

Journey to the Edge:

Peer-to-Peer Computing and Content Control

by **Steve Arnold**

President

Arnold Information Technologies

Peer-to-peer computing or P2P, as it is often abbreviated, had a poster child whose impish grin annoyed the music industry. And now Napster may be no more - or at least as its true self. The legacy of Napster, however, remains.



In case you've forgotten already, Napster offered users a free, downloadable program that let people gather MP3 files from computers connected all over the World Wide Web. People would convert CD-Audio tracks to the more compact MP3 format and then post basic information on the file to Napster along with the location and speed of the computer connected to a network carrying the track. Napster users clicked in, found lists of computers carrying their favorite music, picked one, and started downloading. The process of converting a CD-Audio file to the MP3 format is called "ripping." (A moment of levity occurred at the Consumer Electronics show in late 2000 when the president of Intel was firmly corrected by a female engineer who said, "No, Mr. Barrett. It is called ripping a song.")

Well the music industry and the judiciary combined to cure Napster of that little habit. But the idea of grassroots MPS-ripping and distributed indexing have traction.

Currently the post-Napster tool of choice is Aimster. The name Aimster was cobbled from America Online Instant Messenger and Napster. Developed by John Deep of Troy, New York, Aimster software allows AIM users to offer other AIM users a way to locate and copy files on one another's computers. Unlike Napster, AIM users are on one another's buddy lists. The index of files exists on each AIM user's personal computer. When one AIM user wants to copy a file from a "buddy's" computer, the transfer takes place between the two machines.

Aimster has a search function that prowls the directory of the buddies' computers. When it locates the desired file, the transfer takes place. The

speed of the transfer depends on the bandwidth available to the machines. Otherwise, the request and transfer are almost instantaneous. Aimster uses ICQ, a popular messaging client, and AOL's Instant Messenger to detect buddies. In order to prevent an Aimster-type of search-and-retrieve function from working, changes in the architecture of these programs would have to occur or Aimster users would have to be denied access to these popular services.

Aimster eliminates the centralized index model Napster used on its servers. The Aimster service is a true peer-to-peer technology. Napster, like Ray Ozzie's Groove, has a server coordinating certain functions among the peers.

The Threat

By the time you read this, Aimster may have been shut down, because peer-to-peer architectures pose three formidable challenges to users, copyright holders, and organizations.

First, P2P systems are often persistent. As a result, bandwidth and connectivity requirements differ from such common applications as checking e-mail or browsing a handful of Web sites. The data transfer associated with Napster, for example, eroded the performance of some university computing systems. Institutions such as Indiana University (Bloomington, Indiana) had to block university computing users from using Napster in order to restore system performance.

Second, security in P2P systems is a work in progress. Intel Corporation has developed and released to members of its developer community a P2P security toolkit. But the reality of P2P computing is that resources available to members of the community are

subject to the security of each individual machine connected by the P2P system. As they say, a chain is no stronger than its weakest link.

Third, P2P systems are finding their way into the next-generation computing architectures from Microsoft Corporation and Sun Microsystems. Microsoft's Dot-Net initiative allows programmers access to a powerful suite of tools that supports the building of enterprise-class applications using C#, SOAP and XML. Similarly Sun Microsystems's JXTA (Juxtaposition) supports building P2P applications using Java, technology from Infracore [gonesi.com], and Solaris's built-in functions. A good example of next-generation information management tools may be seen by visiting Mirror World, the recent innovation of David Gerlanter, at <http://www.mirrorworld.com>. Using Java, Mirror World's software creates a dynamic, aware "computing space." Content added to a machine in the "space" is indexed and made available to the other machines on the network. A user can perform a search, or a standing profile can alert the user that information of interest has become available.

Managing these three issues promises to challenge the most seasoned information technology manager.

The marketing role of Napster has been significant. Napster was a viral success. In a matter of months, an estimated 60 million people took what amounted to a crash course in the benefits of P2P computing. The demographics of the Napster user hit the sweet spot of wired youth between the ages of 13 and 25. Little wonder that P2P applications are flowing into the market at a lightning clip. Microsoft and Sun are playing a bet-the-farm game with Dot-Net and JXTA. P2P is

Peer-to-Peer Computing

very real, and it promises to become what management consultants call a “structural disconuity” “in the present online information storage, search, and retrieval world.

Napster showed millions worldwide how to share their favorites without fooling around with trips to Tower Records to trade cash for tunes. In retrospect, for those who chose over 55, the furor over Dr. Dre’s mellifluous “Nuthin’ but a ‘G Thang” is amusing, even quaint. Copyright-sensitive information professionals, however, are, shall we say, disquieted.

So Napster has spawned Son of Napster. Remember how in those 1450 Japanese B movies *Mothra*, *Rodan*, and *Godzilla* kept coming back? Each return brought a tougher, nastier beast. The P2P torch has passed to other entrepreneurs such as Mr. Deep and to industry superpowers such as Microsoft and Sun.

