modelling and Architecture, Third IEEE International Conference on Peer-to-Peer Computing (P2P 2003), Linkoping, Sweden, 1-3 September 2003.

[19] P. Antoniadis and C. Courcoubetis, Market Models for P2P Content Distribution, International Workshop on Agents and P2P Computing, Bologna, Italy, July 2002.

[20] P. Antoniadis, C. Courcoubetis, E. Efstathiou, G. Polyzos, and B. Strulo, The Case for Peer-to-Peer Wireless LAN Consortia, 12th IST Summit on Mobile and Wireless Communications, Aveiro, Portugal, June 15-18, 2003.

[21] E. Efstathiou and G. Polyzos, A Peer-to-Peer Approach to Wireless LAN Roaming, First ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots (WMASH), San Diego, California, USA, September, 2003.

[22] Thanasis G. Papaioannou and George D. Stamoulis. "Effective Use of Reputation of Peer-to-Peer Environments". In *Proc. of the IEEE/ACM CCGRID 2004* (presented at the *GP2C 2004 Workshop*), Chicago, Illinois, USA, April 2004.

[23] Andy Oram. "Peer-to-Peer : Harnessing the Power of Disruptive Technologies", Chapter 17(Reputation), O'Reilly & Associates, 1st edition (March 15, 2001), ISBN: 059600110X.

MMAPPS Project Deliverables:

[D5] Deliverable 5: Peer-to-Peer Services Architecture [final version], to be delivered.

[D8] Deliverable 8: Results of Peer-to-Peer Market Models, to be delivered.

[D10] Deliverable 10: Specification and Implementation of the Peer-to-Peer Middleware, available on request.

[D12] Deliverable 12: Specification and Implementation of Peer-to-Peer Adaptation Layer, available on request.

[D14] Deliverable 14: Specification and Prototyping of Peer-to-Peer Services, available on request.

[D17] Deliverable 17: Description of Trial Scenarios, available on request.

The MMAPPS Consortium

BT Research, Ipswich, UK (Coordinator)
Athens University of Economics and Business, EL
Eidgenössische Technische Hochschule, Zürich, CH
Darmstadt University of Technology, D
Mysterian, Glasgow, UK
Telekom Austria, Wien, A
University of Lancaster, UK

For more information on the MMAPPS Project, please see:

www.mmapps.org

or contact:

Peter Loosemore
pp MLBG13, Orion Building
Adastral Park, Martlesham Heath
Ipswich, Suffolk,
IP5 3RE, UK
E-mail: peter.h.loosemore@bt.com

Legal Notices

Note that parts of this whitepaper were originally published in P2P Journal, March '04