The Personal Server: Changing the Way We Think About Ubiquitous Computing

Roy Want, Trevor Pering, Gunner Danneels, Muthu Kumar,
Murali Sundar, and John Light
Intel Research
2200 Mission College Blvd
Santa Clara, CA 95054
E-mail: {roy.want, trevor.pering, gunner.danneels,
kumar.muthu, murali.sundar, john.light}@intel.com

ABSTRACT

The *Personal Server* is a mobile device that enables you to readily store and access the data and applications you carry with you through interfaces found in the local environment. Unlike conventional mobile computers with relatively poor user interfaces, it does not have a display at all, instead wirelessly utilizing displays, keyboards and other IO devices found nearby. By co-opting large screens such as those found on desktop PCs, public display monitors, information kiosks, and other computers, a Personal Server is more effective than relying on a small mobile screen. This model goes beyond the mobile context and has wider implications for how we think about computing in general. A prototype system, including applications, system infrastructure, and a mobile platform, has been built to fully explore this model. This prototype sheds light on the suitability of standard components to support such a computing model, and from this illuminates directions for the design of future ubiquitous computing systems.

KEYWORDS: Ubiquitous Computing, Mobility, Device Discovery, Adaptive Interfaces, Personal Server

1 INTRODUCTION

The *Personal Server* [33] is a mobile system designed to enable interaction with a user's personal data *through* the surrounding computing infrastructure: the device itself (Figure 1) possesses no display, instead co-opting the screens and keyboards of nearby computers through a short-range wireless link. This model addresses two major problems associated with mobile information access: the inherent difficulty of using small user interfaces on handheld devices, and the limited access to personal digital information afforded by public access points. In addition to a localized communication capability, the device contains enough high-density storage and low-power, high-performance processing to serve a user's mobile computing and storage needs. The result is that a mobile user can enjoy the benefits of a large display and full-sized keyboard without having to carry a bulky computing platform with them.

The Personal Server aims to overcome the fundamental limitation of cell-phones, PDAs, and laptops: if they're small enough to carry, then the displays are too small to easily use. Fortunately, the computing infrastructure is becoming well established in many of the places we wish to use computation. For example, large-screen PCs can be found in many homes, most businesses, Internet cafes, and even public spaces such as airports and shopping centers. By seamlessly enabling access to *your* mobile data through any of these computing elements, the Personal Server system creates a mobile digital experience based on large-screen interfaces instead of small hand-held displays. Furthermore, since the device possesses no display of it's own, it can be manufactured to be much smaller than a traditional PDA or laptop.

Figure 1: Personal Server Prototype

Our prototype system explores three main aspects of the Personal Server model: the user experience, system infrastructure, and mobile platform. For this prototype, the user experience focuses on simple mobile web pages, file shares, and remote-control applications. Supporting these activities, the system infrastructure must be able to discover and connect to the user's mobile device. The mobile device itself is designed to explore emerging wireless standards, novel power management techniques, and novel device form-factors. Several limitations of existing technologies and avenues for future work have been identified from this research. Specifically, desktop applications do not adequately support dynamic usage by mobile users, and the discovery capabilities of existing

wireless standards do not adequately support personal mobile devices.

This paper contributes an understanding of the basic technologies and concepts necessary to support aspects of ubiquitous computing [36] not tied to a specific location, instead associated with the mobility of a user. The Personal Server concept pushes on the boundaries presented by traditional devices, utilizing the resources afforded by the local computing infrastructure to provide a compelling mobile user experience. In the long run, the emerging user experiences, system components, and device technologies can either be used to augment existing mobile platforms or to create an entirely new class of mobile device.

2 MOTIVATION

The Personal Server is motivated by the emergence of high-density storage and low-power computing, which can be used to address the frustration many users have with mobile computing by allowing access to mobile data through the fixed infrastructure. Emerging storage technologies can easily accommodate 1GB of data in one square inch of disk space, with capacities doubling annually [28]. Instead of keeping your 'real' data on a big network server, you can carry it all with you or treat the mobile device as a large data cache [20]. Further, mobile processing is advancing nearly as fast, allowing us to carry computation with us as well. Based on these capabilities, the Personal Server addresses the limitations of existing mobile devices by leveraging the existing computing infrastructure (Figure 2), providing the interface of a desktop system wherever you carry your mobile.

