LYN BARTRAM AND
MICHAEL BLACKSTOCK,
COLLIGO NETWORKS

Designing Portable Collaborative Networks

A middleware solution to keep pace with the ever-changing ways in which mobile workers collaborate.

Peer-to-peer technology and wireless networking offer great potential for working together away from the desk—but they also introduce unique software and infrastructure challenges. The traditional idea of the work environment is anchored to a central location—the desk and office—where the resources needed for the job are located. Even in the many professions where the practitioners move among different field locations, such as professional consulting, health care, or resource exploration, the full set of information and technology resources has been available only in fixed locations where the workers "check in" periodically to integrate their field results back into the larger picture.

The nature of the workplace is changing, however. People increasingly need and expect to be able to plug in and work wherever they are—at the desk, roaming in the office, or fully away from the office. [For examples, see the following references: "Walking Away from the Desktop Computer: Distributed Collaboration and Mobility in a Product Design Team," by V. Bellotti and S. Bly, *Proceedings of CSCW '96*, ACM Press, 1996; "Dealing with Mobility: Understanding Access Anytime, Anywhere," by M. Perry et al., *Transactions on Computer-Human Interaction,* pp. 323-347, 2001; "IT Mobility Road Map," Intel Corp., 2002; "Navigating the Future of Software," *Technology Forecast: 2002-2004*, Vol. 1, PriceWaterhouseCoopers, 2002.]

This idea of work mobility presents unique challenges to people who wish to collaborate. In an era long predicted to see the acceptance of the paperless office as standard operating procedure, paper is still the shared technology most frequently used by mobile

Designing Portable Collaborative Networks

workers (and, indeed, for collaborative work in general). [Refer to *The Myth of the Paperless Office*, by A. Sellen and R. Harper, MIT Press, 2001.]

Why? In short, because paper can be used anywhere, while digital technology is still hampered by inflexible networking requirements and rigid design. Networking infrastructure is not always available; security is a constant problem; and even if users are able to connect to each other, their systems aren't set up to make collaboration easy and efficient. To make things even more complex, collaboration tools and devices vary widely, especially when they come from different enterprises.

The underlying problem is that these two issues—mobility and collaboration—have been dealt with as separate parts of the enterprise software solution. As software designers, we need to rethink this problem in two fundamental ways:

- Treat collaboration as a basic capability to be provided at the infrastructure level. Collaboration isn't a specialized task that maps to a particular application; it's a basic process in getting work done.
- Redefine the issue as one of portability instead of mobility. Tools and services should adapt to the user's environment rather than forcing the user to adopt different modes specific to location.

The major design challenges include how to:

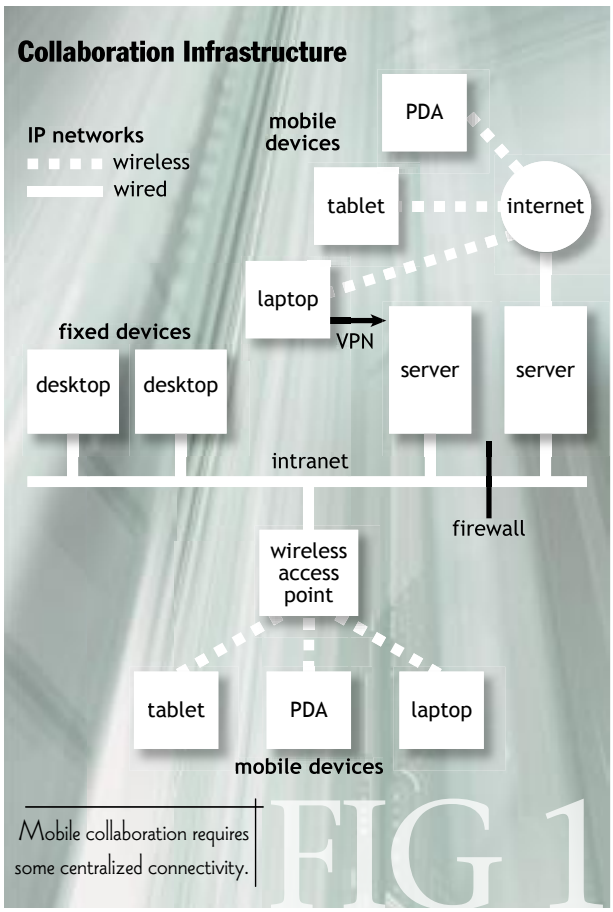
- Provide services that are network-agnostic and adapt gracefully to different network environments.
- Integrate those services and functionality seamlessly with and across existing tools.
- Design services and any new tools to map flexibly to the many actual ways in which people work together, from the structured meeting to the impromptu encounter.

