Under the label of peer-to-peer (P2P), developers are creating new kinds of distributed services that thrive on the Net. P2P services connect computers directly as equals, in contrast to the centralized models of mainframes and client-server. Though Napster kindled the excitement around P2P, the real story goes far beyond sharing music files. The Internet “cloud” is becoming intelligent. P2P architectures promise greater flexibility and scalability, as well as new applications that tap the latent power of millions of connected devices. The hype is intense . . . but underneath it lies much that is real and sustainable.

P2P draws on well-developed distributed-computing and peer-based communication models. Only today, there is enough bandwidth, processing power and storage cheaply available to make these approaches viable for many real-world tasks. So many things change in a world of ubiquitous connectivity. Some problems become harder, while others are simplified. Software that lives on the Net must work like the Net, in all its ragged glory. It must thrive on chaos and uncertainty, take advantage of network effects and give each user the power of an administrator. P2P companies are now building that kind of software.

P2P applications rely on intelligent and independent peers at the edge of the network to take the place of central servers. Napster, for example, lets users transfer MP3 files directly from hard drive to hard drive, instead of downloading them from a Website. P2P links together three distinct classes of applications that already existed by the end of 1999: distributed file sharing (Napster, Gnutella and Freenet), instant messaging (ICQ, AOL Instant Messenger) and distributed processing (SETI@Home, Distributed.net). Technology is
evoinding in each of these areas, but new categories of P2P applica-
tions are also developing. The P2P space is growing up, from non-
commercial efforts and creative hacks to thriving open-source
communities and robust platforms.

Napster and Gnutella raise fascinating questions about intellectual
property and the future of digital content, but that alone won't
make P2P approaches commercially successful. Companies building
P2P applications must articulate some reason other than coolness
for customers to buy their products and services. The smart ones
are starting to do so; their effectiveness remains to be seen.

Yet P2P may take hold as an industry-changing model even if the
startups in the field today fail. The P2P awakening is part of the
Net's evolution beyond the Web - designed for sharing information
- to shared applications and transactions. Services such as instant
messaging were hugely successful before anyone thought to label
them P2P or analyze their architecture, because they addressed real
needs. The true potential of P2P lies in solving hard problems cre-
atively, employing servers when it makes sense while using distrib-
uted architectures to scale and give users new capabilities.

P2P is a state of mind, a way of approaching challenges as much as a
technical architecture. That's one reason it's maddeningly difficult
to pin down. Philip Bernosky of Intel's business vision group puts it
well: "Grammatically, P2P is an adjective. People are trying to make
it a noun and finding it's more difficult than it seems." To thrive,
P2P-oriented companies will have to listen to the market as it catch-
es up with them. P2P will have truly succeeded when, as with client-
server and the Internet protocol, the focus shifts to building
solutions on top of the new foundation.

The P2P story is interesting from so many angles that we've split it into
two issues. This month we consider the basic value proposition of P2P
and examine the post-Napster generation of companies building peer-
based applications. Next month we'll dive deeper below the surface and
explore distributed computation, P2P infrastructure, organizational/
market dynamics and business models.
P2P v2

My, how you've grown!

We discussed decentralized, post client-server computing in April under the rubric of “data soup,” (see RELEASE 1.0, APRIL 2000). At that point, there were hardly any visible P2P companies other than Napster, which was still an angel-funded startup developing its business strategy and filling out its management team.

Since then, many things have changed, but some important ones haven't. As we noted in the spring, the key to the emerging new architectures is that they collapse the arbitrary distinctions between clients and servers, and between local data stores and remote repositories. Whether a service is fully decentralized or, like Napster, employs a central server for some functions matters little to the rest of the world. What's important is that applications work, scale, provide value to their users and take advantage of the vast computing resources available in a fundamentally connected world.

The data soup notion captures the user’s view of information and resources in a P2P world. Similarly, syndication (see RELEASE 1.0, JULY/AUGUST 1999) describes the fluid and interconnected business relationships that can emerge when every market participant is potentially an originator, aggregator and distributor at the same time. There is also story from the developer side. P2P architectures promise to solve some difficult technical problems efficiently and elegantly. In a world where scaling across an ever-growing and diversifying Internet is a constant challenge, P2P techniques have become a valuable new tool.
The new P2P companies use decentralized architectures to address large-scale business and consumer needs such as information discovery, collaboration and content distribution. The most robust of these platforms have been in development for some time (three years in the case of Groove; see page 13). Napster energized developers and jacked up awareness among users and financial markets, but most of the underlying engineering principles of P2P applications have been floating around for years.

Software wars: a new hope?
The current P2P foment immediately calls to mind the environment in 1993-1994, when the Mosaic browser catalyzed the commercial growth of the Web. It was clear to everyone involved at the time (admittedly a much smaller group than today) that Mosaic heralded something big. It just wasn’t clear exactly what that was.

In the years since, other phenomena such as Web-based email and instant messaging (see release 1.0, June 1997) have enjoyed similarly explosive adoption rates. Each wave leverages previous ones: Near-ubiquitous connectivity in many countries provides a massive distribution platform for new technologies; communications tools allow users and developers to interact and collaborate freely; and the open-source paradigm fosters rapid software innovation beyond the company boundaries.

Numbers depend on how you define the term, but P2P news and information site www.peerprofits.com listed 101 P2P companies as of early November. Big companies such as Intel, founder of the Peer-to-Peer Working Group, are talking up P2P and devoting resources to support further developments. (The first meeting of the working group drew over 300 attendees.) Intel has reviewed over 80 P2P business plans for possible investments, while Draper Fisher Jurvetson claims to have seen 200.

Peer today, gone tomorrow?
Indeed, there is already a backlash against the P2P movement. Sensing interest from the media and the venture capital community, many vendors have simply re-labeled themselves as P2P players. As we discuss below (see page 25), major questions remain about how successful any of the P2P applications will be, both in terms of technical efficacy and business viability. Even after its recent deal with Bertelsmann (see page 9), Napster remains under a legal cloud and has yet to demonstrate it can produce revenues from its huge user base.
Peer-based communication between network nodes is really part of a larger phenomenon of decentralization. And decentralization itself is both a cause of and a response to complexity.

From 30,000 feet, most of the services P2P companies are offering are already available using centralized architectures. Napster was hardly the first online music distribution service, for example. What, then, is all the fuss about?

First, it’s important to distinguish P2P infrastructure from P2P applications. The Internet has always been decentralized. Arguably, P2P is the essence of the Net. Paul Baran developed packet-switching partly to eliminate central failure points during a military attack. Every router moves traffic independently, with only a local map of network topology. End-to-end system design, which has allowed new services to flourish on the Net (see RELEASE 1.0, FEBRUARY 1999), treats every point on the network as directly connected, without worrying about what’s in between them.

There are similar examples in other network domains. The Macintosh operating system, for example, has treated peripheral devices as peers since its introduction over 16 years ago. Sun’s Jini technology creates peer-based connections between heterogeneous network devices, even to the point of assembling virtual computers out of loosely coupled components. Above the infrastructure, however, applications such as the Web distinguish clients from servers.

Second, there’s a difference within applications between physical and logical P2P. Internet email, for example, is physically deployed in a centralized manner, with messages routed through central mail servers on both ends of the communication. Users, though, see a message move directly between themselves and the recipient, in P2P fashion. Most instant messaging services (and Napster) use central directory servers to set up P2P physical connections between users, who then communicate directly.

**Mere anarchy is loosed upon the world**

Peer-based communication between network nodes is really part of a larger phenomenon of decentralization. And decentralization itself is both a cause of and a response to complexity.
Throughout the history of computing, the centrifugal force of decentralization has battled the centripetal force of central control. As any physics textbook will tell you, though you feel outward-pulling centrifugal force, only centripetal force is real. Similarly, though decentralization and user empowerment are powerful ideals, centralization efforts are usually driven by concrete realities. Engineers may advocate centralization to maximize control or to guarantee performance. Or business and policy imperatives may push network designers to emphasize central servers.