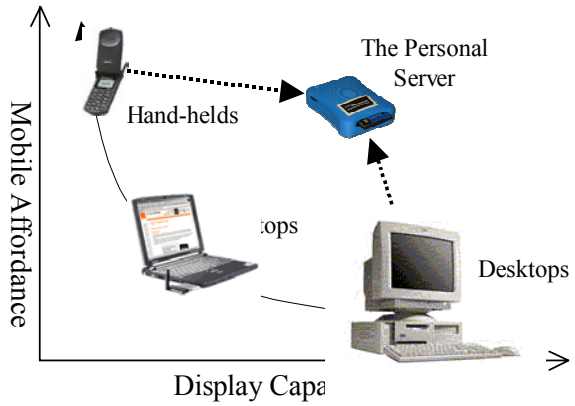

Figure 2: The Display Capability and Mobile Affordance of various platforms. The Personal Server enables the use of large displays from a small form-factor mobile device.

This system incorporates the positive aspects of today's mobile platforms by combining the interface capability of a desktop in the form factor of a mobile. Of existing devices, laptops arguably provide the most complete mobile experience because they allow users comprehensive access to computing resources wherever they may travel: desktops are simply not mobile enough and handhelds are not capable enough. Although laptops do have a "full-sized" screen and full-featured keyboard, they are still too large (typically a 10.4in diagonal) and heavy (~5lbs) to carry with you at all times. Instead of trying to improve the UI capabilities of small mobile devices, the Personal Server circumvents the problem by making it easy to use the surrounding computing infrastructure. Wearable computers, which can provide laptop-level functionality without the size and weight, rely upon a literally "in-your-face" heads-up display that many people find too invasive to be acceptable.

The Personal Server provides low-latency always-available access to a user's data over a short-range wireless network, instead of relying on a remote server with the associated problems of network outages, high-latencies, and high access fees. Basic availability is very important in many mobile situations: for example, a traveling sales person who needs to present information about their products to a customer. Relying on remote access for wide-area mobility is dangerous because a network connection may be unavailable, restricted by firewalls, censorship, or other factors out of your control. Even assuming a working connection, access latencies may be high due to network congestion, multi-hop routing, or because your data does not have enough priority in the network. In addition to spotty coverage and variable latencies, wide-area wireless network access also brings the element of cost, which can be prohibitively high. This model inverts the popular "thin-client" paradigm, which works well for well-connected local environments but fails in mobile contexts.

Fundamentally, the Personal Server is about accessing information through the most *convenient* interface available, instead of the one interface dictated by a particular device. For example, a user may be able to borrow a PDA from somebody sitting next to them, pull a display tablet out of their bag, *or* choose to walk over to a nearby information kiosk. Similarly, this system would enable a seamless transition between a user's home and office setup without having to explicitly mange files and connections between the two. The Personal Server does not replace existing mobile technologies such as PDAs or cell-phones, instead working with them to provide flexible access to personal resources. Generalizing this capability leads to the notion of "scrap" display devices, which are treated much the same way as pens and scrap paper, i.e., as communal property with no explicit ownership.

Appropriately enough, many applications don't even require immediate use of a display, instead allowing

interaction to be deferred until convenient or necessary: a natural usage model for the Personal Server. For example, emerging wireless standards will enable a device to remember the menu du jour broadcast by "passed by" restaurants. This collection of menus could then be used later on when the user is actually hungry. Similarly, a *proactive* system may continuously monitor the local context and only notify the user when an appropriate trigger occurs, e.g., when a nearby store is offering a specific item on sale. For the most part, these applications are constantly capturing the local context and only require interaction when certain conditions are met; therefore, the user is free to wait until they can access the data though a convenient interface.

3 RELATED WORK

Related work can roughly be divided into two categories: server-based systems and mobile solutions. Server-based architectures assume that a user's "home server" is reliably available over a network, while mobile systems carry all the data locally and only periodically sync-up with a centralized repository. Similar to mobile systems, there are several compact devices that provide access to personal storage when attached to another system, but these are simply storage devices and do not provide any processing capability.

Server-based systems must fundamentally deal with how to transfer data between the server and the client over a network. Many "thin client" approaches, such as web-browsers, InfoPad [29], ParcTab [32], VNC [9], Roma [26], et. al., assume a low-latency, relatively high-bandwidth connection between the client and home server, which may not be a valid assumption in mobile contexts. Internet suspend/resume [8] assumes the availability of local computing resources and proactively migrates computation from the home server to a local surrogate, mitigating the effects of a potentially slow network connection. Even WAP [34], which is nominally a mobile-device system, is fundamentally server-based because it assumes all significant processing and storage is accomplished on a remote server accessed over the cellular wireless infrastructure. The Personal Server model differs from these approaches because data and computation are made available locally, and information can be quickly and conveniently accessed, without requiring "on-demand" data migration. In order to reconcile data with a central server, the system could use some kind of distributed file system technique, such as Coda [20].