Ticket to Ride (The Beatles)

The furor over peer-to-peer computing is one of those peculiarities of technology. P2P has been around a long time, if one views the Internet as an older model of today’s streamlined P2P systems. Telnet can provide some rudimentary P2P functionality, but the command line hardly offers the same allure as a graphical user interface, an agent-based spider, and a chunk of broadband for data transfer.

P2P technologies are often called “edge of network services.” The phrase “edge of the network” can somewhat mislead, but it is catchy. The peer-to-peer architectures move functions once reserved for a centralized server to clients located anywhere and everywhere. Servers — particularly in commercial-grade P2P applications such as Consilient’s knowledge management and collaboration system — are used within the P2P architecture. Edge, however, conveys an image of moving functions from a centralized clutch of servers that are chokepoints for some users’ activities to a borderless and fuzzy anywhere and everywhere of Web services.

The basic idea for peer-to-peer services is that the server is a gateway for the many actions that capable, powerful personal computers can take. The server for some types of applications is little more than an excuse for a systems administrator to become a gatekeeper. Servers have to be watched and maintained by system administrators, whose administrative processes get in the way of what users want to do. Centralized servers give those who control them a considerable degree of power over what users can do and when.

Users of the original Napster taught millions of computer users to turn away from the old-fashioned server set-up. Some enlightened network users recognize that servers do perform some useful functions, but with the increasing power of the basic desktop machine, seemingly ever-increasing bandwidth for those who can afford it, plus a growing array of wireless computing devices, servers and clients no longer offer the most efficient way to deliver one-to-one or one-to-a-community computing. Peer-to-peer architectures are well on the way to becoming the ticket to ride on the Internet in an unencumbered, organic way.

Most peer-to-peer architectures distribute computing tasks across two or more machines. Because peer-to-peer computing is comparatively immature, even the most robust peer-to-peer systems routinely suffer performance and maintenance problems. Peer-to-peer architectures are not perfect. In some peer-to-peer architectures, servers (virtual and real) must be used to provide directory and database services. Napster had, before its death, a server farm reminiscent of the “old” Dialog Information Services’ computer room.

Dot-Net and TXMLA will facilitate the development of many “flavors” of peer-to-peer computing. There is considerable disagreement about the details of how specific services can and should be provided in a particular implementation of a specific peer-to-peer architecture. To its credit, Microsoft has made the C# programming language (a combination of Visual

Basic and C++) and the Simple Object Access Protocol (a mechanism for the exchange of messages and data across and among machines running different operating systems and programs) into “open” technologies. Sun Microsystems, on the other hand, has kept proprietary control of Java. As a result, a battle of the giants is shaping up for 2002.

In the examples of P2P systems included in this essay, a handful of common functions are used within different P2P architectures. Here are what might be thought of as the six building blocks of P2P services:

- *File transfer*: Moving a binary file from Machine A to Machine B without need for a human to do much more than click a button.
- *File sharing*: A set of security bits that tells another system that it is okay to read, change, copy or perform some other function on another machine offering a directory to the public or to users with specific permission.
- *Telnet services*: The basic plumbing of any Internet Protocol function resides in some form in peer-to-peer architectures.
- *Instant messaging*: America Online-type communications with provisions for voice and streaming video.
- *Software transfer and distribution*: Mechanisms to distribute to multiple clients the updates or new builds of the software required to make a peer-to-peer architecture work. New services that support automatic network-based storage and retrieval services increase in importance.
- *Collaboration*: Peer-to-peer networks support shared functions often intended to replicate the features of sitting in a meeting, looking at Power Point slides, and making changes to a spreadsheet so everyone in the meeting can see the numbers ripple.

What makes peer-to-peer applications interesting to many developers is that the network of machines can support agent-based and automated services. In fact, the significance of

Dot-Net lies in the architecture's awareness of events within a network environment. Practical applications of this technology translate to single-logon for a user's services, a way to roll up financial data in a real-time view from within any windows application, and seamless cross-platform access by users to address books, files, and colleagues. If Dot-Net and JXTA perform, a user will be able to begin work on a project at a computer connected to his employer's system, continue work on a wireless PDA, and then output the spreadsheet to a notebook connected to the wireless network at home.

Without a server acting as a traffic cop, intelligent scripts can ferret out information and perform a wide range of tasks. Software can watch for new information and take specific actions when a particular event occurs in a folder on a machine in a peer-to-peer network. For example, a lover of a great "artiste" such as Eminem will be alerted via e-mail when a new Eminem's album is released. The structure of a peer-to-peer network unfolds a rich landscape which can be mined, monitored, and explored in many ways. Individual peers on a network can largely control their own actions.

Each of the major architectures for peer-to-peer networks can be modified. The six building blocks can mix and match to meet specific needs.