The Net changes the equilibrium. As it races to every corner of the Earth and extends beyond PCs to mobile devices, set-top boxes and remote sensors, the Net is rapidly becoming a laboratory for testing the limits of centralized control structures. Engineers managing the server farms behind Yahoo!, Hotmail and eBay face similar challenges to those faced by central planners in the Soviet Union. Keeping up with growth and managing infrastructure across an increasingly complicated and heterogeneous Internet is no small challenge.

Scientists have demonstrated that complex adaptive systems can self-organize and generate stability far beyond what centralized or top-down approaches could achieve (see RELEASE 1.0, JUNE 1989). As the Net has proven, redundant unreliable local connections can out-perform globally managed network infrastructure. P2P is a way to turn the complexity of the network from a challenge into an opportunity, by taking advantage of bottom-up self-organization. (David Gelernter’s essay, “The Second Coming,” is a thoughtful meditation on these themes, not specifically focused on P2P. See Resources, page 28.)

As Steve Burbeck of IBM’s software group notes, it’s a rule of thumb in complexity theory that when the complexity of a system increases two to three orders of magnitude, some qualitatively different structure emerges. With the growing power and storage capacity of PCs, the growing number of users on the Internet and the growing size of objects moving across the networks, he ventures, “In some sense this had to happen.”

**Theory into practice**

Self-organization is neither automatic nor guaranteed; many interacting systems are either static or chaotic. And decentralization doesn’t mean that central authorities are never the most efficient local control mechanisms. P2P aspirants face the challenge of tuning local rules and functions so that systems do self-organize. Evolutionary and biological metaphors are prevalent. Says Miko Matsumura, ceo of
P2P content-delivery startup Kalepa: “The big question is whether there is enough organizing matter for data to achieve a primordial soup-like quality.” As with biological systems, the goal is to build reliable wholes out of unreliable parts.

The intellectual underpinnings of the P2P model have been around for a long time, though not implemented in large commercial networks. For example, Groove Networks ceo Ray Ozzie (see page 13), points to papers published over 30 years ago by pioneers such as J.R. Licklider and Douglas Engelbart that described architectures similar to what we now call P2P. “My job was to bring human-computer interaction into the 1960s,” Ozzie quips.

To put it another way, P2P companies must turn theory into practice. That means systems that scale in the real world, overcome the idiosyncrasies of hardware and of network administrators, preserve security, build trust, create a compelling user experience and add value in a way someone is willing to pay for. Less exciting than proclaiming the end of the world as we know it, perhaps, but more important in the long run. As Infrasearch ceo Gene Kan (see page 20) points out, these days “you don’t look at an application and say, ‘Oh, that’s client-server!’ You just look at it and see that it works.”

**P2P Taxonomy**

Though its roots are decades old, Internet-based P2P computing is a new world. Traditional application categories aren’t necessarily useful in understanding it, just as labeling a Web browser as a file-transfer or multimedia tool isn’t very informative.

Moreover, P2P thinking is disorienting. The concept of end-user machines at the edge talking to a server in the middle gives you a firm sense of where you stand. P2P is not just decentralizing, it’s decentering in the post-structuralist sense. People become the new core of the network, but people are complex and deeply enmeshed in relationships. They move from place to place, task to task, group to group and machine to machine. You can identify the physical nodes in a P2P service, but you can’t pin down its deep structure the way you can with centralized applications.

The first challenge for any observer is to distill the essence of P2P-ness. The various applications we describe all: devolve significant autonomy and control to independent nodes; capitalize on under-utilized network-connected computing resources at
The growth of P2P services will be retarded if this world fragments into warring proprietary platforms, forcing users to make unpalatable choices and killing synergistic network effects. Some existing proposed standards fit naturally into P2P models, including simple object access protocol (SOAP) and universal discovery description and integration (UDDI). Things are still developing too quickly, but at some point it will make sense to have at least de facto standards for common P2P elements.

One possible set of de facto standards are the protocols in Napster. With almost 40 million copies in circulation, it may be the most ubiquitous piece of non-Microsoft desktop software ever. Others have reverse-engineered Napster’s message format, with informal assistance from employees at Napster itself. No commercial developer will launch a product interoperating with Napster, however, so long as the record industry’s lawsuit hangs over the company.

Standards bodies also provide a place for industry participants to gather, compare notes, identify shared challenges and find common ground. Such informal discussions are just as important to the growth of new areas as the formal standards processes. That’s especially true when, as with P2P, companies don’t even have common terms to express what they’re doing.

There have been two high-profile efforts to create such a “shared space” for the P2P world. Intel began the peer-to-peer working group to promote the concept of P2P computing and to foster a stronger developer community. The first meeting, in October, drew a huge crowd. Unfortunately for Intel, many of the participants took issue with Intel’s proposed structure for the group, ironically a hierarchical model with a steering committee of paying corporate members. Intel has said it will rethink the structure, though no alternative has yet emerged.

The other process was launched by Tim O’Reilly of O’Reilly and Associates (see RELEASE 1.0, NOVEMBER 1998). O’Reilly convened a meeting of about 20 P2P thought leaders in September to start a dialogue, and he is planning a P2P conference for February.

The growth of P2P services will be retarded if this world fragments into warring proprietary platforms, forcing users to make unpalatable choices and killing synergistic network effects. Some existing proposed standards fit naturally into P2P models, including simple object access protocol (SOAP) and universal discovery description and integration (UDDI). Things are still developing too quickly, but at some point it will make sense to have at least de facto standards for common P2P elements.

One possible set of de facto standards are the protocols in Napster. With almost 40 million copies in circulation, it may be the most ubiquitous piece of non-Microsoft desktop software ever. Others have reverse-engineered Napster’s message format, with informal assistance from employees at Napster itself. No commercial developer will launch a product interoperating with Napster, however, so long as the record industry’s lawsuit hangs over the company.

Standards bodies also provide a place for industry participants to gather, compare notes, identify shared challenges and find common ground. Such informal discussions are just as important to the growth of new areas as the formal standards processes. That’s especially true when, as with P2P, companies don’t even have common terms to express what they’re doing.

There have been two high-profile efforts to create such a “shared space” for the P2P world. Intel began the peer-to-peer working group to promote the concept of P2P computing and to foster a stronger developer community. The first meeting, in October, drew a huge crowd. Unfortunately for Intel, many of the participants took issue with Intel’s proposed structure for the group, ironically a hierarchical model with a steering committee of paying corporate members. Intel has said it will rethink the structure, though no alternative has yet emerged.

The other process was launched by Tim O’Reilly of O’Reilly and Associates (see RELEASE 1.0, NOVEMBER 1998). O’Reilly convened a meeting of about 20 P2P thought leaders in September to start a dialogue, and he is planning a P2P conference for February.

One way to make sense of the P2P landscape is to concentrate on what the different services are useful for. Users care about – and pay for – things like functionality and convenience. In this way, P2P applications can be grouped into four categories:

- **EXCHANGES** (value in the goods and services stored or transferred)
- **COLLABORATION** (value in the communications and information-sharing between nodes)
- **INFORMATION DISCOVERY** (value in the intelligence the group effort delivers to the individual user)
- **DISTRIBUTED COMPUTING** (value in the aggregate product)

The categories could also be labeled as shared storage, shared spaces, shared information and shared processing. Some application suites may cross category boundaries. For example, a virus-protection service could use P2P file replication to disseminate patches (as MyCIO’s Rumor does today) and distributed information discovery to identify new viruses from exception patterns across many machines.
Below we describe the first three of these areas in greater detail, along with interesting companies in each. Next month we’ll tackle distributed computing and the underlying infrastructure that supports P2P applications.