Mobile systems, such as cell-phones and PDAs, suffer from the inherent difficulty of accessing data through small-screen displays. Systems such as mLinks [21] and gestures for mobile devices [14] explore alternate browsing techniques and input modalities, respectively, to overcome these limitations. In contrast, the Personal Server utilizes external-computing elements, i.e., "borrowed" displays. The MetaPad [11] is a mobile computation and storage device that can be fitted with many different display sleeves (i.e., full size monitors, PDA screens, tablets, etc..); however, this requires physically placing the device in a sleeve, while the Personal Server *wirelessly* interfaces with the interaction elements, which yields a fundamentally different user experience and system requirements. Wearable systems [24] rely on heads-up displays and hand-held keyboards to provide an interface, which is literally too "in your face" for many users.

A string of very small devices are very similar to the Personal Server in that they don't possess a significant display and instead rely on nearby computing infrastructure. Disk-on-key [18], iPod [7], Pockey [17] are compact storage devices that connect to desktop systems through a USB or Firewire interface; however, these devices have no active component and, similar to the MetaPad, need to be physically connected to operate. Factoid [10] is a wireless device that records little bits of data received from the environment, which can then be accessed later from fixed systems, but it is not powerful enough to support a full-featured digital experience.

4 SYSTEM OVERVIEW

The Personal Server implementation can be divided into three conceptual levels, each with specific research challenges:

- **User Experience**: creating a compelling usage model that overcomes the difficulties of mobile interaction.
- **System Infrastructure**: quickly discovering and supporting mobile devices in dynamic wireless contexts.
- **Mobile Platform**: developing a physically unencumbering device that is capable enough to be a user's primary computing/storage device.

The three essential characteristics for a compelling mobile experience are rich interaction, seamless interaction, and information availability. Rich interaction is accomplished by utilizing the surrounding infrastructure, which must therefore contain the supporting software necessary for a user to access their personal device. Seamless interaction is enabled by wireless technologies, allowing information access without an explicit "connection" step. A wired solution, such as the USB standard, is fairly troublesome because it requires the user to manually locate sockets needed for connectivity and then plug/unplug the appropriate cables. Furthermore, as described in the

motivation, information availability necessitates the physical device itself. These three criteria predicate the research challenges outlined for the system infrastructure and mobile platform.

The system infrastructure is needed to support two main functions: discovering mobile devices and providing interface and computation support for mobile users. First, seamless connectivity implies the use of an automatic discovery protocol initiated by the *infrastructure*, allowing a user's device to be aware of its surroundings without unnecessarily giving away its presence. This discovery process sets up a local ad-hoc network, often referred to as a Personal Area Network (PAN), for generic communication between the interface and mobile device. Second, the borrowed interface must support access to the mobile through generic mechanisms such as a network file system or web-browser. Furthermore, once a file system is available, standard application software available on the host can be used to view or edit a user's data.

In addition to basic shared-drive and web-browser access, a limited form of wearable interface needs to be provided for situations where no other input mechanism is available, such as when a display is mounted high up or behind a shop window (Figure 3). Therefore, the system should support a limited interface back channel from the mobile device to the public display. Furthermore, this interface does not necessarily need to be *physically* associated with the mobile device itself, instead, for example, being incorporated as part of a user's wristwatch. This limited wearable interface would be insufficient for data-entry tasks, but quite adequate for most simple information navigation tasks.

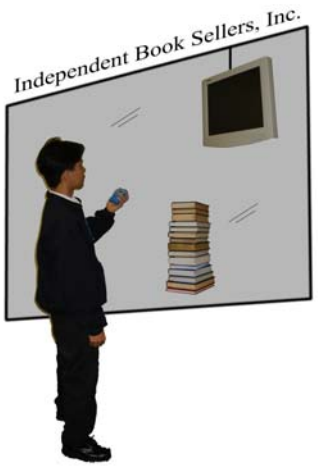

Figure 3: Interacting with inaccessible displays, which might be behind a store-front window.

The key to the hardware platform is designing a compact low-power system that provides enough resources to sufficiently support a mobile user. The wireless implementation must provide sufficient bandwidth while being low-latency, low power and physically small. Furthermore, in order to be useful it must connect to a wide variety of devices and coexist with other wireless users. The overall size of the device is dictated by its power consumption – more power means a bigger battery. Therefore, all components must be designed with power as a primary consideration. Additionally, a compact design with no significant interaction requirements opens the door to novel and interesting industrial design opportunities.