SUPPORTING THE MOBILE INDIVIDUAL

Most enterprise systems and applications now offer some provision for the disconnected worker so that work done offline can be integrated into the centralized system once the worker is again linked to the corporate network or Internet portal.

The standard approach has been to solve the problem of ensuring access to enterprise information in mobile scenarios by one or more of the following:

- Creating a link back to enterprise servers using a combination of dial-up, virtual private network (VPN), middleware, and wide area network (WAN) technologies. The worker downloads, uploads, and synchronizes data in batches when connectivity is constrained by access or performance.
- Using thin clients to increase the device's dependence on constant connectivity, while reducing the need for implementations tailored for specific devices.
- Supporting disconnected work by partially replicating data on capable devices using local partitions. This involves partitioning application logic and tiers to allow both connected and disconnected operation transparently. The problems with this approach are that the data isn't always up to date and a large amount of local storage is required.



Mobile collaboration requires some centralized connectivity. FIG 1

Using these methods means that when two mobile workers want to communicate and share information, they have to be connected to some infrastructure (Figure 1). In many mobile scenarios, that infrastructure is either unreliable or unavailable. Connecting to the local network is often infeasible because of security policies. Even locally connecting to each other may not be enough, because many data-synchronization strategies require server intervention. Moreover, their point-to-point connections are insecure if they are using 802.11* wireless, because its Wired Equivalent Privacy (WEP) security protocol is significantly flawed [see "Wireless Network Security: 802.11," by T. Karygiannis and L. Owens, *Bluetooth and Handheld Devices*, National Institute of Standards and Technology, 2002].

This lack of security constrains public Internet connectivity as well: Using a wireless Internet from home or the airport to connect to the office network requires additional protection. For these reasons many mobile workers still rely on slow WAN connections and periodic dial-up to link back to the office. Without access to some fixed infrastructure, they are denied even the basic collaborative services of communication and file sharing.

COLLABORATION IN THE WORKPLACE

Over the past decade myriad collaboration applications and middleware have entered the workplace, running the gamut from Web-based groupware to component-based frameworks to the increasingly ubiquitous instant messaging (IM) networks. The combination of approaches (both intra- and inter-enterprise) includes e-mail, dedicated groupware platforms such as Lotus Notes, Web-based portals, IM networks, and most recently middleware platforms based on open standards and driven by business rules.

Each of these approaches supports one type of collaborative interaction (formal/informal, realtime/asynchronous, restricted/open), but no single approach flexibly accommodates all the ways in which people work together within and across enterprises. In particular, the standard groupware system tends to be a large, inclusive server-client application hosting the user's work. This means either importing and exporting documents created with other enterprise tools or using the specialized additional tools inside the groupware.

This application approach may map well to marketing (it's easier to sell a stand-alone solution) but doesn't reflect how people work. Collaboration isn't a separate task as much as it is a process and means of accomplishing other primary tasks.

As a result, most collaborative work still involves a combination of tools and services, some of which are poorly integrated with each other, requiring the participants to manage disparate tools and silos of information. People may not want to keep adding to their tool sets, but they are quick to adopt new technologies that help them get their work done when their tools prove inadequate. The most obvious example is the rise in the use of consumer IM, even in the face of corporate resistance. The unpredicted transition of IM from a teenage phenomenon to a standard workplace utility occurred precisely because IM has a more unconstrained and unencumbered interaction paradigm. It allows users to see immediately which of their contacts are available via presence and supports lightweight, instant, informal communication with them without a lot of application overhead or preparation. That informality, however, is in direct conflict with enterprise requirements for robust security and IT-sanctioned participation.

The move toward application integration has seen many large-enterprise frameworks (e.g., customer relationship management, content management, or project management) take the inverse approach and embed, or interoperate with, some basic collaborative services such as e-mail, document versioning, messaging, and threaded discussion forums. Even though information portability is increasingly better defined as Extensible Markup Language (XML) and standard service description languages take hold, support for cross-application paths is still poor without extra middleware. (How do I connect this e-mail to my spreadsheet and then embed them in my workspace?)