**Exchanges: “I Get Something from You”**

Thanks to Napster, file sharing is the function most closely identified with peer computing.\(^1\) Peer-based file exchange shares the storage capacity of client computers, giving each of them access to a vast redundant network virtual hard drive. Looked at a different way, though, Napster is just a market. True, it’s limited to music and doesn’t involve any financial transfer. But it’s a way for those who want something of value to find others willing to provide it. “Exchange” usually implies some central trading post, though that’s not necessary if there are other ways for buyers and sellers to come together and transact efficiently.

**Beyond Napster**

Putting aside copyright issues, it’s not obvious that a peer-based architecture is the best way to distribute music files over the Net. Napster offers terabytes of content, but what exactly is available at any given time is unpredictable and unreliable. A particular song may show up one time on a T1 connection, the next time on a slow dial-up line and the next time not at all. (This is compounded by the fact that Napster uses multiple directories and only shows each user a single, random one.) You may start downloading a song but lose it mid-way when the other party disconnects, or you may find that a file is truncated, mislabeled or poorly encoded.

Napster users tolerate all this today because “free” is a compelling price. A service that offered higher assurances of quality and reliability, though, would offer convenience that some users would pay for. Companies such as Emusic.com offer downloadable music today from central Web servers, albeit with mixed financial results. And Bertelsmann is about to test this concept: As part of its deal with Napster, Napster has agreed to offer licensed content for a fee, though presumably still in a peer-to-peer manner.

---

\(^1\) It’s most accurate to call Napster file replication, because users who “share” files actually make additional copies from someone else’s hard drive. Napster itself has emphasized the term “sharing” for legal reasons, seeking to draw an analogy to personal sharing of songs on CDs and cassette tapes.
Most of the early post-Napster P2P platforms were also designed primarily for file trading. Scour Exchange and competitors such as iMesh supported media types other than MP3 files, Gnutella did away with the need for a server-based central directory and Freenet promised virtual immunity from regulation by governments or by private lawsuits. The commercial efforts have struggled, partly because of the legal cloud hanging over Napster, but mostly for the same reason smaller auction sites have had trouble competing with eBay. Napster is so huge that its network effects put competitors at a severe disadvantage. And with Napster committing to develop a secure licensed-music version, “Napster without the copyright issues” becomes a less-than-compelling business plan.

Gnutella and Freenet, with their collaborative development efforts and broader vision beyond media files, have enjoyed more success, though both are still in the development stage. Both are probably best seen as laboratories for prototyping new technical approaches and applications, though it remains possible one or both will make a Linux-like jump to the mainstream.

**MoJoNation: mojo rising?**
MoJoNation CEO Jim McCoy was an early employee of Rocketmail, which later became Yahoo! Mail. But he traces the roots of his new startup earlier, to the mid-1990s controversy over the US government’s proposal to embed a wiretapping-friendly “Clipper chip” in communications devices. On the Cypherpunks email list at the time, tech-savvy privacy advocates kicked around ideas for uncontrollable data havens that could foil the government’s efforts. The Clipper chip was never deployed. McCoy, however, remained interested in decentralized economic regimes, adding ideas from research work in the ’70s and ’80s on distributed markets for CPU time on mainframes and computer networks.

McCoy started MoJoNation to implement decentralized, self-organizing markets in the real world. The trouble with previous systems was that they required digital currency, which imposed a huge set of technical, operational and security challenges. “The big jump we made was saying you don’t need digital cash, you just need a way of keeping score,” says McCoy. Again, this is a familiar concept in academic work, but MoJoNation wants to apply it to a large-scale real-world system, with file exchange as the first application.

Mojo, the currency of MoJoNation, is valuable only inside the system (though in theory it could be converted into other currencies or Web-based loyalty points.
based on an exchange-rate mechanism). If Al has spare storage capacity or bandwidth that Ethel wants, she can pay him for it in Mojo, which he can then use to purchase resources from others. Every transaction in MojoNation is monetized. In effect, MojoNation is a self-contained barter economy, where participants set the value of their resources without recourse to a central economic authority.

"It started off as a cool idea and a way to efficiently aggregate a bunch of machines together toward a common task," McCoy recalls. Then Napster happened. McCoy quickly positioned MojoNation as a generalized P2P platform with a built-in economic exchange structure, and released a beta version to the world at the DefCon hacker conference in July. MojoNation now must turn this concept into a business. Though implemented in a distributed way, MojoNation enjoys some of the same advantages — and faces some of the same hurdles — as online business-to-business exchanges (see Release 1.0, September 1999). Success depends on liquidity, and in creating efficiencies for buyers and sellers in a way that allows the market-maker to generate revenue.

McCoy argues that users will prefer MojoNation to alternatives such as Napster because if they are willing to pay in Mojo, they can guarantee higher quality of service. By imposing costs for every transaction, he claims, MojoNation will overcome the "tragedy of the commons" scenario that threatens to swamp Gnutella (see page 25). Each MojoNation node keeps track of network load, allowing capacity to be offered nearly for free when demand is low. Also, the system has a built-in distributed file-transfer architecture. Files are broken up into pieces spread across many machines, in multiple copies; when a node downloads a file it requests the pieces in parallel rather than downloading the whole thing from one machine as on Napster. "It’s basically a market-driven cache in the background," says McCoy.

MojoNation the company acts as a matchmaker to bring together buyers and sellers. It is developing tools for specialists to help users find particular kinds of information or resources on the network ... a service they would of course charge for. (MojoNation thus might also fit into our “information discovery” category, though its initial use is content exchange.) McCoy sees MojoNation’s primary revenue coming from licensing the platform to other companies. For example, a movie studio might benefit from MojoNation’s content-distribution mechanisms, or a portal could tie Mojo to its loyalty points program. If MojoNation gets to scale, these local economies will feed into the larger pool, but it will be a challenge to reach critical mass.
 Collaboration: “I Learn Something from You”

File-sharing is exciting for individual users, but doesn’t provide much of an argument for P2P technologies in business. Fortunately, another class of problems around collaboration and messaging also lends itself to P2P solutions. The value here comes from the communication links between users, and the shared activities those links enable. By contrast, file-sharing emphasizes the content that moves across the links from one party to another. In collaborative applications, the P2P technology hides the network, applications and other elements of the communications channel, allowing participants to collaborate as though they were in the same local environment.

Instant messaging (IM) (see Release 1.0, June 1997) is a collaborative P2P application, because users communicate with one another directly (see page 18). Flexible, rapid communication among small communities of users is a foundation for all collaboration. Groupware platforms such as Lotus Notes are built around communications tools such as email, which are logically P2P though physically dependent on central servers (see page 5). Moving the underlying communications platform to a P2P architecture is thus in some sense a logical step.

PUBLIUS: CENSORSHIP-RESISTANT DECENTRALIZED PUBLISHING

P2P advocates often cite difficulty of control or regulation as a benefit. The same technology that allows college students to acquire commercial music files for free also can help dissidents and whistleblowers evade censorship and retribution, whether by government or corporate entities. Napster, though, relies on central directory servers that can be shut down by court order (a prospect that is all too real). Gnutella has no such limitation, but it also offers no way to verify that messages are actually from a particular publisher.

Publius was developed by Lorrie Cranor and Avi Rubin of AT&T Labs and New York University computer science grad student Marc Waldman. Cranor is involved in several efforts to develop technical responses to policy challenges including privacy violation, spam and content deemed inappropriate for children, including the World Wide Web Consortium’s platform for privacy preferences project (P3P). Publius follows in that vein.

The Publius system consists of publishers (who post content), servers (who host it) and retrievers (who read it). A publisher encrypts content using a special form of encryption that allows the key to be split up and later reconstituted. The content is uploaded to a number of different servers, each of which gets the (random-looking) data and one share of the key. Publius’ publishing tool creates a special URL that allows retrievers to download the content and enough shares of the key to decrypt it.