5 IMPLEMENTATION

The initial Personal Server prototype is based upon existing technologies whenever possible to both increase compatibility with conventional computing environments and ferret out problems with existing standards. The implementation is described in reverse order from that of the previous section, starting with the base hardware platform leading to a description of the resulting user experience.

5.1 Mobile Platform

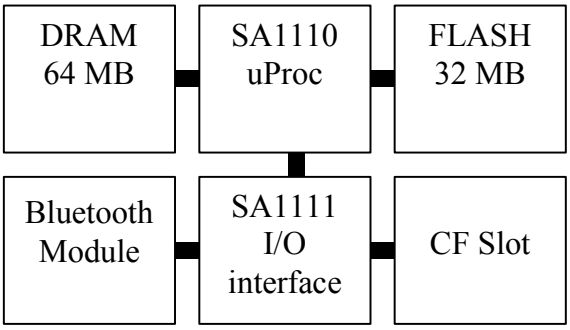

Figure 4: Personal Server hardware architecture.

The mobile platform (Figure 4) is comprised of processing, communication, and storage subsystems, which are combined together to form a compact, low-power mobile device (Figure 5). The design is based on the StrongARM SA1110 running the Linux OS, a combination available for many popular hand-held PDAs. Bluetooth [3] provides the wireless communication capability, while storage is provided through of on-board FLASH, DRAM, and a removable Compact Flash (CF) slot for user-specific storage. Power is provided through a 920mAhr Li-ion cell. Additionally, to provide the remote control functionality, the Personal Server has a jog dial and two buttons.

StrongARM and Linux were chosen because they are widely supported and provide attractive power and performance characteristics. The StrongARM processor nominally operates at 206 MHz, consuming 650 mW during active operation, and it also possesses several low-power operational states, including the ability to run at lower clock frequencies. Furthermore, when supported by the StrongARM-1111 companion chip, it provides all of the base I/O requirements, such as managing the CF slot and the on-board USB connection to the Bluetooth subsystem. Linux provides open-source support for all the necessary system components, only slightly modified to support our specific hardware configuration.

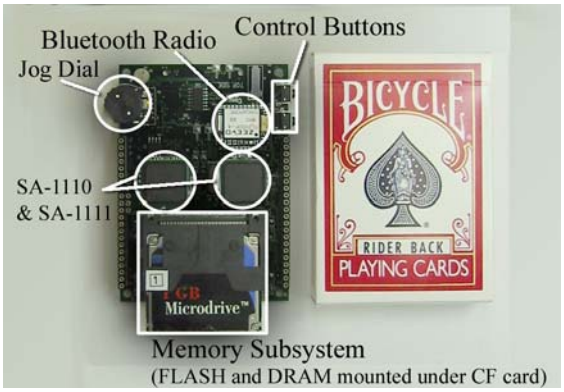

Figure 5: Personal Server hardware device.

An on-board Bluetooth module provides device discovery, short-range connectivity, low-power operation, a one-chip solution, a 723kbps maximum application data rate, and coexistence with other radio standards. Besides connecting with a "heavy-weight" PC-oriented interface, it also enables connection to a wide variety of smaller electronic devices, such as printers and audio headsets, which are currently being released for the consumer market. The device's Bluetooth software stack is based on the standard Linux OS release that has been augmented with an in-house implementation of the Bluetooth PAN profile (since released to open-source), providing basic TCP/IP functionality. The system also provides a direct connection to emerging sensor networks [5] or wearable systems.

The on-chip memory subsystems supports 32 MB of FLASH, 64MB of DRAM, and a CF memory slot, enabling external storage devices of *at least* 512 MB. Currently, only 16 MB of on-board FLASH is used, providing basic support for booting Linux and establishing wireless network connectivity. The CF slot can be used for FLASH-based devices, micro-drives, or even for experimenting with other wireless standards such as 802.11 [6] using commercially available cards.

The complete system is approximately the size of a deck of cards, and can be encased in an attractive enclosure, as shown in Figure 1 on the first page. The complete design weighs 135g, including prototype case (40g) and battery pack (55g). The current system dissipates between 1mW and 700mW, depending on system activity. The optimization of this power consumption is the subject of future research.

5.2 System Architecture

The supporting system architecture (Figure 6) can be divided into two main pieces: device and infrastructure. The software on the Personal Server responds to discovery requests, as per the Bluetooth standard, and supports basic web-services, using an on-board Apache web-server, plus administrative and remote control functionality, through a custom daemon. The host infrastructure is implemented using a standard Windows XP system augmented with a Bluetooth stack and custom middleware components. The host infrastructure is responsible for discovering mobile devices and providing common access protocols. Once an IP connection is established, the various software components shown in Figure 6 are activated. Ideally, this common access layer would be present on all systems, allowing access to a Personal Server from any desktop.