Peer-to-peer architectures perform the same type of functions that most users are familiar with. The jump into

the "great beyond" is that peer-to-peer services are distributed over multiple computers unbounded by the reach of the cabling in an office. There are three types of peer-to-peer architectures in use at this time:

- many-to-many computing
- few-to-many computing
- few-to-few peer computing

The terminology is confusing enough to make differentiating them tricky. Very simplistic diagrams of each of the three types of architecture accompany brief descriptions of each of these main types.

The Many-to-Many Architecture

Napster and its variants provide excellent examples of many-to-many computing, the architecture that gave sweet dreams to many music lovers. The architecture is sometimes described as the Gnutella model or public information sharing. Any number of computers can make files available to other users. The database containing information about what files are available to others resides on a database server.

The next-generation peer-to-peer architectures exploit virtual directories and database servers. Because these exist only in memory, shutting next-generation servers may be more difficult. In a peer-to-peer network, a search engine allows any user to locate information on any machine included in the peer-to-peer system. When a match is found,

the user can link to a specific machine and copy the required file or information. An important distinction between Napster and Gnutella is that Gnutella does not have a central directory exactly like Napster's.

Few-to-Many Architecture

Few-to-many computing is cleverly implemented in the SETI@Home system. (SETI is an acronym for the Search for Extraterrestrial Intelligence.) A small number of master servers distribute tasks to computers connected to the Internet. The master servers collect the results of the tasks or problems distributed to the individual PCs. Many PCs work on specific tasks and make the results available to the master servers. This approach is also called distributed computing. Many computers work on problems that would require massive centralized computing resources to handle if the lower-cost, few-to-many model were not available.

Few-to-Few Architecture

Few-to-few peer computing allows users to create a virtual "space" without having to involve a system administrator or invoke complex commands. Any user, regardless of location and type of Internet connection, can enter a secure virtual space and interact. Functions are controlled by the participants. Typical activities include shared Web browsing,

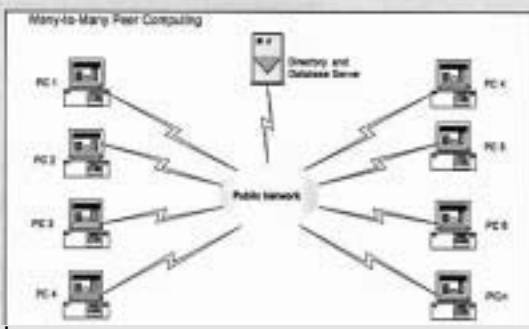


Figure 1 The many-to-many system does a good job of providing access to files on peers. However, when network traffic is heavy, system performance can degrade.

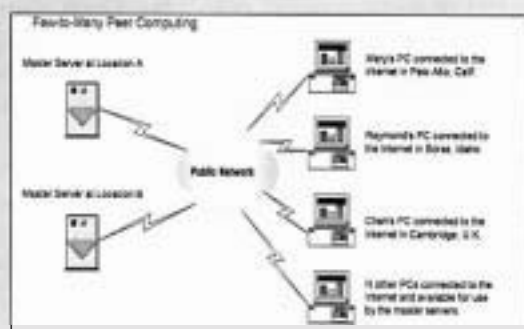


Figure 2 The few-to-many architecture of SETI@Home has inspired a number of commercial enterprises to assemble peers.

Peer-to-Peer Computing

text and rich media messaging, sharing files, etc. Users see and share as if each user were link on a single network in a private virtual space.

Into the Great Wide Open (Tom Petty and the Heartbreakers): Gnutella

Many-to-many computing is the Information Age's equivalent of Spanish conquistadors standing on the shores of the New World - an earlier period's "great wide open." The opportunities are limitless and the resistance mounted by the indigenous people ineffective.

Peer-to-peer computing is personified in the media by one Gene Kan, whom we will encounter several times in the following pages. He has been one of the most visible drivers behind the Gnutella portal [<http://gnutella.wego.com>]. But the motive force of Gnutella was Justin Frankel, the programmer who created winamp, one of the First and a hugely popular "skinable" program for listening to music via the Internet. Winamp gave users unprecedented control over what to play and what type of interface best suited each user's personal tastes. The 'skins' available for Winamp can make the Winamp interface look like a zebra-skinned settee or a control panel from the starship Enterprise.

Mr. Frankel sold his company to America Online (soon to acquire EMI and Warner recording properties). While an employee of AOL, Frankel and a handful of programmers polished Gnutella in their spare time. Without alerting AOL executives, Mr. Frankel released the program as open source code. Once released, anyone could download the code, make changes, and use the software to facilitate many-to-many online services.

Mr. Frankel seemed unconcerned about the machinations of corporate America in general and AOL in particular. He soon found himself the focal point of AOL corporate scrutiny. America Online yanked Mr. Frankel's

Nullsoft Web page that provided access to the Gnutella source code. But ... the action was a textbook case of locking the digital barn after the digital horse galloped into the great wide open Internet.