Because of the encryption mechanisms involved, Publius users can determine whether content has been tampered with. Publishers can update or delete content they have uploaded, but if anyone else tries to do so the material will become unreadable or identifiably tampered. Though Publius doesn’t require anonymity or digital signatures, those could naturally be layered on top.

Publius’ creators concede the service isn’t censorship-proof. A censor could still block enough servers to prevent reconstruction of the key, for example. It also doesn’t do away with servers, just with the model of publishing content on a single server that could be blocked. Publius currently caps objects at 100 kilobytes to server overload, which also keeps the system from being used for swapping media files. Publius is up and running in a live trial, using servers donated by over 40 individuals and organizations (see Resources section).
**Groove Networks: shake your groove thing**

When Lotus Notes developer Ray Ozzie left Lotus in 1997, following its acquisition by IBM, he wanted to build a new product as revolutionary as Notes. He just wasn’t sure how. “We didn’t really know what technology specifically was going to be needed to build what we wanted to build,” Ozzie recalls. What was essential was that “people had to be able to spontaneously make decisions to connect with one another.”

The biggest limitation of client-server groupware systems such as Notes and Microsoft Exchange is that they require approval and coordination from the center. Deployment, configuration, adding new users, upgrades and new features are all laborious tasks requiring heavy IT investment. At the other end of the spectrum, messaging services such as email are totally flexible and ad hoc, but lack the context and structure necessary for deep, ongoing collaborative activities.

Ozzie’s solution is a distributed, peer-to-peer architecture. Groove puts the bulk of its functionality in the client, allowing users to create “shared spaces” for direct communication among team members. A user installs the Groove client (currently Windows-only, but with Linux support planned and a Macintosh client likely) and invites others into a space through email or an instant message. If the recipient accepts, that automatically sends a message back requesting the shared space that directly connects the two users. The objects and content that make up the space are replicated on each user’s machine. Participants can engage in a range of functions including messaging, threaded discussions, shared whiteboarding and calendaring. They can also drop in external files, which launch the appropriate desktop application for editing.

The tools built into Groove are just the beginning, though, because the software was designed explicitly as a platform. Developers can create specialized tools that run on top of Groove, much as companies built on Notes for their own needs. “The reason people don’t build distributed apps is that it’s really hard,” notes Ozzie, explaining that Groove allows developers to build services in familiar languages without worrying about thorny details underneath. In addition, Groove supports Microsoft’s COM model and SOAP (see page 17) for integrating with other applications.

**It only looks simple**

Groove can work in a totally decentralized manner from the user’s perspective, but it takes advantage of servers to provide much of its functionality. For example, Groove
employs relay servers to manage presence detection, cache information when users go offline, serve as proxies to go through firewalls and more efficiently broadcast multiple copies of information to groups. It also takes advantage of familiar applications such as email whenever possible, so that users aren’t forced to change behavior.

All this ties into Ozzie’s design philosophy for Groove, which involves hiding network complexities so that users can focus on how they want to use the software. “We had a very rigorous goal of having a user interface where you never saw security options displayed, and you never saw communications options displayed,” says Ozzie. “It should just do the right thing.”

Implementing this maxim consumed the bulk of Groove’s long gestation. For example, despite the vision of a transparent end-to-end Internet, features such as firewalls, network address translation (NAT) and dynamic IP addressing (DHCP) break most applications that go beyond static Web pages (see RELEASE 1.0, MAY 1999). Groove uses sophisticated adaptive techniques to overcome these barriers behind the scenes.

The hardest part, Ozzie says, was creating the appearance of a controlled, localized network when users are dispersed over the chaotic, unreliable Internet. If participants in a space are on connections with different speeds, or if one of them disconnects entirely for several hours to get on a plane, and all of them make changes, how can the platform keep everything in sync?
“The essence of what we had to do is synchronization,” Ozzie explains. When a Groove user takes any action, the command is formatted into XML and sent into message queues that drive the display on the user’s screen and simultaneously push the change information or “delta” out to others in the shared space. The delta is also encrypted and stored locally in an integrated XML object store. Unlike the Web and Notes, built around documents, Groove has no fundamental datatype. Each tool has its own engine that runs in the client and decides how content, gestures and commands are sent across the network and placed in context with similar changes made by other members of the space.

If a user is offline, the information is stored in a relay server, which the Groove client automatically contacts when logging back on. Because every action is encoded as a time-stamped delta from the previous state, the system can always rewind and reinsert changes based on when they were made rather than when they were received over unreliable network connections. Groove provides a default storage quota on its relay own servers, and hopes to sell additional capacity and service-level agreements to its corporate customers. For security, Groove encrypts all communication between peers and all deltas before they are stored locally or sent over the wire.

Early interest in Groove has come from industries such as financial services (where companies want tighter relationships with their best customers), entertainment (which hope to tie into organic groups of fans who communicate around artists or content) and business-to-business (B2B) areas such as collaborative design (where transaction exchanges only address some of the relationships between partners).

**Roku: leverage rather than sync**

At first glance, Roku resembles synchronization services such as Fusion One (see RELEASE 1.0, JUNE 1999). Behind the scenes, though, Roku uses a distributed peer-based architecture that distinguishes it from products based on central data repositories. “We’re about leveraging information, rather than synchronizing and replicating information,” says CEO Chuck Ennis.

Roku has been in development since 1995, when co-founders Richard Kilmer and Scott Dankman started work on a Java-based object model to connect an individual’s personal information and resources across scattered applications and devices. (Roku, the Japanese word for six, refers to the six types of information – who, what, where, when, why and how – the model supports.) In its first
commercial release, in 1999, Roku used this platform to send Web and email content to users’ mobile phones. Since then, though, the company has turned its efforts to the broader P2P implications of its technology.

Roku now offers two products: Roku Access and Roku Share. The first lets users view and manipulate information on their PC, in any major application, through a browser interface. It supports notification via email and wireless device when important changes occur. The data stay where they are, in contrast to synchronization solutions, but they are universally available through the Net. Ennis argues this approach is superior to synchronization because users need not decide in advance what content to synchronize, and need not pay a synchronization provider to replicate and host their own data. “I just want to know that I can get to any of my stuff at any time,” he says.

The second product is more directly collaboration-oriented. Claims Ennis, “If me, you and two others are working on a project, there’s no need for central mediation. There’s no one in the center adding intelligence.” Roku Share lets users drag content such as documents, email messages and URLs into a shared folder, where it is immediately accessible to others in a self-defined group. The initial user invites others into the group by email, and the recipients click on an embedded URL to view the folder contents in a browser. For security, all Roku connections are encrypted using secure socket layer (SSL) and the packets going over the wire are themselves encrypted.

Aspects of Roku’s approach parallels that of Groove, though its use of a browser rather than desktop client software makes it both simpler to use and more limited in scope. Roku’s business model is also somewhat different. The company sees the primary benefit of its software in its ability to tie together other local and network-based applications, while building a rich local profile of each user behind the scenes. Because the profiles stay on the user’s machine, under the user’s control, they don’t raise the same privacy issues companies such as DoubleClick face.

Roku will license its platform to service providers as a means to drive additional transactions. It has announced deals with 3Com and Hewlett-Packard, with other similar-sized partnerships under discussion. Users of HP-developed corporate portals, for example, will be able to gain access to their files and other resources from any browser. By early next year, Ennis expects over 10 million copies of Roku to be in circulation thanks to such partnerships. Roku charges an up-front licensing fee, but Ennis is most excited about the upside potential from revenue sharing with its partners whenever Roku mediates a transaction.
**Consilient: process-centric collaboration**

Consilient’s mission is to automate business processes that span diverse groups of people, organizations and systems. Though it only officially launched this month, Consilient already has major global enterprises as customers including BP Amoco. The name Consilient suggests consilience, the concept of unification of human knowledge across disciplines popularized by biologist E.O. Wilson.