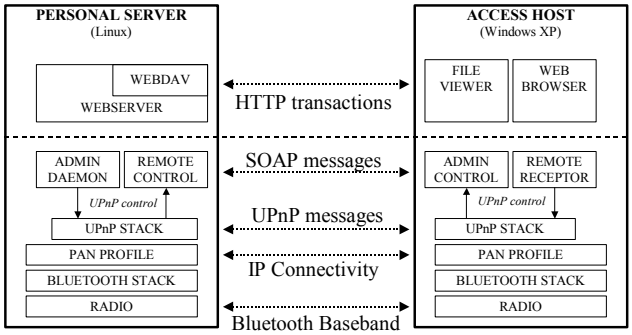

Figure 6: Overview of the Personal Server communication architecture.

In order to facilitate device discovery, a discovery monitor runs on the host and periodically searches for devices, automatically loading the Bluetooth PAN profile when one is found. The basic Bluetooth discovery scan takes approximately 2.56 seconds, which is repeated 4 times back-to-back, as recommended by the Bluetooth spec, every 20 seconds.

Once a networking connection is established, the Personal Server supports three capabilities through the UPnP [30] infrastructure: web services, which provides access to a WebDAV [35] file share and web server, remote control, which provides a mechanism to send user interface events from the mobile to the host, and administrative daemon, which enables access to sharing setup, password control, and basic device information such as memory or battery usage.

On the host side, there are three corresponding elements that handle interaction with the Personal Server: standard web browser and file explorer applications, a remote-control target, which takes commands from the Personal Server and pushes them to applications through the Windows input queue, and an administrative control, which provides access to the administrative and information capabilities.

Communication between the device and host is based upon standard Web protocols such as UPnP, SOAP [23], and HTTP, which are layered on top the IP networking enabled by the Bluetooth PAN profile [13]. Basic UPnP setup is accomplished by a UPnP "device"(i.e., "Administrative Daemon" and "Remote Target") broadcasting a service description which is picked up by the respective UPnP "control point" (i.e., "Admin Control" and "Remote Control"). SOAP is used to directly communicate between these entities. Along with WebDAV and web server access, it is also uses the standard HTTP protocol.

5.3 Applications and Usage Models

Three basic usage models, supporting many different applications, are provided by the initial prototype: web browsing, wireless file access, and remote control. In fact, once these three capabilities are in place, the system behaves much like any other networked system, with the expected caveat of network bandwidth. Given a standard desktop system is used as an access point, the user can easily give presentations or slide-shows from their Personal Server.

Overall, the basic system allows a user to walk into a room with their Personal Server device and relatively quickly start using an enabled access point: it only takes about 21 sec for the system to discover and mount the wireless file system. However, given that the discovery process can start when the user is 10 meters away from the intended host, this process could be completed before the user reaches the terminal.

Based on the web-browsing and file share capability, the system affords the user quick access to their mobile personal page, which can be set up to provide bookmarks, an address book, etc… This is semantically different than a user's public web home page, which typically contains items such as a list of hobbies, a self portrait, or general "what the public should know" information. Since the host is situated between the Personal Server and the greater Internet, assuming it is connected, then any link browsed from the user's mobile home page will connect directly to the destination site, *not* going through the user's mobile device.

The canonical example using the file share and remote control capabilities together is that of a presentation sourced from the mobile device. First, the host automatically discovers the Personal Server device and exposes the public file share, thereby allowing the user to launch their presentation and advance through their slides using the remote control capability. Even for large presentations, the access time is not unreasonable: a 1 MB file takes about 20 seconds to load and display at an effective 400kbps transfer rate. Since the actual application code is provided by the host, the mobile device does not provide anything besides the raw presentation data and remote control commands.

6 Architectural Issues & Future Work

The formulation of the Personal Server model and development of the initial prototype has uncovered several issues relating to the design of the Personal Server system. Some of these issues indicate where existing standards are inadequate, while others are opportunities to examine previous research in a new light – creating new challenges and solutions.

6.1 Discovery

The entire connection process takes about 21 seconds, due to the combination of the wireless and service discovery layers. Although tolerable for the basic Personal Server usage models, this delay does not adequately support the desired seamless interaction model. Some of this delay is inherent to the layered technologies employed by the system, while others are due to the layering itself. An out-of-band radio channel could provide an alternate discovery mechanism specifically tuned for mobile discovery, thereby reducing the delay.