Almost immediately, other programmers took Mr. Frankel's packet of code, disassembled it, rebuilt it, and in a matter of months, made available the original code base plus six or seven important variants by the summer of 2000. Today one can see Gnutella as one of those programs playing an historical role similar to that of wordstar in word processing or Lotus 1-2-3 in spreadsheets. Although better programs come along, Wordstar, Lotus 1-2-3, and Gnutella have changed behavior.

Anyone wanting to use Gnutella must download software. For example, Gn00a can be used as it is. For the adventuresome and Big Music's squads of programmers, download Gn00b, a version that one can use to build a purpose-built Gnutella application. [For copies of these programs, go to The Gnutella Developer home page, <http://gnutelladev.wego.com/>, managed by Gene Kan.]

A Gnutella application plays a dual role. Once a user installs Gnutella, the personal computer is both a client responding to requests from others in the Gnutella "space" and a server providing services to other users also in the "space." (A "space" is the vimal meeting room in which a group of users gather.) The essence of Gnutella's many-to-many model lies in the fact that it does not need a separate server for most functions. Peers provide the types of services normally assigned to a server in a traditional client-server network.

The Gnutella space is a virtual environment in which peers can interact as if working on one system. Anyone with the Gnutella software running is a peer in this type of architecture. Security and access are not features implemented with the type of stringency one would find in a corporate network or in America Online's Instant Messenger environment. There are some similarities to the "party line" tele-

phone connection. Anyone on the line can talk and exchange information.

After installing the software on a personal computer, the user connects to hosts that act as entrance points to Gnutella peers. The hosts usually provide only the IP address and port data and do not share files. One can locate Gnutella hosts by clicking to such portals as Gnutelliams and looking at the list of gateways now online. Gnutelliams [<http://www.gnutelliams.com>] is a useful Gnutella portal offering a directory of Gnutella client downloads for Windows, Linux/Unix, Java, and Macintosh.

Because there is no central database containing look-up tables for mapping users to data, Gnutella affords some user anonymity. The GnuteUa software implements a procedure to tokenize and then hide the identity of a user generating a query.

When a connection is established, machines on the network communicate via messages. Each peer receives and sends messages. Gnutella supports a number of standard Internet and special purpose messages, including the following:

- Internet pings that ask, "Are you online?-"
- "Pongs" or messages that say, "Ready to send and receive." A pong contains the IP address of the ponger, the port to use, and data about the files available to peers. Peers forward pongs in order to allow others in the space to locate machines and files.
- Queries are messages that say, "I am looking for Mozart files." These are forwarded to all the machines in the space. Each query has a unique identifier, but the originator of the query is not included in the message.
- Answer messages are replies to queries. These messages include the IP address, port, and any other information required to complete the file transfer. These responses include a unique identification string associated with the replying peer. Answers do not broadcast to all peers in the space. Instead, the answer travels backwards along the

path the original query took to reach a specific peer. It is, therefore, nearly impossible to trace query responses in a Gnutella system.

- Get and push messages request file transfer from the peer with the data to the peer wanting the data. When a peer with the requested data lies behind a firewall, Gnutella allows that information to be pushed through the firewall to the requesting peer. This is necessary because firewalls are set up to prevent certain types of file transfers. In both the peer requesting data and the peer providing data are behind firewalls, the file transfer is not possible and the user is out of luck.

In order to keep the Gnutella space from flooding available bandwidth with infinite rebroadcasts of messages, time-out mechanisms are baked into Gnutella. Once sent, each message has a time-to-live (TTL) function which decrements to zero. A zero TTL value causes the message to die.

At the same time as university programming classes were digging into Gnutella and many-to-many applications in class and in extracurricular hack sessions, the savvy Gene Kan emerged as the spokesperson for the Gnutella movement. In March 2001, he sold his InfraSearch technology to Sun Microsystems. (The splash page for the now discontinued service is shown in Figure 4 below.) InfraSearch

will reemerge sometime in the coming months as a key component in JXTA, Sun Microsystems P2P system.

As president of XCF Ventures, Mr. Kan has been tapped by the media as the principal spokesperson for Gnutella. He has a high profile and presents the polished presence necessary to navigate in the treacherous straits between hostile legal forces and paranoiac multinational content companies. Gnutella variants are used by thousands of Internet users to locate and download pirated software and to digitize motion pictures, pornography, and, of course, MP3 files. To get a feel for the traffic in Gnutella-fueled services, query newsgroups for the string "warez" and follow a handful of postings.