Even a relatively simple functional task such as procurement can involve many different and unconnected resources. B2B exchanges and transactional software such as Ariba automate the ordering process, but are only loosely connected, if at all, to ancillary resources such as third-party information providers, financing, internal enterprise resource planning (ERP) systems, customer-support representatives and workgroup members in the company doing the procurement (see **RELEASE 1.0, OCTOBER 2000**). With potentially thousands of suppliers, tens of thousands of user desktops and hundreds of applications in the loop, even sophisticated enterprise application integration frameworks can only go so far.

“Our solution allows them to leave the diversity as it is, to aggregate rather than integrate,” says Consilient CEO Jonathan Hare, previously CEO of Evolve Software. Hare sees the problem as a perfect illustration of the tenets of complexity theory. No top-down structure can scale with the growth of connections between users and systems. When business users think about the activities they engage in, they conceptualize them in bottom-up, ad-hoc terms rather than focusing on traditional applications. “Processes are grown; they are not engineered,” Hare says. The only effective way to automate these processes is to build models around users themselves and connect systems to those models directly.

---

**Writing the Web**

“What I want to do is turn the Web into a writing environment,” proclaims Userland’s Dave Winer. Building on the browser-based editing features of Manila, its Weblog and content-management tool (see **RELEASE 1.0, JULY/AUGUST 1999**), Userland is exploring new forms of P2P Web publishing.

In the traditional model, users create content then upload it to a central Web server. In Winer’s vision, selecting “Save” in the file menu of a word processor should automatically make content available over the Web (assuming that’s what the author wants). Websites should talk directly to desktop applications, and those applications should be able to talk to one another directly over the Net. Userland was one of the developers of SOAP, a standard co-authored by Microsoft, IBM, Developmentor and Lotus for distributed information exchange among applications.

In Winer’s vision, everyone can be a Web author as well as a filter, without sacrificing the richness of existing desktop software.
Sitelets are the centerpieces of Consilient’s offering. These are small dynamic agents built in Java and XML that embody a virtual process. A sitelet can be delivered via the Web, email or other communications mechanisms. It connects to applications, content and client devices and assembles material into a unified view. The result looks like a hybrid of Zaplets (see RELEASE 1.0, MARCH 2000) and Web aggregators such as Yodlee, Octopus and OnePage (see RELEASE 1.0, APRIL 2000). The sitelets are virtual spaces connecting users peer-to-peer, similar to Groove, but they can travel in email and aggregate dispersed content into a single consistent interface. This is important when users are spread around the world and frequently in transit, as is the case with managers at most global businesses.

The tricky part for Consilient is securely plugging into all the disparate applications and systems to create a single process-centric view for each user. Despite clear desire to create interoperable XML-based IM, illustrates this broader conception. Jeremie Miller, then working at an ISP, created Jabber in 1998 as an XML-based framework for generalized IM services. Jabber isn’t another IM platform, such as Yahoo Messenger, Odigo or MSN Messenger; it’s a back-end system that can plug into all those services.

Basically, Jabber switches real-time XML messages, and those messages can be formatted to display in its own client or to interoperate with all existing IM clients. The applications Jabber’s leaders see for the technology include real-time auctions and e-commerce, groupware, live customer service, distributed software development and wireless device notification. Another company called Aimster has combined IM with file sharing by using AIM’s messaging and addressing platform as the basis of a Napster-like service.

A for-profit company, Jabber.com, now develops Jabber as a commercial platform, while Jabber.org continues the open-source effort in parallel. Jabber hasn’t yet been widely adopted, in part because the current software isn’t always easy to install and use. It will face the same business challenges as other IM players going forward.
and substantial expenditures, major enterprises have largely failed to integrate their processes end-to-end because solving these problems from the center is so difficult. Moreover, when processes span organizations (as with B2B exchanges) the battle over who controls the integration layer is often unresolvable.

Consilient doesn’t actually integrate systems, it just puts documents and information side-by-side so that business users can utilize them in connection with a consistent workflow. BP, for example, is using Consilient to automate global financial budgeting, which involves countless loops of projects, proposals and approvals.

Information Discovery: “I Find What I’m Looking For”

The Net’s vastness makes finding things difficult. When the Web first grew beyond the ability of individual users to locate what they were looking for, the result was search engines and directories. When the first generation of search engines ran out of steam, new technologies such as Inktomi’s clustering and Google’s link-structure analysis were brought to bear (see RELEASE 1.0, January 1999).

Google, with 1.2 billion pages in its index, does a remarkably good job of keeping up with the growth of the Net, but it’s far from being a universal solution for online information discovery. Search engines must constantly spider the Web to find new and changed pages, which takes time and ever-increasing computational power as the Net grows. By and large, search engines and directories index only static Web pages and other documents; they can’t look into the databases behind a site or on a corporate network. They also have only limited ability to understand what a particular object is about.

I may know what I’m looking for, but if the network doesn’t know, and doesn’t have any way to understand my request, it’s difficult for me to get results that match up with my specific interests. To overcome this hurdle, the Net itself must act with intelligence. This is a major leap from the relatively static, passive-search paradigm that Web users take for granted today. It’s possible, though, if a large number of machines at the edge of the network start contributing their own intelligence and processing power to the tasks at hand. Some startups such as PurpleYogi hope to improve search by mapping together classification engines running on the client and server sides. Others push decentralization even farther.
Infrasearch: the P2P search engine
Perhaps no company symbolizes the excitement around P2P as well as Infrasearch. Seven months ago Gene Kan was a programmer at WeGo, which sells software to allow membership organizations to create and manage dynamic Websites. He was also active in the Gnutella project. With two friends, Yaroslav Faybishenko and Cody Oliver, he created a prototype distributed search engine running on the Gnutella platform. The original aim was to solve a problem WeGo was experiencing: how to search across information its customers had stored in databases in addition to static Web content.

But these were the high days of Napster hype, and Infrasearch took on a life of its own. Marc Andreessen invested, Infrasearch’s founders started appearing on magazine covers . . . and another Silicon Valley legend was born. Now all Infrasearch has to do is develop a product, launch it, articulate a business model, sign up customers and make money. To that end, the erstwhile Infrasearch went into stealth mode and moved its site to Gonesilent.com. The company is building a next-generation distributed search solution that runs on its own platform instead of Gnutella.

To Kan, the argument for P2P is obvious: “Peer to peer makes so much sense,” he begins. “You have a very powerful computer on your desktop or even in your server room, and that computer’s resources aren’t being utilized fully.” Despite this, Kan says, the predominant philosophy of application development taught in computer-science programs is client-server oriented. As a result, many developers assume P2P architectures are infeasible. Yet Napster and Gnutella are existence proofs that P2P applications work. Kan believes the speed of network connections and processors today, along with the low price of disk storage and RAM, make distributed architectures previously written off in academic circles viable.

At this point we come to a religious conflict . . . Many observers see search as the application least amenable to P2P approaches. Though all contemporary search engines use distributed crawlers to identify new and changed pages, they pull information back into central databases which users query. Running each query through a decentralized P2P network, the argument goes, is a recipe for performance meltdown. On the other hand, some P2P advocates see search as a perfect use for distributed mechanisms. If the point of search is to find information distributed around the Web, they argue, why do so in a centralized way? “It’s counter-intuitive to centralize search,” says Andreessen, who
compares doubters to the phone companies in the pre-Internet days who argued that packet switching would never work. There are no commercial P2P search services yet deployed, so we'll have to wait for the verdict of the marketplace.