The recommended Bluetooth device discovery mechanism takes approximately 10 seconds, during which the inquiry device can handle other radio traffic. One solution to mitigate this delay would be for an "inactive" host to continuously inquire, thereby detecting a new device as soon as it came into range; however, this unnecessarily pollutes the radio spectrum with discovery traffic. Furthermore, a mobile user could easily pass through the wireless cell (approximately 10m diameter) before the discovery process is complete. Complicating matters further, the discovery process is quite power-hungry, which would be a problem for setting up peer-to-peer topologies.

One possible solution to this problem would be to use a separate out-of-band radio for the discovery process. This radio, which could be lower power and lower-speed than the primary communication channel, would be dedicated to discovering nearby devices. Outside of basic discovery, this channel could be used to provide the necessary information to *quickly* set up the primary link without performing the entire discovery sequence.

6.2 Privacy & Security

Any mobile device that is continually communicating with the environment is a potential threat to a user's privacy, an issue uncovered by the Active Badge system [31], but compounded by the Personal Server because it *requires* you to interact through the environment to be useful. Part of the solution is to make the mobile device passive with respect to the discovery process: the environment broadcasts its services, and the mobile device only responds when necessary. Eventually, however, the user is likely to want something from the environment, at which point they will have to reveal something about themselves. To protect a user's privacy, it will be important to fully consider workgroup dynamics and personal preferences.

Similar to privacy, although subtly different, is the security of a user's data as they interact through the public infrastructure, which has the fundamental problem that it could be compromised by an unscrupulous third party [1], thereby allowing partial access to the information stored on a user's device. Furthermore, a large-screen display in a public place may be easily viewable by more than one person, and so the person standing next to you may be able to see what you are doing on the screen. Some solutions may be social, such as a personal rule that you don't view sensitive information on a display when somebody else is standing next to you, while others may be technical, such as secure public authentication [16] or enabling the system to be aware of the security of the display and appropriately adapt the available content [19].

6.3 Adaptive User Interfaces

In order for a user to co-opt "the most convenient" interface, applications must be able to adapt their UI to a variety of display sizes and modalities. Currently, interface designers must create "one-off" interfaces for each desired display, i.e., a separate interface for desktop browsers, cell phones, PDAs, etc.. Although tolerable, the Personal Server paradigm exacerbates this problem by *relying* on a dynamically changing set of devices, bringing the UI adaptation problem to the forefront. Various systems such as ICrafter [22], PIMA [2], and the Composite Capabilities/Preferences Profile [4] all address this problem to some degree. Furthermore, there is an opportunity to adapt interfaces based on the security or accessibility of the display, not just its properties.

6.4 Usage Models

In addition to the basic usage models and applications supported by the current prototype, there are several scenarios that are well suited to the Personal Server model and still need to be fully explored. For example, the device could serve as central hub for wearable peripherals, such as a remote control watch, health monitoring device, or wireless headset (and the Personal Server could perform the requisite speech recognition). Additionally, by monitoring the discovery inquiries from local infrastructure, it could act as a "data sponge" for serendipitously accumulating information, which could be useful if the user later decided they had a need for information from where they *had been*. Although these concepts are not entirely new, the Personal Server provides a concrete platform on which to implement and evaluate them.

6.5 Power Management

As with any mobile device, power consumption is a primary concern of the Personal Server platform. Apart from choosing low-power subsystems like the processor and communications module, there are several avenues of power management available. The power consumption profile of the device must support an "always-on" usage model, where it is continuously available for wireless discovery, which is a different model than supported by a PDA that is manually turned on and off. Furthermore, the power consumption of the (non-existent) LCD display is not an issue with the Personal Server device, and so the processing and communication subsystems have a greater impact on the overall device battery lifetime.

There are several techniques available to reduce the power consumption of the processor and communication subsystems in support of the "always-on" usage model. For the processor, Dynamic Voltage Management [15] enables the processor to self-regulate its operating speed and save considerable amounts of energy while still continuing to operate, instead of just putting itself into a non-active sleep mode. A hierarchical out-of-band discovery mechanism, as discussed previously, or routing management protocols [25] can be used to reduce the effective power consumption of the radio subsystem, which can be easily dominated by "listen time", when the device is simply waiting to be discovered. Techniques such as closed loop power monitoring, which allows the processor to actively monitor the power consumption of individual system components, or mobile agents, described below, can be applied to the system as a whole.