The many-to-many environment demonstrates asymmetrical sharing. Although there are thousands or millions of people on a system, the majority of the content comes from a small percentage of users. According to Eytan Adar and Bernardo A. Huberman, "almost 70 percent of Gnutella users share no files, and nearly 50 percent of all responses are returned by the top 1 percent of sharing hosts. Furthermore, we found out that free riding is distributed evenly between domains, so that no one group contributes significantly more than others, and that peers that volunteer to share files are not necessarily those who have desirable ones" [Eytan Adar and Bernardo A. Huber-

man, Free Riding on Gnutella, <http://www.firstmonday.dk/issue/issue5-10/adar/index.html>, October 2000].

A glimpse at the future of Gnutella sparkles off the slick exterior of the most recent release of Bear Share, now in version 2.2.0. This Gnutella-based application comes from Free Peers, Inc., of which its marketing collateral says, "Bear Share provides a simple, easy-to-use interface combined with a powerful connection and search engine that puts thousands of different files in easy reach." [For a copy of the software, click to <http://www.bearshare.com>. For gateway information, click to <http://wimw.bearshare.net>.]

The most recent build provided the screen shot of the initial connection screen. Once a user locates a host, tabs provide point-and-click access to upload, search, and monitor functions.

The latest release of the program provides greater code stability, support for multiple active searches, active monitoring and updating of shared directories, and reverse lookup for computer addresses, among others. The Florida-based company generates revenue from advertising and the operation of various hosting services.

Gnutella improved on the Napster model by substantially eliminating the centralized server that made Napster vulnerable to its critics. Many-to-many applications work when large numbers of people participate. The

#-

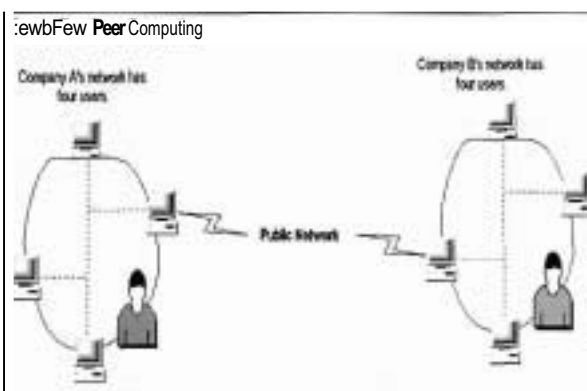


Figure 3. Few-to-few architectures can provide useful, task-oriented services without the cost and complexity of a Lotus NoteVDomino installation.



Figure 4. The InfraSearch service was online for a short time. A user entered a search word or words and the results from geographically separate Sewers were displayed.

Peer-to-Peer Computing

more data and people using the system, the greater the likelihood that the information one wants will be available from someone on the network. But success has a price.

The number of users on the system creates two problems. First, many-to-many systems are sluggish. With the increased number of users, visibility and buzz increase. The heightened visibility and the greater information available increase scrutiny of the content on the system. And, of course, copyright is the main pain of contention in Gnutella-type systems.

Content company executives may want to ponder the words of Rob Lord, one of the programmers who developed Gnutella, “we didn’t get into this ‘space’ cuz we’re internet gold-seeking cockos. We’re legitimate nihilistic media terrorists as history will no doubt canonize us” [<http://www.nullsoft.com/>, retrieved on March 14, 2001]. [From this site one may download BlorpScript, a GPL’d PHP imagelcontent browsing system” and Amazing, a nifty Winamp visualization plug-in.]

Just Can’t Get Enough (New Radicals): Low-Cost Supercomputing

At any one time, millions of computers sit idle, even while connected

to the Internet. Why not harness the computational capabilities of these computers, creating one of the world’s largest and least-expensive supercomputers? This is the question that Berkeley computer scientists, among others, asked and answered. Distributed computing is the process of linking computers together over the Internet and putting their idle processing power to work to create a “virtual supercomputer.” Many computer scientists cannot get enough computing horsepower to do their work. Using a peer-to-peer architecture offers one way to get access to calculating and storage resources.

The most visible outcome of a scientist’s need for inexhaustible computational horsepower is SETI or Search for Extraterrestrial Intelligence. [For more information, go to <http://setiathome.ssl.berkeley.edu/>]

The SETI Institute serves as an institutional home for scientific and educational projects relevant to the nature, distribution, and prevalence of life in The universe. The Institute conducts and encourages research and related activities in a large number of fields, including, but not limited to, all science and technology aspects of astronomy and the planetary sciences, chemical evolution, the origin of life, biological evolution, and cultural evo-

lution. The SETI@Home project is not directly sponsored by the SETI Institute. But the Institute has benefited from the publicity generated by the SETI@Home project.

The SETI@Home project allows individuals to make their computer available for crunching the data returned from monitoring devices. A user downloads a software bundled that installs as a screen saver on the individual’s computer. Participants include individuals, academic institutions, and various corporations, including Lehman Brothers, Alcatel, the French telephony company, and Booz, Allen & Hamilton, the upscale consulting outfit.

Once a personal computer is set up to participate in SETI@Home, servers at Berkeley copy a chunk of data collected with the Arecibo Radio Telescope in Puerto Rico as part of Project SERENDIP. The SETI Institute is a major supporter of the SERENDIP search.

