# The Athena Palladium Print System

Brian Handspicker
bd%vines.DEC@decwrl.dec.com

Richard Hart
hart@decvax.dec.com

Mark Roman *
marcus@athena.mit.edu

December 9, 1988

**Abstract**

Large-scale application of traditional print systems to distributed computing environments and modern printing hardware have exposed the limitations of those print systems. *Palladium* is a distributed printing system being developed at Project Athena [Athena85] to address these issues. It will supply both distributed access to printing services and centralized management of those services as required by Project Athena. The interfaces to these *Palladium* services are designed to conform with developing international standards. *Palladium* supports the features found on current and recently developed printing devices, and provides, in a distributed environment, the print job queuing and management that users have come to expect.

## Introduction

Recent advances in document publishing products and printer technology have outstripped the support provided by traditional print systems. Users may create documents with complicated formats, fancy fonts, and embedded graphics on single user workstations or PC's. When these advances in printing capabilities are combined with the new distributed computing environments now in use, it becomes evident that printing systems need to be redesigned.

The Athena Palladium Print System has been designed to provide equitable access to shared printing resources in the Athena [Athena85] computing environment. Athena is a distributed computing environment for use by all students, faculty, and staff at MIT. It currently consists of about 800 hundred workstations and 53 file servers. It is planned that Athena will grow to a few thousand workstations.

The Athena system is based on the BSD 4.3 operating system. BSD 4.3 was originally developed for an environment consisting of a central processor with a well defined collection of user terminals and other devices attached. To adopt BSD 4.3 to the distributed Athena environment, changes were made, and important pieces of software were added [Athena85, Hesiod88, Kerberos88, Zephyr88].

Because the Berkeley Spooler [Campbell83] was written for a central processor environment, it has certain shortcomings that make it difficult to use, or that are unnecessary, in the distributed Athena environment:

- The desirability of placing printers, users, and spool queues all on different machines in a network was initially unforeseen.

*MIT Project Athena, 1 Amherst Street E40-300, Cambridge, MA 02142

- The Berkeley Spooler does not easily support the centralized management and logging needed at Athena.
- Tradeoffs were made based on the assumption that very little memory is available.
- It assumed that printers only read, they never write.
- Authentication of users was not needed.
- Access controls were not necessary for printers and queued print jobs.

The goals of the Athena Palladium Print System were developed to address these issues and others. This paper describes those goals and it then shows how distributed systems design concepts have been applied to meet them.

## Palladium Print System Goals

The goals of the Palladium project at Athena can be split into the following categories:

- Computing Environment. Palladium is being designed for use within the Athena [Athena85] distributed environment. Within this environment, Palladium users, print queues, and printers, may all exist on different systems. A user may submit a print job from one workstation, then make inquiries about that job from a different workstation. If a printer malfunctions, jobs originally scheduled for that printer may be rescheduled on some other printer on the network.
- Printer Support. Palladium must eventually support many types and brands of printing devices. Of particular importance are newer printers that both receive text to be printed, and that return status and control information. Also, it must be possible to support any and all page descriptions languages. At Athena, the page description language of choice is Postscript [PostScript87], but others will be supported.
- System Management. Centralized system management is an important feature of any distributed system having hundreds (and eventually thousands) of workstations. The Athena system is no exception. Palladium will make all printer and print server state information available through standard interfaces. It will be possible to stop and to start individual printers and print servers remotely, without changing the availability of any other printing services. It will also be possible for operators to redirect print jobs to different printers from those originally specified for the print job.
- Authentication and Accounting. Palladium will allow no access to print system services and information without proper authentication. It will also be possible to log all printer usage and to set quotas to control that printer usage.
- Standards. The Palladium print system will conform with the emerging European Computer Manufacturers Association (ECMA) printing standard [ECMA88]. Palladium is a reference implementation of that standard. It is hoped that this will allow future printing systems to be more easily integrated into the Palladium system.

## Distributed Systems Concepts

The goals of the Palladium Print System have been satisfied by exploiting the distributed systems concepts of:

- Client-Server Model
- Name Service-based Server Advertisement
- RPC-based Server Access Protocols
- Embedded Tasking Systems

The application of these concepts to our model can be seen in Figure 1.

### Client-Server Model

The client-server model proves very useful for the relationship between local user applications and shared resources, which are most likely remote. This approach allows for many clients to obtain central services and facilitate management.

### Name Service-based Server Advertisement

The name service allows the client to easily locate the address and information associated with the desired print service. Name service basing minimizes problems with information update and maintains the notion of a public, stateless workstation.