6.6 Distributed File Systems

Although a Personal Server device can potentially store a user's entire personal data collection, a distributed storage system such as Coda [20] or Bayou [27] would be very useful in case of theft, loss, damage, or concurrent access. Using such a system, the user could modify their mobile data while disconnected, and automatically transfer changes to the infrastructure when able. Such a capability would allow easy recovery in the case when a device becomes inaccessible. Similarly, these systems would allow data to be directly modified in the infrastructure,

eventually propagating to the mobile device. Without such a capability, the user would be required to manually manage their data backup and migration, which would significantly detract from the user experience.

6.7 Mobile Execution

Allowing code to be executed either on the mobile device, or in the supporting infrastructure, enables the system to optimize for power consumption, latency, and availability. For example, if a compute intensive task can be executed in the environment, then it will consume power from the infrastructure rather than the mobile device itself. Similarly, executing a user interface component in the environment will improve the interactive response. However, the mobile device needs to be prepared to execute all the code locally in case the infrastructure is not able to support the desired computation. This capability is enabled by dynamic execution systems [12], but the Personal Server model introduces a different application model, where the interface is separated from the storage and (sometimes) the computation resource.

6.8 Industrial Design

Since the Personal Server device has no display, the entire system could theoretically be integrated into a very small single chip module; however, until a renewable energy source is made available, users will still have to "interact" with the device to recharge its energy supply. The basic challenge is to position the device such that it is not noticeable or encumbering to the user, such as a watch or piece of jewelry, while being "important" enough that the user does not lose it and can remember to charge it. For example, the device could be housed in a user's shoe, and then placed in a special charging shoe rack during the night to recharge. Fully exploring how the device seamlessly fits in with a user's physical world will be a critical part of the overall Personal Server experience.

7 CONCLUSION

The Personal Server model changes the basic assumptions that we make about the limitations of mobile devices and the way we think about accessing our data in a computationally rich world. By utilizing computers in the local environment, it will considerably improve the mobile user experience and allow the mobile device itself to "disappear" from a users interaction. Although it has been possible to build a basic Personal Server device for some time, it is only now that four enabling factors have come together to make it practical for mainstream use:

- High-density portable storage devices.
- Low-power, high-performance processors.
- Short-range, low power wireless standards.
- Extensive public computing infrastructure.

The initial prototype system combines the three technology advances with the extensive infrastructure to provide a mobile system that supports a user's mobile computing needs without forcing them to interact through a small-screen device. This system has already demonstrated the feasibility of the Personal Server concept, and the supporting trends will only become stronger over the next few years. The initial experience using the device to remotely control a presentation sourced from a wireless storage device has proved a compelling experience and demonstration. Going forward, it will be necessary to expand the system and deploy it to a larger audience to adequately develop, explore, and evaluate the complete Personal Server experience.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the contribution of Steve Swanson, Jim Kardach, Graham Kirby, Rebecca McKinney, Peter Adamson, Ryan Macaluso, Sandeep Chivikula, Alexander Chou, Neil Yang, Paul Wright, Shad Roundy, and Alex Nguyen – all members of the extended team contributing to the implementation of the first Personal Server system.

REFERENCES

1. Anderson, R.; "Why cryptosystems fail", *Communications of the ACM,* November 1994

2. Banavar, G.; Beck, J.; Gluzberg, E.; Munson, J.; Sussman, J.; and Zukowski, D. "An Application Model for Pervasive Computing". *Proceedings of. ACM MOBICOM*, Boston, MA, August 2000. Also found at http://www.research.ibm.com/PIMA.

3. Bluetooth SIG, http://www.bluetooth.com/, April 2002.

4. CC/PP project home page. http://www.w3.org/Mobile/CCPP.

5. Hill, J.; Szewczyk R.; Woo, A.; Hollar, S.; Culler, D.; Pister, K.; "System architecture directions for networked sensors". *Ninth international conference on Architectural support for programming languages and operating systems*. November 2000.

6. IEEE 802.11b, WiFi Standard, http://grouper.ieee.org/groups/802/11/index.html, April 2002.

7. iPod: apple's MP3 player http://www.apple.com/ipod/

8. Kozuch, M., and Satyanarayanan, M., "Internet Suspend/Resume," *Proceedings of the Workshop on Mobile Computing Systems and Applications,* Callicoon, NY, June 20-21, 2002

9. Li, S. F.; Spiteri, M.; Bates, J.; Hopper, A. “Capturing and Indexing Computer-based Activities with Virtual Network Computing”. *Proceedings of the 2000 ACM Symposium on Applied Computing*, Como, Italy, Vol 2. Pages 601-603, March 19-21, 2000

10. Mayo,R., The Factoid Project (Compaq WRL technical report) http://www.research.compaq.com/wrl/projects/Factoid/factoid.html, April 2002

11. MetaPad: http://www.research.ibm.com/thinkresearch/pages/2002/20020207_metapad.shtml

12. Noble, B.; Satyanarayanan, M.; Narayanan, D.; Tilton, J.; Flinn, J.; Walker, K.; “Agile application-aware adaptation for mobility”. Proceedings of the 16th ACM Symposium on Operating System Principles, October 1997.