The idea behind SETI@Home is to take advantage of the unused processing cycles of personal computers. When the participants’ computers are idle, the SETI@Home applet downloads a 300 kilobyte chunk of SERENDIP data for analysis.

The results of this analysis are automatically sent back to the SERENDIP team and combined with crunched data from SETI@Home participants in more than 200 countries. In the 2 years of SETI@Home operation, no sign of extraterrestrial life has been found. However, the distributed system has received about 300 million results from users around the world.

The limitations of this system are that only a 2.5 megahertz piece of the observed spectrum is analyzed by SETI@Home. The data processing does not occur real time so that interesting signals are flagged and then analyzed further by project engineers.

The advantage of this scheme is that it permits looking for a variety of signal types that the current SERENDIP lacks the processing capacity to analyze.

Other academics have tapped distributed computing to tackle other computationally intense problems, in-

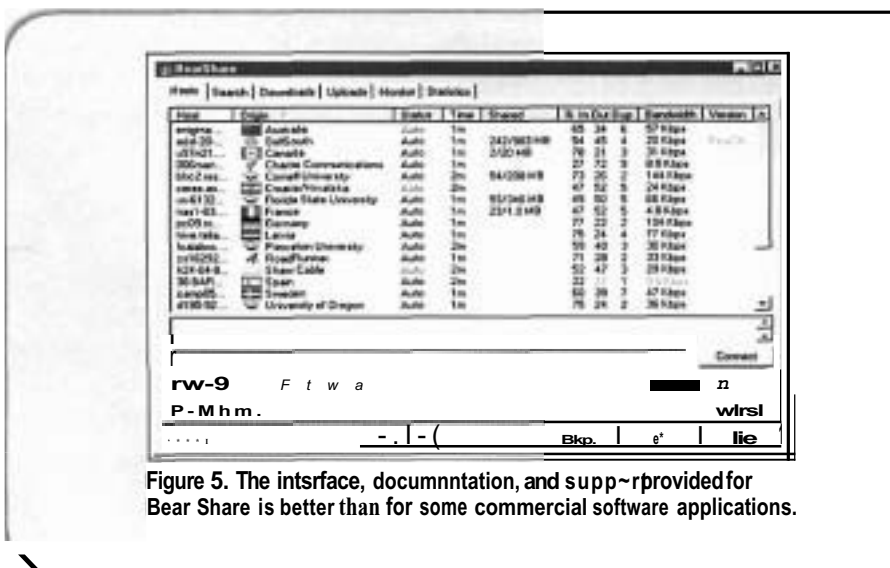


Figure 5. The interface, documentation, and support provided for Bear Share is better than for some commercial software applications.

cluding prime numbers, Fermat numbers, and optimal Golomb rulers. An organization called Distributed.net has focused on cryptography since 1997.

There are numerous commercial enterprises developing or offering products and services for few-to-many architectures :

- Applied Meta Computing [<http://www.appliedmeta.com>] has government customers such as NASA and the Defense Department as well as clients from the Fortune 500. The company has received what it calls “a significant investment” from PQ-Laris Venture Partners, an early-stage venture firm with over \$1.2 billion under management.
- Data Synapse [<http://www.datasyapse.com>] offer its Web Proc product to financial institutions, banks, broker deals, and others with high-demand computational requirements.
- Distributed Science [<http://www.distributedscience.com>] arose from a merger of DCypher.Net into Distributed Science. The company provides both computational and network storage services. The company aggregates excess computational and storage capacity and sells it. Distributed Science’s customers do not have to make massive investments in additional hardware. At the end of 2000, Distributed Science said it had signed up 145,000 personal computer users with more than half (about 72,000) from outside the United States. Each computer user makes a personal computer’s idle time available to Distributed Science. [For more information, read Bob Savage’s “Distributed Science Spreading the PC,” Bob Savage, Local Business.com, http://www.localbusiness.com/Story/0,1118,LAX_537355,00.html, December 13, 2000.]
- Entropia [<http://www.entropia.com>], another distributed computing startup, obtained \$7 million in funding from Mission Ventures and Silicon Valley Bank’s San Diego Technology Group. Entropia focuses on enterprise-distributed computing.

The firm licenses its Entropia 2000 Enterprise Server. The software enables customers to get the same power and scalability of its Internet technology on local and wide area intranet networks. Entropia offers application software integration services.

- Parabon Computation [<http://www.parabon.com>] offers the Frontier distributed computing product and for-fee services aimed at biotechnology, financial, and pharmacology research. A person wanting to participate in a Parabon project can download the Pioneer applet. Parabon pays individuals to participate in its projects.
- Popular Power [<http://www.popularpower.com>] has started paying people for making their personal computers available for influenza virus research. Brian Behlendorf, a co-founder of the open-source Apache Web server project, is one of the investors in the company.
- United Devices [<http://www.uniteddevices.com/home.html>] started by SETI@Home founder David Anderson, secured \$13 million in venture capital funding from Softbank Venture Capital, Oak Investment Partners, and others. The company’s major customer is Exodus Communications. United Devices provides software tools to allow those required distributed networks to build, enable, deploy, and support an Internet-distributed computing project.