Infrasearch believes it can trump current search engines by connecting information providers to one another. Search for “watch” on Infrasearch's prototype system, and you'll get a headline starting “All eyes watching vote count” from a news provider and a list of watches currently on sale, with prices, from an auction site. Infrasearch itself doesn't generate alternate word-forms such as “watches” or “watching,” nor does it decide how to digest the information around the search term. It leaves those functions to the information providers, who best understand the logical structures appropriate for their resources. For example, WeGo returns a logo and a brief description of sites in its database when queried through Infrasearch. Traditional search engines, unaware of context, simply return words around the search term or the beginning of the page.

Information providers can also serve results directly from databases, because they are responding directly to a live query. Sites can expose whatever data they want to the search engine, even if it's encrypted or resident in a non-Web application such as a legacy financial system. And because each query gets a live response, Infrasearch promises fresher results than existing search engines. All this suggests Infrasearch may find a niche initially in site-search and vertical-market situations, even if it can't displace companies such as Google and Inktomi in Web-wide search.

Infrasearch hopes to generate revenue by convincing Websites to pay to connect to its network. Because Infrasearch can query resources that conventional search engines miss, it may be able to drive incremental traffic to participating sites. Like most Internet startups in these unstable times, though, Infrasearch also has a B2B story. The company plans to license its technology or sell subscriptions to businesses looking for a better way to find information on their far-flung internal networks.

OpenCOLA: shake it up
OpenCOLA's three founders, Cory Doctorow, John Henson and Grad Conn had worked together for the past five years on various IT projects. “We set out to make a personal tool for improving our ability to stay abreast of information that we tracked on the Net,” says Doctorow. They quickly realized that spidering the Web like a search engine and applying relevance criteria requires massive computational
resources. The solution was to use a radically distributed approach. (COLA stands for collaborative object lookup application.)

OpenCOLA calls its nodes “clervers” because they function as both clients and servers. Clervers are general kernels for distributed applications, with software robots running on top that seek out and filter resources on certain topics. A user creates a topical robot to keep abreast of a particular subject, and the robot operates persistently until shut down. The robots use collaborative filtering (see Release 1.0, November 1996) to discover likely sources of documents, then apply relevance criteria to evaluate the documents they find there.

The twist is that each user’s robot also checks with other robots to enhance the quality of the results. “As everyone looks at documents on the Internet they make decisions, whether overt or automatic,” begins Doctorow. “We can capture that information and then route it to the people who are most likely to benefit from it. OpenCOLA becomes a collaborative, idiosyncratic Yahoo! that indexes any part of the Internet that anyone using OpenCOLA is interested in.”

Clervers query one another to calculate their similarity. “My robot and your robot negotiate their similarity by checking how they rank documents,” explains Doctorow. The nodes compare how each of them ranks the relevance of a series of documents, and compute a difference factor. Once nodes know how “close” they are to one another, the network can self-organize, much like a roomful of people who assemble into a line in order of height by individually attempting to stand to the left of someone taller than them. This approach also overcomes the scaling issues plaguing Gnutella networks, Doctorow says, because queries can be directed first to “closer” nodes rather than being broadcast indiscriminately to the entire network.

OpenCOLA builds in a distributed reputation system, based on “pedigrees” of information (where it has been). Every message has a pedigree in addition to its immediate source and destination. Robots assign scores to documents and present them to users, who can give feedback about whether the documents are useful or not. That feedback flows back through the system, so that users are ranking not only documents, but their peers as well. “The people who make consistent and heroic efforts to consume information will have those decisions passed along to a great number of people who are less active and simply follow along their decision making,” says Doctorow.

OPENCOLA INFO

Headquarters: Toronto, Canada
Founded: June 1999
Employees: 50
Funding: $2.5 million from Mosaic Venture Partners and individuals
URL: www.opencola.com
OpenCOLA avoids the problem of a few robots doing all the work because best robot-trainers can use their (large) audiences’ resources to spider more sites and filter more documents. Discoveries by low-impact participants are quickly passed to the “super-recommenders” and from there to the entire network.

Doctorow draws an analogy to a bidding agent he wrote to use in eBay auctions. To find auctions of interest, the agent keeps track of competing bidders in one auction, then identities other things they are bidding on. Another analogy would be Epinions, which uses a similar reputation model but in a server-based environment.

All this is only the beginning. OpenCOLA is experimenting with a reputation stock market, to identify which participants have the largest following for the quality of their robots. It is also working on a technology called Swarmcast for distributing large files around the Internet. Based on work at the University of Minnesota, Swarmcast uses techniques similar to those employed in RAID (redundant array of inexpensive disks) storage arrays, only done across many PCs on the Net. The first person who requests a file gets an initial chunk; the next person gets the second chunk and a pointer to the first person and so on. Doctorow calls this “Adhocamai.”

Doctorow also describes a novel distributed digital rights management application based on auditing of authenticity rather than copy protection as another possible
addition to the OpenCOLA platform. Finally, all OpenCOLA code is open-source, so others can build new services and features beyond those the company is developing.

From Peer to Eternity

Everyone who looks at P2P technologies, it seems, brings his or her own viewpoints and backgrounds to bear. Software developers see new platforms for creating applications, economists see new kinds of markets, messaging and routing experts see familiar architectures supporting new applications, writers see a publishing environment, event-based integration proponents see event-notification buses and complexity theorists see... well, complexity. There's no point fighting over what's really new here. For P2P applications and platforms to succeed they must provide some benefit to either developers or users.

Why P2P?
Putting aside the application-specific benefits and legal factors that are so important to Napster, P2P platforms have four primary benefits over centralized approaches.

First, they are inherently flexible, because every network node is also a control point. With Groove (see page 13), for example, users need not coordinate with an administrator to update a list of participants or to deploy a new application to a project team. (If there is a central directory, it's updated automatically in the background.) The participants themselves simply add a new member or a new application plug-in to their shared spaces. Each user can have his or her unique experience and organizational structure, because the application logic resides in a user’s peer client rather than a central server for many users.

Second, P2P configurations may be more scalable, because if designed well they take advantage of self-organization (see page 6). For certain applications, especially exchanges, P2P services enjoy increasing returns to scale, what Trellix cto Dan Bricklin calls the “cornucopia of the commons” phenomenon on Napster. The more people who use the system, the more content the system has available. For some problems, such as information discovery, P2P approaches promise to do better than centralized mechanisms ever can. In other cases, the potential advantage takes the form of reduced cost or greater performance.
Pat Gelsinger, CTO of Intel’s architecture group, estimates Intel has saved more than $500 million over the past ten years using its NetBatch application to distribute engineering calculations over more than 10,000 workstations on its internal network. He also calculates that distributed storage costs twenty times less per thousand clients than the cheapest centralized storage available, and that peer-based file transfers on Intel’s local-area network can take advantage of 75 times the bandwidth available on wide-area connections over the Internet.

Third, P2P platforms leverage the under-utilized power of networked PCs, what Clay Shirky calls “the dark matter of the Internet.” Instead of turning powerful desktop PCs into dumb terminals, P2P services take advantage of their processing power either individually or collectively. Because the collective processing power of many PCs exceeds that of the smaller number of servers in the network’s core, this aspect of P2P platforms further bolsters scalability.

Finally, P2P networks are harder to control. This may be an advantage or a disadvantage, depending on the situation and who is doing the evaluation. If the goal is to publish information that a restrictive government wants to censor (see PUBLIUS, page 12), or to allow small teams of workers to innovate without being stifled by sclerotic management, uncontrollability is a good thing. Of course, the music industry sees the difficulty of controlling P2P services such as Napster and Gnutella in an entirely different light.

On the other hand . . .

In addition to legal issues such as Napster’s challenge to intellectual property, there are other potential pitfalls for P2P networks. P2P platforms live or die from network effects. When many individual users participate, as with Napster or SETI@Home, the results exceed what is possible with centralized infrastructure. The opposite possibility is the tragedy of the commons, which appears to be occurring with file sharing on Gnutella.