A fundamental issue in for both remote access and cross-enterprise collaboration is securely identifying and validating user identities from trusted organizations and being able to provide the appropriate level of access to those users. Robust authentication mechanisms must be provided by either the application itself or an additional centralized identity management service such as Microsoft Passport, IBM Federation, or VeriSign. The major drawback to IM from the IT department's viewpoint has been the lack of secure infrastructure. A flurry of enterprise IM (EIM) products and extensions to consumer IM are now being released with full authentication and encryption capabilities.

These groupware, IM, and enterprise application integration (EAI) approaches have some aspects in common:

- They are server based and rely on a centralized infrastructure.
- Second, each focuses on one of the two flavors of col-

laboration: Either it is good at enabling asynchronous, formal, planned collaboration, or it leans to a messaging approach at supporting realtime, informal, or spontaneous collaboration.

- The collaborative functionality is largely targeted at the desktop or powerful notebook user who has the resources to accommodate the demanding clients.
- These approaches are still limited by proprietary, vendor-dependent protocols requiring gateways and other code to interoperate.
- All of them—including IM—provide only limited support for presence via a server and poor support for discovery. *Presence* technology is a type of application that makes it possible to locate and identify a user on a computing device (including, for example, handheld computers, as well as desktop models) wherever it might be, as soon as the user connects to the network. *Discovery* is the ability to find out about the presence of other users and activities without knowing about them in advance.

WHAT ABOUT MOBILE COLLABORATION?

These systems map poorly to the ever more fluid requirements of the mobile worker, both as individual and as team member. It has been said that the key to mobile collaboration is the ability to work offline [see "Collaboration Comes Together, by S. Sanborn and C. Moore, *InfoWorld*, 2001], but, in fact, the key issue is the ability to share valuable business information in mobile scenarios with those co-located and remote, not necessarily from the same enterprise. Contrary to the portal-centric view that full connectivity to the Internet is always available, the mobile worker's access may vary widely given the place and the worker's role. Furthermore, even if connecting to a colleague from another company is feasible via a link to the Internet, the implication of being on separate corporate networks is that there is no common security infrastructure and, therefore, the worker cannot assure that the connection is secure and that the other person has been sufficiently authenticated.

Collaborative work occurs in many different places under widely varying conditions of infrastructure and foresight. Two people may grab the opportunity to exchange a document in a quick encounter at an airport; an audit team may spend several weeks reviewing data at a client site with limited access to the client's network and the Internet; a sales team may have a scheduled presentation with a group of clients at a trade show while remotely involving experts at other offices; or a consultant may want to share files with clients but they have no common network and security infrastructure. Time and effort are especially important in the kinds of opportunistic scenarios that are frequently characteristic of mobile collaboration. If it is too cumbersome to set up, invite, and authenticate other users in a session, the advantage of collaboration may be lost.

These scenarios are made more complex by the different devices used: laptops, personal digital assistants (PDAs), and the emerging smartphones are not created equal in networking and computing capacity. The large footprint and computing requirements of groupware systems and field Web servers render them infeasible for smaller portable devices.

The mobile worker is often reduced to combining some aspects of the disconnected mode with cumbersome ways to share and update information. Consider one example from an audit team. The members of the team replicated a well-known database from the home office before going onto the client's site. Once on-site, however, they had no way of updating each other's copies of the database with the information they had obtained. Instead, at the end of each day each team member would connect via dial-up to the main server and replicate the individual database. Then once everyone had uploaded the data, each person would have to reconnect in turn to download the now up-to-date database. This example highlights another problem with the lack of transition support: Shifting work patterns and tool use between different infrastructure environments is time consuming and awkward.

Cross-enterprise collaboration introduces other issues. Two users in the same room but on different corporate networks may be able to communicate only via e-mail, an ineffective solution if I want to send you an 8MB file and your e-mail quota is 5MB.

The combination of short-range wireless networking, newer mobile device capabilities, and peer-to-peer archi-

tectures networking can alleviate some of these problems by making it possible to extend the concept of the LAN subnet to ad hoc networks independent of any external infrastructure anywhere, anytime. Workers without any access to the network can establish ad hoc networks on the fly and exchange information; people from different enterprises can connect directly to each other without ensuring that their respective servers can coordinate. Substantial issues of presence, discovery, addressing, security, and privacy have to be addressed, however, before those peer-to-peer networks are usable. These issues are compounded when the goal is to expand past the boundaries of the short-range subnet to include remote peers.