For enterprises looking for a low-cost way to build a distributed computing application, TurboLinux sells EnFuzion for about \$400. EnFuzion lets corporations farm out computing tasks across its computers using any operating system. Companies exploring distributed applications include JP. Morgan, Procter & Gamble, and Motorola.

The applications for few-to-many computing focus on computational or storage intensive tasks. This technology has some interesting commercial applications. The principal benefit

of the technology is reducing the capital cost associated with adding specialized computers and storage systems. Costs for system upgrades cannot easily be driven to zero, but more efficient use of computing resources is a primary benefit of distributed architectures. The downside is that optimizing distributed system performance remains a challenging task. Schedulers and caching systems offer some performance improvements, but the field is comparatively new. Distributed computing has tremendous potential in engineering, scientific, and research applications.

Chemicals Between Us (Bush): Groove.Net

Few-to-few peer computing involves what one might call a “hybrid architecture.” It blends one or more virtual servers that can perform security and performance functions with Gnutella-type peer-to-peer computing. The potential of distributed computing can be tapped if those participating require access to specific computational tasks or need specialized storage services greater than any single user’s hard drive capacity. It allows the fluidity and openness of the many-to-many architectures but in a more controlled, tightly structured environment, one that can support a group of people working on a project, bonding and interacting in a secure virtual space.

The principal problem with the many-to-many model is security—or lack of it. The hybrid architecture allows log in, authorization, usage tracking, and other services to be handled by a virtual server running necessary authorization services. Once a user is logged in, the few-to-few peer network looks and feels like a Gnutella session or an America Online chat room.

The leader of the few-to-few architecture is Ray Ozzie, the developer of Lotus Notes. After leaving IBM, Mr. Ozzie conceptualized a software environment that would exploit Internet

Peer-to-Peer Computing

services, yet provide the specialized functionality that enterprises and commercial organizations require.

Mr. Ozzie and his team worked for about 3 years to get Groove to the market. Groove [<http://www.groove-networks.com>, <http://www.groove.net>] feels like a house decorated in Early American and Modern furniture. Handled with restraint and good taste, the mix is at once familiar and trendy. Groove has Notes-like functions, but it has the throttle touch of a Gnutella applet.

Groove is a system that brings together people, messages, discussions, documents, and applications in the context of a shared project or activity. Groove includes a range of collaborative tools, including a threaded discussion, chat, a notepad, a sketchpad, a calendar, and a file archive.

One can emulate Groove functions to a degree by cobbling together a suite of applications for a Notes and Domino environment. One can provide similar functionality by integrating Internet-accessible services from such companies as vJungle, building the missing bits from vJungle's Open& tools. Groove is one of the first companies to offer the hybrid service with security and control features baked in, not left out or glued on as an afterthought.

The Groove server is actually a bundle of code called the "Groove Relay."

No single machine performs dedicated Relay functions. Instead, each peer executes Relay functions as required.

Purists may argue that Mr. Ozzie has not created a hybrid peer-to-peer architecture. Those concerned with security should understand that the implementation of a virtual Relay server solves, in part, many of the security challenges associated with Gnutella, for example. Groove can be configured to integrate with users' existing directory and certification systems.

Groove users, even though invited to a shared space with security services running, can delete information from a shared space. They can invite new users into a shared space, opening up communications to new sets of eyes that the original community members may not have expected. They could even attach a file such as a ".exe" file with a virus using the Groove files tool. Alternatively, a user could attach a file containing a virus into the files tool. In this case, the virus would not fire until someone launched the attachment. In that case, the virus could affect the Groove user's personal computer.

Groove supports spontaneous collaboration. There is none of the complex set up required for audio conferencing needed. Groove accommodates users on local area networks behind different corporate firewalls, on dial-up

connections with dynamic IP addresses, behind cable-modems, and so on. Regardless of connection, a Groove user can create a shared space and invite other Groove users onto it to work on a project.

When Groove members create information with Groove tools, the data resides on the machines of all invited group members. The data are synchronized when a user or a group of users makes changes. The biggest drawback to Groove is the quirky terminology the company uses. Instead of "voice chat," Groove prefers "transceiver." Groove's peculiar jargon impedes a new user's understanding of the services and features.

Invitations to join a "space" are issued using messages. A person invited to a Groove space must install the Groove application. Once installed, the Groove applet can copy appropriate data to the new user's machine. Each Groove member receives a complete copy of the shared-space data. The copying takes place in background.