As Bernardo Huberman and Eytan Adar of Xerox PARC demonstrated in a recent paper (see RESOURCES), most Gnutella users free-ride. In Huberman and Adar’s test, 70 percent shared no files, and nearly half of all files came from one percent of hosts. Other P2P platforms such as MojoNation and OpenCOLA have built-in brokering mechanisms to avoid this problem, though, and Clip2.com is developing a brokering layer that runs over Gnutella called Reflector.
P2P is in some ways less efficient than centralized approaches, because the same data is replicated many times on client machines. The question is what resource it's better to waste. For example, content-delivery networks such as Akamai waste storage space on edge servers in order to economize on bandwidth and processing at central servers. P2P platforms waste bandwidth and storage but economize on processing by utilizing all the computers in the network. The answer may depend on the application. Argues Lucas Gonze, CEO of P2P consulting and application-server developer WorldOS, “Computer scientists want to ask the question, ‘Does this use resources efficiently?’ The more important question is, ‘Does this use resources profitably?’”

P2P platforms such as Gnutella are essentially flat, meaning that individual nodes must often send messages throughout the entire network to find a resource. This can result in heavy message traffic and affect performance. Other P2P networks get around this problem by allowing nodes to identify which other nodes are “closer” to them (OpenCOLA) or by using central servers to relay and queue messages between nodes (Groove).

Another objection to P2P networks is that they re-invent the wheel. Sure, Yahoo!’s categorization model based on employee editors doesn’t scale well, but the Open Directory Project’s distributed volunteer network provides an alternative. And combining the human-generated ODP with efficient server-based (but distributed) crawler technology, as Google now offers (see RELEASE 1.0, JANUARY 1999) is better still. The question is whether the additional complexity of deploying robust P2P networks is worth it.

And of course there are those pesky little issues around security and business models. Do I really want a bunch of strangers poking around in my machine? And is that something I’d be willing to pay for? As discussed above, most of the business-focused P2P companies encrypt all their communications by default and build in other security features, but no security system is perfect. As always, the business issues will ultimately turn on whether the service providers can provide services that someone else finds valuable. More on these questions next month . . . .

Peer preview

Explaining his excitement about the P2P phenomenon, Marc Andreessen recalls his prior experience creating Mosaic and Netscape Navigator: “After the Web emerged, people started to think about things differently. It unlocked creativity that was there all along.” Peer-based approaches represent a similar shift in thinking. There isn’t
anything radically original going on at the fundamental technology level, but the conditions are ripe for a Cambrian explosion of diverse applications and services.

The hard work of turning insanely great ideas into robust, popular products and services has just begun. Most of the dozens of P2P startups now being funded will fail. Yet P2P, taken generally, makes sense for the same reason the Internet makes sense: There will always be more collective intelligence at the edge of the network than in the core. Given billions of years, evolution chose distributed architectures for some of the most complex systems on Earth, including the human brain. It’s hard to escape the feeling that nature is on the side of P2P. ■ R 1.0

COMING SOON

- Peer-to-peer, Part II.
- Broadband service platforms.
- How safe is your site?
- Triumph of the Weblogs.

- And much more. . . (If you know of any good examples of the categories listed above, please let us know.)
Resources & Contact Information

Clay Shirky, Accelerator Group, 1 (212) 274-1305; fax, 1 (212) 274-1848; clay@shirky.com; www.acceleratorgroup.com or www.shirky.com

Lorrie Cranor, AT&T Labs, 1 (973) 360-8607; lorrie@research.att.com

Jonathan Hare, Consilient, 1 (510) 981-9200; fax, 1 (510) 981-9700; jonathan@hare.com

Gene Kan, Gonesilent (Infrasearch), 1 (650) 697-0922; gene@gonesilent.com

Ray Ozzie, Groove Networks, 1 (978) 720-2222; fax, 1 (978) 525-8377; ray@groove.net

Steve Burbeck, IBM, 1 (919) 254-0732; sburbeck@us.ibm.com

Bob Knighten, Intel, 1 (503) 677-4315, Bob.Knighten@intel.com

Philip Bernosky, Intel, 1 (408) 765-8095; philb@intel.com

Miko Matsumura, Kalepa, 1 (650) 329-8888; miko@kalepa.com

Rohit Khare, Adam Rifkin, KnowNow, 1 (206) 264-2897, rohit@knownow.com, adam@knownow.com

Doc Searls, Linux Journal, 1 (650) 361-1324; fax, 1 (650) 361-1348; doc@searls.com

Marc Andreessen, LoudCloud, 1 (408) 744-7525; fax, 1 (408) 744-7381; pmarca@loudcloud.com

Jim McCoy, MojoNation, 1 (415) 810-7224, mccoy@mad-scientist.com

Cory Doctorow, OpenCOLA, 1 (415) 726-5209 or 1 (416) 588-1774; fax, 1 (650) 745-2844; cory@opencola.com

Tim O'Reilly, O'Reilly & Associates, 1 (707) 829-0515, tim@oreilly.com

Chuck Ennis, Roku, 1 (703) 449-1700 x304; fax, 1 (703) 449-9302; chuck@roku.com

Alex Cohen, University of California at Berkeley, 1 (415) 824-0274; xcohen@doxadox.com

Dave Winer, Userland, 1 (650) 851-7083; dave@userland.com

Lucas Gonze, WorldOS, 1 (917) 805-4391; lucas@worldos.com

Bernardo Huberman, Xerox PARC, 1 (650) 812-4147; fax, 1 (650) 812-4388; huberman@parc.xerox.com

For further reading:


Working Group on Peer-to-Peer Computing: www-peer-to-peerwg.org

Publius project: publius.cdt.org

Proceedings of July 2000 TWIST conference on decentralization (organized by Rohit Khare):

www.isr.uci.edu/events/twist/twist2000
Calendar of High-Tech Events

2000

DECEMBER 1  THE POLICY IMPLICATIONS OF END-TO-END - Stanford, CA. Technologists, policy makers and lawyers will explore the values implicit in the end-to-end design, and attempt to understand the consequences of deviating from e2e. For more info, call 1 (650) 736-1521; fax, 1 (650) 723-8440; email, maverick@forsythe.stanford.edu; www.law.stanford.edu/e2e.

DECEMBER 4-7  GROUND ZERO 4 - Los Angeles, CA. Net market makers gather to ponder what lies beyond the smoking ruins of the current market for B2B stocks. For more info, contact Margie or Kristine at 1 (510) 647-3799 or email, margie@netmarketmakers.com; www.nmm.com/events/groundzero4.

DECEMBER 5-7  HOMENET FORUM: WINTER 2000 - San Jose, CA. The fourth in a series of summits on home networking. For info, please contact Warren Lee at 1 (212) 983-9582; fax, 1 (212) 986-2124; email, conference2@marcusevansny.com. pull.xmr3.com/791-D1F3/1110356/HOME.HTM.

DECEMBER 7-8  JUPITER ENTERTAINMENT FORUM - Los Angeles, CA. Is the Internet a viable medium for entertainment? To register, call 1 (800) 611-1693; fax, 1 (212) 475-3896; email, customerservice@jup.com; www.jup.com/jupiter/events.

DECEMBER 7-10  ISEA2000 - Paris, France. The 10th International Symposium on Electronic Art will explore the effects of the technological revolution on art and its impact on society through new forms of representation. For more info, call +33 (1) 46 48 66 36; fax, +33 (1) 46 48 66 59; email, isea2000@art3000.com; www.isea2000.com.


DECEMBER 12-14  STREAMING MEDIA WEST - San Jose, CA. The largest exhibition for streaming, broadband and content companies. For more info, call 1 (888) 301-8890; fax, 1 (888) 301-8895; email register@streamingmedia.com; www.streamingmedia.com/west.