PCN: MIDDLEWARE FOR COLLABORATION

What does it mean to collaborate? The simple definition is "to work jointly, especially in an intellectual endeavor." Operationally, this translates to some or all of the following characteristics:

- **Sharing information and data** (including the creation of new information).
- **Communication, from the casual conversation to the measured discussion.** Sometimes one type turns into another, as when someone takes notes during a telephone conversation and forwards them later in paper form for confirmation.
- **Sharing processes and handing them off sensibly to enable workflow.** (You do this while I do that and Bob can do the next thing.)
- **Sharing context.** Information and communication exist in a given context common to the people sharing them.
- **Presence and activity awareness.** People like to know where people are, what others are doing, and what to expect of them.

Our focus here is to explore issues in the design and development of effective realtime collaboration support for mobile workers and workgroups in both local and remote team situations. Collaboration support is best provided as a pervasive framework underpinning a diverse set of computing devices, networking infrastructures, and software tools.

A major problem with the approach to mobile work as a disconnected process is that it encourages the development of a dedicated strategy to that condition (with check-in, check-out, and replication procedures to learn and follow), rather than extending the notion of where work is done across a gamut of possible infrastructure scenarios. Instead of having each application designer accommodate these scenarios in yet another proprietary way, a middleware layer can provide the infrastructure underpinnings for collaboration without mandating how an application uses them.

This is hardly a novel idea. Current trends in enterprise software development are moving beyond specific application integration to the definition of middleware based on open standards and Web services to stitch together collaborative functionality and existing enterprise application components. Businesses want to manage their existing application components, want business logic to reside in middleware that's vendor independent, and don't want to write new code to integrate each application [see "Collaborative Challenges," by D. Margulius, *InfoWorld*, 2002].

Web-based portals are attaining increasing use as the front end to combining collaboration services and application functionality. Enterprises favor them because they have the potential to hide application boundaries from the user and are the counterpart to back-end application integration. Implicit in this model is the concept that the middleware resides on the centralized infrastructure and the user accesses it via a Web browser or a specialized client. The approach that relies on connectivity to some central point, however, still falls short in supporting all the environments and contexts in which people collaborate. New collaborative software infrastructures—optimized for a gamut of scenarios, tasks, and devices—are required to support these diverse environments.

The challenges in designing these new infrastructures relate not only to where and when users work together, but also to how they integrate the software and services optimized for these diverse environments back into the centralized service environment. Data portability and service frameworks are still in their youth and have many unresolved issues. By definition, because users need to work in a variety of disconnected and connected modes, across a variety of network types and architectures, the infrastructure needs to have both distributed and central components. We term our approach portable collaborative network (PCN). A PCN is a software middleware framework that accommodates diverse networking and device constraints, interoperates with standard enterprise infrastructure, and supports the development and deployment of distributed, collaboration-oriented applications for mobile users. The PCN model draws its inspiration from how people work together (the "group-") rather than from current software parameters (the "-ware"). PCNs can address many of the problems identified in supporting mobile collaboration, but they introduce their own set of design questions.

Designing Portable Collaborative Networks

PCN DESIGN PRINCIPLES

We model the PCN as an organized framework, incorporating:

- People and devices (such as desktops, laptops, PDAs, or phones).
- Collaboration sessions including ad hoc meetings and structured, persistent spaces.
- Context including location, time, purpose.
- Services for collaboration. Basic services include messaging, file transfer, notifications, access control, distributed storage, and meta-data synchronization. These can be extended to support chat, whiteboard, data sharing, personal information sharing, and specialized application sharing.
- Resources (access to peripherals).
- Information (documents, databases).
- Networks (wireless, wireline, ad hoc, administered, peer-to-peer).

According to John Grundy and John Hosking ["Engineering Plug-in Software Components to Support Collaborative Work," *Software Practice and Experience*, pp. 983-1013, 2002], realtime groupware component architectures are based on three pillars:

- Communication support is the low-level framework for exchange of presence, data, and messaging, whether synchronous or asynchronous.
- Coordination support provides event mechanisms for the notification, synchronization, and locking of data, meta-data, objects, and activities.
- Collaborative work support is the fine-grained support for application sharing, view synchronization, and real-time shared work, such as shared editing.

We add a fourth pillar of distributed data management support to provide access to, aggregate, synchronize, and manage information about data objects that reside on various peer devices in the PCN.