The shared space data for a particular Groove collaboration provides a context for the messages and data readily available to any user. This differs from the contextless snippets of electronic mail that often require one or more rounds of telephone tag to resolve and/or considerable inductive effort. Anyone in a Groove shared space has the same information and data on view as other members. The Groove shared space formed in this manner is secure. Each Groove space can be extended at any time to any person with whom members need to collaborate. No system administrator or coding is required.

The Groove relay brokers connections for devices behind firewalls and devices that talk with Network Address Translation conventions. Groove provides store-and-forward service so shared-space members can go offline, then reconnect, and be resynchronized automatically.

Groove is a new type of product. Like any new network-centric tool, users must embrace it. Enterprise pricing for



Groove 1.1 is \$49 per user. Pricing for individual purchase of Groove will be announced later this year. As of August 2001, Groove Preview Edition, a subset of the full Groove product, is available at no cost for personal use or business trial. Groove may be a harbinger of how Internet operating systems will work.

Alternatives to Groove will pop up with increasing frequency in the months ahead. The impact of Gene Kan and InfraSearch on Sun Microsystems will almost certainly lead to a product or toolset that supports Groove-type services. Java and the Solaris operating system are attractive conduits for secure, collaborative peer-to-peer applications.

Microsoft's Sharepoint Portal Server promises to deliver much of the same functionality for Windows 2000 environments. Coupled with Microsoft's "Dot-Net" initiative, the millions of programmers familiar with Microsoft languages will be able to snap together peer-to-peer applications by dragging and dropping widgets from a toolbar.

The impetus behind this new type of peer-to-peer service is the human need to communicate with the Internet packet as the digital morpheme. One can see mobile devices, notebook computers, and deskbound professionals entering virtual, secure spaces for a wide range of activities.

Living on the Edge (Aerosmith)

Peer-to-peer computing is an essential component of the pervasive network. Each of the principal peer-to-peer architectures summarized in this article can support a wide range of applications. What is the impact of peer-to-peer computing on information companies producing electronic content?

First, many information companies will make extensive use of peer-to-peer technology in their own organizations. The collaborative power of the few-to-few architecture and the potential cost savings of the few-to-many

architecture are easy to explain in terms of costs and benefits. As pervasive computing becomes easier and always-on connections become a reality in major cities, peer-to-peer functions slipstream easily into many workflows. For example, a discussion about a marketing campaign or a new product development project can exercise peer-to-peer systems in a comfortable, intuitive way. It makes sense to share information among team members. It makes sense to perform complex computing and storage tasks using all the available resources as efficiently as possible.

Second, peer-to-peer computing will accelerate the development of new types of information products and services. It is not clear if the drivers of innovation will be the established firms or startups such as Napster. Digitized video, digitized motion pictures, and digitized anything lend themselves to systems similar to those of Napster and Gnutella. This fact has not been lost on legions of programmers under the age of 13 who see no good reason not to make it possible to share games, images, books, videos, and music. Portals, print magazines, books, and entirely new types of search-and-retrieval mechanisms are already flowing into the market space blasted into Internet users' consciousness with Napster.

Third, commercial enterprises face two short-term tasks.

Job One for commercial information operations is to understand, monitor, and figure out how to deal with a continuing flow of software that can sidestep digital rights and for-fee distribution systems. The task will be neither easy nor rewarding. Large publishing companies are poorly equipped to deal with the technical innovations of a high school student in Boise, Idaho, or a college student in Osaka, Japan.

Job Two is to recognize that peer-to-peer technology offers professionals and consumers new ways to integrate

content into their everyday activities. My term for this new publishing model is "in-phase distribution." Peer-to-peer technology, combined with pervasive network connections, allows an individual to access facts, numeric data, and information at the precise moment the data is required.

"In-phase" information implies a much closer blend of predictive statistics, software agents, and event triggers. A doctor making rounds wants access to colleagues knowledgeable about a particular patient. At the precise moment information is needed about drug interaction, the physician will be able to shift from data collection, to collaboration, to retrieving specific information about what medication change to make. Similar opportunities will exist in sales, legal, technical, and financial arenas. And, of course, intermediated searching may rise again using this new technology.

In-phase communications means the delivery of access when it is needed and in the format required by a specific work or leisure context. This points to new revenue opportunities. Napster demonstrated that peer-to-peer technology can galvanize an industry largely unchanged by the advent of the Internet. After the recording industry, peer-to-peer technology will make its impact felt upon motion pictures, professional publishing, and anywhere content and needs can be matched in real time.

The edge of the network is fast becoming the center of content delivery innovation. For many information innovators under the age of 25, Journey's song "Don't Stop Believing" is the perfect background to inventing a new information distribution system. *

Mr. Arnold is an information consultant. This article is derived from a chapter in his most recent book, *The New Trajectory of the Internet*, published in May 2001 by Infonortics, Ltd. He resides with 13 servers and two boxers in rural Kentucky.