13. PAN – Personal Area Networks. http://grouper.ieee.org/groups/802/15/index.html IEEE 802.15, May 2001.

14. Pirhonen, A.; Brewster, S.; Gestural, “Audio Metaphors as a Means of Control for Mobile Devices”. *Proceedings of the SIGCHI conference on Human factors in computing systems*, April 2002

15. Pering, T.; Burd, T.; Brodersen, R., “Voltage scheduling in the lpARM microprocessor system.” ISLPED, July 2000

16. Pering, T.; Light, J.; Sundar, M.; Want R., Photographic Authentication through Untrusted Terminals. *Intel Research Technical Report,* April 2002

17. Pockey: pocketable disk drive, http://pockey.co.kr

18. ‘Q’ Drive, Agate technologies inc, http://www.eiwww.com/products_q.html, May 2001.

19. Ross, S.J., et. al.; “A Composable Framework for Secure Multi-Modal Access to Internet Services from Post-PC Devices”. *Proceedings of the Third IEEE WMCSA*, December 2000.

20. Satyanarayanan, M. "The Evolution of Coda", *ACM Transactions on Computer Systems,* Volume 20, Number 2, May 2002.

21. Schilit, B.; Trevor, J.; Hilbert, D.; Koh, T.; m-links: An infrastructure for very small internet devices, *Proceedings of the seventh annual international conference on Mobile computing and networking (MOBICOM)*, July 2001.

22. Ponnekanti, S.; Lee, B.; Fox, A.; Hanrahan, P.; Winograd, T; “ ICrafter: A Service Framework for Ubiquitous Computing Environments”. *Proceedings of Ubicomp 2001*, pp56-75.

23. SOAP: Simple Object Access Protocol http://www.w3c.org/2002/ws/

24. Starner, T., "The Challenges of Wearable Computing: Part 1 & 2." *IEEE Micro* 21(4), July 2001, pp. 44-52 & pp. 54-67.

25. Singh, S.; Woo, M.; Raghavendra, C. S., “Power-aware routing in mobile ad hoc networks”,. *The fourth annual ACM/IEEE international conference on Mobile computing and networking (MOBICOM)*, October 1998

26. Swierk E.; Kiciman, E.; Williams, N.; Fukushima, T.; Yoshida, H., L.; and Baker, M., "The Roma Personal Metadata Service." To appear in *Mobile Networks and Applications (MONET)*, volume 7, number 5, September/October 2002.

27. Terry, D. B.; Theimer, M. M.; Petersen, K.; Demers, A. J,; Spreitzer, M. J. and Hauser, C.; “Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System” *Proceedings 15th Symposium on Operating Systems Principles*, Cooper Mountain, Colorado, December 1995, pages 172-183.

28. Toigo, J., W., “Avoiding a Data Crunch”, *Scientific American*, May 2000, Vol 282, No. 5 pp58—74

29. Truman, T.; Pering, T.; Doering, R.; Brodersen, R., “The InfoPad Multimedia Terminal: A Portable Device for Wireless Information Access”, *IEEE Transactions on Computers*, October 1998, Vol. 47, No. 10

30. UPnP, “Understanding Universal Plug and Play”, Microsoft white paper available at http://www.upnp.org

31. Want, R.; Hopper, A.; Falcao, V.; Gibbons, J., “The Active Badge Location System", *ACM Transactions on Office Information Systems (TOIS)*, Vol. 10. No. 1, Jan 1992, Pages 91-102

32. Want, R.; Schilit, B.; Norman A.; Gold R.; Goldberg D.; Petersen K., Ellis J., Weiser, M., "An Overview of the Parctab Ubiquitous Computing Experiment", *IEEE Personal Communications*, December 1995, Vol 2. No.6, pp28-43.

33. Want, R.; Pering, T.; Borriello, G.; Farkas, Keith I.; “Disappearing Hardware”, *IEEE Pervasive Computing*,Vol. 1. Issue #1, April 2002, pp36-47.

34. WAP Forum: http://www.wapforum.org/

35. WebDAV http://www.webdav.org

36. Weiser, M., "The Computer for the 21st Century", *Scientific American*, September 1991, Vol. 265 No. 3, pp94-104 .