Our philosophy is that a flexible lightweight middleware solution should concern itself with communication, coordination, and data distribution support, providing a network and transport layer for developers to implement the more specialized collaborative work support required for their particular applications.

The PCN approach is based on several guiding principles, listed below, each with its own set of challenges and unresolved issues. Because we realize that one of the key challenges in mobile work is how to integrate knowledge from and back into the larger work context, we are concerned with portability rather than just mobility, and with persistent, as well as immediate, collaboration contexts. Finally, following the general principles of vendor-neutral middleware, PCN should be based on open standards.

Heterogeneity. PCN software has to adapt to heterogeneous networking and hardware environments. It runs on a variety of portable, handheld, and desktop computing devices and overlays the underlying network protocol, switching between wired, wireless, local, personal, and wide area networks as appropriate. In the cases of devices with less capacity, how the services are made available will have to be carefully designed and policies established for best use of resources. For example, file-transfer limits on a handheld may need to be curtailed to match the device's capacity.

Appropriate data distribution can collect information about all the shared objects on the network and replicate only the meta-data on smaller devices, providing access to the object without requiring the resources to host it. This will require appropriately flexible meta-data schema to accommodate emerging collaboration protocols such as Web Distributed Authoring and Versioning (WebDAV) and workflow, as well as synchronization mechanisms to manage sporadic connectivity. User interfaces will impose a design challenge to accommodate the widely varying form factors of portable and mobile devices.

Portability and seamless transition between infrastructures. A fallout of the PCN philosophy is that we are no longer proposing a mobile solution. Instead, we are saying that work and work tools involving collaboration should be portable—that is, anywhere you go, you have the tools you need to collaborate, rather than depending on externally supplied infrastructure.

This principle has several related requirements:

- The first is that we need to ensure that the basic infrastructure required for a robust, secure PCN resides in a smart, "fat," non-browser client-side utility so that a group of users can automatically set up an instant peer-to-peer network with the basic communication,

coordination, and data distribution services.

- Second, the PCN should be able to leverage existing infrastructure when it is available and adapt its services appropriately without involving the application or the user unnecessarily. For example, a peer-to-peer connection may be more appropriate for optimal data delivery, but a server may be needed for storing and forwarding messages to offline co-workers.
- Third, the transition between network architectures should be seamless, so that the user or application is not apparently stopping one mode and starting another. The goal is to automate and reduce the overhead of "checking in and out" to the home infrastructure. In the same way that wireless technology released the user from hardware limitations, so that IP-based systems could run on any IP-enabled network (with IP the common protocol), PCN will enable the application to be at least partially agnostic of the underlying network architecture (peer-to-peer or server-client).

In accomplishing these goals, we must resolve several challenging problems that relate to addressing, presence, and communication. Clearly, peer-to-peer technology forms an essential part of this strategy. In a peer-to-peer network, a device advertises itself using presence signals. The range of these announcements is usually constrained to the local subnet, because they can create performance problems as they propagate past the gateway. But peer-to-peer networks have addressing and scope limitations once they extend beyond the subnet, as a result of dynamic IP addressing, network address translation (NAT), and firewalls.

Dynamic IP addresses mean that it is uncertain how a peer can count on locating and finding another, because, unlike a server or computer with a fixed IP address discoverable via the Domain Name System (DNS), that peer will have a different IP every time it joins a network. Because no common peer-to-peer addressing protocols exist as yet, every system handles addressing differently, complicating interoperability. Current approaches include designating certain nodes as advertisement hubs (such as JXTA's rendezvous nodes) or using servers to accept and relay messages (as in Groove).

Addressing is further complicated by corporate Internet security, which may not allow the PCN communication protocol to pass through the corporate firewall. Many firewalls block all but the most common protocols, so the usual approach is to wrap communication streams in the common HTTP/S Web protocol (called tunneling through the firewall). The drawbacks to this are the requirement of an HTTP server to relay messages outside the firewall and the constraints of message payload: HTTP was not designed to enable efficient transfer of large data chunks (as in file transfer). Until more comprehensive addressing protocols such as Internet Protocol Version 6 (IPv6) are in place, some form of server directory and relaying is almost certainly required.