DECEMBER 12-14  ANGEL SOCIETY FORUM - New York, NY. Find your guardian angel among the investors, VCs, entrepreneurs and professional service providers at this forum. Speakers include Fred Wilson, Brad Farkas, Samer Hamadeh and Guy Kawasaki. For info, contact Jim Talcott at 1 (212) 248-5580 or email at jtalcott@angelsociety.com; www.angelsociety.com.

DECEMBER 15  AMERICA: ON THE NET - Washington, DC. Presented by the Internet Policy Institute and moderated by Charlie Rose. For more info, contact Martin Hagen at 1 (202) 662-2544; email, mhagen@internetpolicy.org; www.internetpolicy.org/aotn/summit.htm.
Calendar of High-Tech Events

2001

JANUARY 6-9 CONSUMER ELECTRONICS SHOW 2001 - Las Vegas, NV. The world's largest annual tradeshow for consumer technology. Keynote speakers include Bill Gates and Craig Barrett. To register, call 1 (888) 237-7469; fax, 1 (703) 907-7675; email, CESinfo@CE.org; www.cesweb.org.


JANUARY 25-30 WORLD ECONOMIC FORUM - Davos, Switzerland. This year's Forum will once again bring together top business and political leaders, concerned citizens and creative academic thinkers. For info, see www.weforum.org.


JANUARY 30-FEBRUARY 2 LINUXWORLD CONFERENCE & EXPO - New York, NY. Learn to develop, integrate, and run a business on the Linux OS, with general championing of Linux along the way. For more info, call 1 (800) 657-1474; email, linuxworldexpo@idg.com; www.linuxworldexpo.com/papers.html.

FEBRUARY 4-6 DEMO 2001 - Phoenix, AZ. Chris Shipley picks the hot startups again. For more info, call 1 (650) 312-0545; fax, 1 (650) 286-2750; email, registrar@demo.com; www.demo.com.

FEBRUARY 10-14 MILIA - Cannes, France. In partnership with Forrester Research and featuring more than 800 exhibitors, this international event is for professionals involved in the creation and distribution of interactive content. For info, contact Ted Baracos at +33 (1) 41 90 44 80; fax, +33 (1) 41 90 44 70; email, info@milia.com; www.milia.com.

FEBRUARY 14-16 O'REILLY PEER-TO-PEER CONFERENCE - San Francisco, CA. Gee, maybe there's something to this peer-to-peer thing? Meet some of the companies covered in this issue. For information, visit conferences.oreilly.com/p2p/.

FEBRUARY 19-22 FINANCIAL CRYPTOGRAPHY '01 - Grand Cayman, British West Indies. Brings together those involved in the financial, legal and data security fields to foster cooperation and exchange of the ideas. For more info, contact G. Davida at davida@cccn5.cs.uwm.edu; www.fc01.uwm.edu.

FEBRUARY 21-24 TED 11 - Monterey, CA. Richard Saul Wurman invites you to celebrate youth, age, the two words "will" and "still" and presentations by people under 30 and over 70. For more info, call 1 (401) 848-2299; fax, 1 (401) 848-2599; email, wurman@ted.com; www.ted.com.
## Calendar of High-Tech Events

<table>
<thead>
<tr>
<th>Date</th>
<th>Event Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>February 27-28</td>
<td><strong>ExpoDot</strong> - Pittsburgh, PA. Offers an opportunity for early stage Internet-related ventures, professional service providers, venture capitalists and research organizations to come together and network. Contact Susan Kundla at 1 (800) 422-0871; fax, 1 (412) 288-1340; <a href="mailto:skundla@pghtbs.com">skundla@pghtbs.com</a>; <a href="http://www.expodot.com">www.expodot.com</a>.</td>
</tr>
<tr>
<td>February 28-March 1</td>
<td><strong>Consumer Online Forum</strong> - New York, NY. Jupiter Communications’ flagship event examines the state of the online world. For information, please visit <a href="http://www.jup.com/jupiter/events/forumlist.jsp">www.jup.com/jupiter/events/forumlist.jsp</a>.</td>
</tr>
<tr>
<td>March 10-14</td>
<td><strong>ACM1: Beyond Cyberspace</strong> - San Jose, CA. Explore how computing is affecting the state of the art in such diverse fields as biology, oceanography, astrophysics, life sciences, social sciences, and education, and how these changes will increase our understanding of the world we inhabit. Speakers include Bob Metcalfe, Steve Ballmer, Rodney Brooks and Vint Cerf. For more info, call 1 (212) 626-0500; fax, 1 (212) 944-1318; email, <a href="mailto:acmlinfo@acm.org">acmlinfo@acm.org</a>; <a href="http://www.acm.org/acm1">www.acm.org/acm1</a>.</td>
</tr>
<tr>
<td>March 20-23</td>
<td><strong>Spring Voice on the Net</strong> - Phoenix, AZ. Let your voice be heard at Jeff Pulver’s spring offering. For more info, call 1 (631) 545-0800; email, <a href="mailto:von2001@pulver.com">von2001@pulver.com</a>; <a href="http://www.pulver.com">www.pulver.com</a>.</td>
</tr>
<tr>
<td>March 25-28</td>
<td><strong>PC Forum</strong> - Scottsdale, AZ. Join us! Release 1.0 subscribers receive invitations with this month’s issue. Additional information will be posted on our Website in mid-December. If you are interested in speaking opportunities, please contact Daphne Kis at <a href="mailto:daphne@edventure.com">daphne@edventure.com</a>; <a href="http://www.edventure.com/pcforum/forumhome.html">www.edventure.com/pcforum/forumhome.html</a>.</td>
</tr>
</tbody>
</table>

Events Esther plans to attend.

Events Kevin plans to attend.

Lack of a symbol is no indication of lack of merit. The full, current calendar is available on our Website, www.edventure.com. Please contact Kara Holmstrom (kara@edventure.com) to let us know about other events we should include.
Join our email list! We’ve started a free email newsletter, *The Conversation Continues*, for commentary, industry analysis and pointers to interesting Websites. To sign up, please visit [http://release1.edventure.com/conversation](http://release1.edventure.com/conversation), or send email to [conversation@edventure.com](mailto:conversation@edventure.com) and you’ll automatically be added to the list.

Do we have your email address? *Release 1.0* subscribers can now download each month’s issue electronically. If you have not already done so, please send your email address to natasha@edventure.com or fax it to 1 (212) 924-0240 in order to enable online access.

---

**Release 1.0 Subscription Form**

Complete this form and join the other industry executives who regularly rely on *Release 1.0* to stay ahead of the headlines. Or if you wish, you can also subscribe online at www.release1-0.com.

Your annual *Release 1.0* subscription costs **$795** per year (**$850** outside the US, Canada and Mexico), and includes both the print and electronic versions of 11 monthly issues; 25% off the cover price when you order from our online archives; a *Release 1.0* binder; the bound transcript of this year’s PC Forum (a $300 value) and an invitation to next year’s PC Forum.

**NAME**

**TITLE**

**COMPANY**

**ADDRESS**

**CITY**

**STATE**

**ZIP**

**COUNTRY**

**TELEPHONE**

**FAX**

**EMAIL**

**URL**

*personal email address required for electronic access.

☐ Check enclosed  ☐ Charge my (circle one):  AMERICAN EXPRESS  MASTER CARD  VISA

**CARD NUMBER**

**EXPIRATION DATE**

**NAME AND BILLING ADDRESS**

**SIGNATURE**

Please fax this form to Natasha Felshman at 1 (212) 924-0240.

Payment must be included with this form. Your satisfaction is guaranteed or your money back.

If you wish to pay by check, please mail this form with payment to: EDventure Holdings, 104 Fifth Avenue, 20th Floor, New York, NY 10011, USA. If you have any questions, please call us at 1 (212) 924-8800; email us@edventure.com; www.edventure.com.