Presence information forms the backbone of peer-to-peer, IM, and realtime collaboration, but integration between these elements is hampered by the lack of any common presence protocol. To date, this has meant that any application that relies on presence must maintain a dedicated presence stream. PCN ensures that the device maintains only one presence stream per PCN, as opposed to per application. We have not seen evidence of a significant movement toward a common presence and IM standard [Instant Messaging and Presence Protocol (IMPP) Work Group, 2002]. Therefore, basing the communication layer on IM is the most likely approach to support vendor-neutral, interoperable middleware. Because all of the major IM networks are server-based, the PCN presence and addressing protocols should include means to contact the peer both directly (if possible) and via the server address. Subsequent presence advertisements will need to be carefully managed to avoid redundant traffic. We anticipate that as peer-to-peer collaboration becomes more mature, industrywide standards will evolve, most likely under the aegis of the International Engineering Task Force (IETF).

A final issue relates to providing IT control over the scope of the PCN. In some cases the IT department will want to restrict access to the intra- or extranet via a corporate server and disable local peer-to-peer access for security reasons.

Appropriate support for context. PCN is founded on a comprehensive model of collaboration that is based on the concept of context and spans both realtime and asynchronous timeframes. People, data, services, and resources can be organized according to context, which can be based on many things: for example, location (everything and everyone in the boardroom) or shared goals (the research project). The PCN context is meant to encapsulate the different kinds of formal and informal collaborations in which people engage, so its services and scope must be configurable. For example, a brief chat between two members of a public ad hoc network is a simple ephemeral context; a workspace with shared resources, distributed data synchronization, and membership restrictions is a persistent one. Organizing contexts around people and defined resources is well understood.

However, contextual interaction based on newer pres-

ence and discovery technology has some exciting implications. Location-based interaction is an obvious candidate ("all the people in the boardroom," "Mike when he's in his office"), as is service-based discovery ("all the people at the trade show with a digital camera"). These imply that context is defined by both static and dynamic attributes. Bluetooth and other short-range wireless sensing technologies hold particular promise for this kind of location-sensitive context establishment.

Presence and discovery services are key to realtime collaboration. Whereas most collaboration systems explicitly model users, devices, and services as entities that are present and active, they do not yet support a broader awareness of the shared contexts and activities. As Jon Udell, a lead analyst with *InfoWorld's* test center, points out, shared spaces, conversations, and activities that are not private should be serendipitously discoverable ["Extending Groove," *InfoWorld*, 2002]. This is especially critical in supporting lightweight, informal interactions that people can spontaneously join, and it is a style of realtime human interaction that is as yet unsupported in any computer-based framework. Therefore, in PCN, a collaboration context is considered an explicit entity in its own right with presence, service, and activity information. A context has both privacy and visibility attributes. It can be defined as private (access is limited to members), public (anyone can join), invisible (a peer needs to be explicitly told how to find it), or discoverable.

Research into discovery methods has been largely focused on service discovery for Web-based computing and business-to-business frameworks [see, for example, "Service Discovery 101," by S. Vinoski, *IEEE Internet Computing*, pp. 69-71, 2003]. Messaging and collaboration systems provide presence information, but the participants need to have somehow registered their interest in (i.e., found) another person. Once found, they add that person (or "presentity") to their buddy list, which is maintained on the server. The server then forwards presence information for each entity on that list to them.

This approach has clear scalability advantages when managing presence information on a large network with hundreds or thousands of users. No user could manage presence information from all others, even if the server were optimized to deliver it. Peer-to-peer systems such as Apple's iChat, Groove, and Colligo Workgroup Edition support automatic discovery on the local subnet: peers' presence information is advertised or polled using IP multicast. As soon as a user is interested in presence information of people beyond the subnet, however, discovery gives way to explicit search using directory servers, placing the burden back on the user. The challenge in PCN design is how to support contextual discovery across the entire reach of the PCN without overwhelming the user by simply returning everyone currently active in the network. No standard discovery protocols exist for presentities, although approaches such as Apple's Rendezvous may hold promise.

For addressing and presence reasons previously discussed, discovery is likely to occur via a hybrid approach, combining local discovery with access to a remote presence server. Because the PCN client will also function in pure peer-to-peer mode, it will have to maintain its own database (buddy list) of saved presentities, adding the extra challenge of synchronizing with the server. Finally, access to discovery information will inevitably have privacy restrictions related to context ("I want to be invisible when I am in the boardroom"). Similar to the model of presence subscription in IM systems, where a user can refuse another user the right to monitor presence, the user must have the right to refuse discovery in defined contexts, and this must form part of the discovery protocol.

Security. Security is the single biggest concern of the mobile professional and the biggest impediment to the wholehearted corporate deployment of mobile data sharing and communication. Collaboration sessions need to be securely authenticated and encrypted by robust and usable methods. The PCN security framework has to be both rigorous and flexible in two ways:

- First, it must be based on the industry standard of public-key infrastructure (PKI). Because it has to function in a pure peer-to-peer mode without relying on access to standard PKI certification authority servers, however, it must be able to provide appropriate peer authentication methods when required and rely on a certification authority (CA) when one is available. (Both Groove and Colligo Workgroup Edition take this approach).
- Second, not all types of collaboration require the same

degree of authentication and privacy. In some situations the user may be quite comfortable interacting with someone using a low level of security (for example, in a casual chat), but may not be willing to engage in more risky interactions (such as opening access to files). The implication of this approach is that security requirements should not necessarily be bound to a user or a context but instead should be coupled to a type of collaboration service. Moreover, the enterprise IT department will want to configure and control these bindings.

Although PKI approaches are not yet interoperable, there is an industry initiative toward making them so [OASIS PKI Member Forum, 2002]. Although computational limitations have restricted the use of PKI, this appears to be diminishing as handhelds become more powerful. This may be one area in which we can safely rely on Moore's Law to solve the hardware limitation.

THE NEED FOR PCN

In addressing the problems of mobile collaboration, we have concluded that a more encompassing framework is required to support collaborative work as it occurs across the gamut of locations and contexts that make up the modern workplace. This approach extends the coming trend of providing component-based middleware to integrate enterprise applications and collaborative services on a wide range of computing devices and networks. We call this pervasive collaborative software layer a portable collaborative network, or PCN.

PCNs support portable and device-independent collaboration. This means that the framework and its associated user interface and APIs must have the ability to scale to different device constraints such as memory and screen real estate, and interoperate at the protocol level among a wide range of devices. Device connectivity configurations must adapt gracefully to different device and networking environments. This may require removing the dependence on fixed servers, for example, and leveraging them only when available or required for presence and discovery. A peer-to-peer architecture may be most appropriate for many portable scenarios.

Because collaboration support is provided as an infrastructure instead of an application, PCNs allow lightweight, specific, appropriate collaborative functionality. Developers may easily integrate its collaboration services into their own products or couple them to existing data, knowledge, or workflow management tools. This means that the collaboration support is effectively transparent. Services are directly integrated into standard office and enterprise tools, as well as being packaged into thin, lightweight utilities that do not get in the way of the user's standard tools and tasks. Therefore, end-user functionality must be integrated into existing user interfaces where appropriate (for example, as a plug-in to Microsoft Office applications). It should also be provided as an API for extensibility by other developers. Some basic utilities and familiar user interfaces for such tasks as basic messaging and file transfer should be supplied.

The result is that user interfaces are intuitive and easy to use. Users employ only what they need, when they need it. Moreover, because the collaboration support is based on context, it more accurately reflects the different styles of collaboration. In particular, presence and discovery of users, activities, and contexts allow people to remain aware of what others are doing without engaging in those activities. This awareness guides their own work strategies and serves as a trigger for spontaneous collaboration around shared resources. This kind of opportunistic collaboration is an important component of all work.

Finally, an adaptive authentication framework based on appropriate existing security standards ensures a secure collaborative network wherever the user is, appropriately tailored to the security needs of the task. Again, this framework needs to reduce the dependence on servers, leveraging fixed enterprise or publicly available key infrastructure for authentication only when available.

Although these principles are derived from our core customer base of mobile work teams, many of the practices they support are equally relevant in more traditional, fixed-location groupware environments. Many of the technologies and standards needed to further the PCN approach are still in their infancy. We anticipate the next few years will prove a fertile testing ground for the PCN concept and the associated development of industrywide protocols and open standards. Q

MICHAEL BLACKSTOCK founded Colligo Networks, a provider of peer-to-peer wireless productivity tools. Prior to that he was vice president of research and development for Infowave Software, a provider of software that connects enterprise applications to wireless devices. He earned a bachelor's degree in electrical engineering from the University of British Columbia and a master's in computer science from Simon Fraser University.

LYN BARTRAM is the senior research scientist at Colligo Networks and adjunct professor at Simon Fraser University, where she earned her Ph.D. in computer science in 1991.

BIO