### RPC-based Server Access Protocols

The Remote Procedure Call (RPC) paradigm can be used to provide a simple interface to distributed services. Using the RPC service, distributed applications developers need only provide an interface specification to one of the RPC utilities (an RPC Stub Generator) to generate the access protocol code. The RPC paradigm can be exploited to shorten development cycles, thus lowering the development and maintenance costs of this application.

### Embedded Tasking Systems

RPC paradigms enforce a synchronous relationship between client and server. To equitably handle all requests that may be submitted to the server in a short period of time, and to supply reasonable response to each request, the server must have some way of logically doing multiple tasks at the same time. A tasking system will be used to provide this capability by either allocating time slices of a single process to different tasks or by creating new processes for each task that must be accomplished. The current implementation plan calls for the former "time-slice" approach.

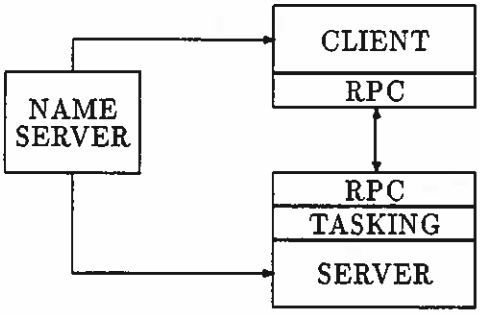


Figure 1: Client-Server Print System Model

## Palladium Print System Model

The Palladium Print System Model exploits specific instances of the Client-Server Model, Name Server Advertisements, RPC paradigms and Tasking to improve the managability of distributed print systems and provide equitable access to shared printer resources in the Athena distributed environment. The design of the Print System is illustrated in Figure 2.

The Palladium Print System Model has the following specific characteristics:

- The print client represents the user's printing needs to the name server to find an appropriate print server.
- The server represents the capabilities of the shared resources managed by the server to the name server.
- The name server tells the client where to find the print server that matches the user's needs.
- The server interface allows access to the services provided by the server.
- The client uses the RPC service to make a service request of the print server.
- The print server tasking system allows multiple clients to make requests at logically the same time.
- Once a request is accepted, the tasking system allows the job scheduler to process the request asynchronous from the submission of the request.
- The server database acts as a mediator between the request submission routines and the processing routines managed by the job scheduler.
- The scheduler passes responsibility for a job off to a specialized process called a 'printer supervisor'.
- The printer supervisor takes responsibility for printing a single job, using a single print format on a single (type of) printer. Note that a printer may support more than one print format, but for any specific job we presume that only one of those formats is exploited.

The separation of functionality exploited in this design allows for greater flexibility in extensibility and management of the system.

### Palladium Client

The Palladium Client selects the server which satisfies the requirements of the user-specified print parameters, file-specific print parameters and any system or user-profile parameters. The client then submits the job to this server. The client may also provide some local job management for the submitted job. Local job management keeps track of jobs for the local user and performs some post-processing such as notification and file deletion. Local job management may also support mapping between global job ID's and the name of the server to which the job was submitted.

### RPC and Tasking System

For Palladium, the HRPC software from The University of Washington [HRPC88] was chosen. HRPC is a common interface to a number of RPC systems. Although HRPC is a common subset of the supported RPC systems, it provides the functionality necessary for Palladium and for the ECMA services. HRPC provides a generic interface for remote procedure calls, and its use will help to keep Palladium independent of underlying protocols and RPC definitions.

We currently plan to use the Nectar threads package [Nectar88] and the Apollo RPC package [Apollo87] at Athena.

### Name Server

The Name Server advertises the capabilities of the shared resources and services provided by different print servers. The name server allows the client to identify the server that satisfies the user's needs and to find that server in the distributed environment. At Athena, the Hesiod Name Server [Hesiod88] is used. Hesiod might take a particular name ("MyPrinter" from the "printer" table for example) and return information needed to submit the request.

### Palladium Server

The Palladium Server handles requests submitted by the Palladium Client and processes those requests. Most requests are processed as they are submitted, returning the results immediately. Print and ModifyPrintJob requests are validated and, if accepted, added to the Job Info Database. The Job Scheduler locates the right printer which satisfies the job requirements, schedules the job on the printer and tracks the progress of the job. The Job Scheduler records job events in the Job Info Database and reports status and significant events to the client. In addition, the job scheduler maintains all accounting and error log information for the job.

The Server does not understand the syntax or the semantics of the file content. The Job Scheduler merely matches the job requirements with the capabilities of the printers managed by this server. In this way the server handles a broad range of file types and printer types.

The server may be implemented as several processes or as several tasks within a single process. However, it must run off of a single job, resource and printer database and use a single centralized scheduling or queuing mechanism for all the jobs and printers that it manages. Thus equitable shared access is provided, since all clients must go through the same server to get to the printers under the control of that server.

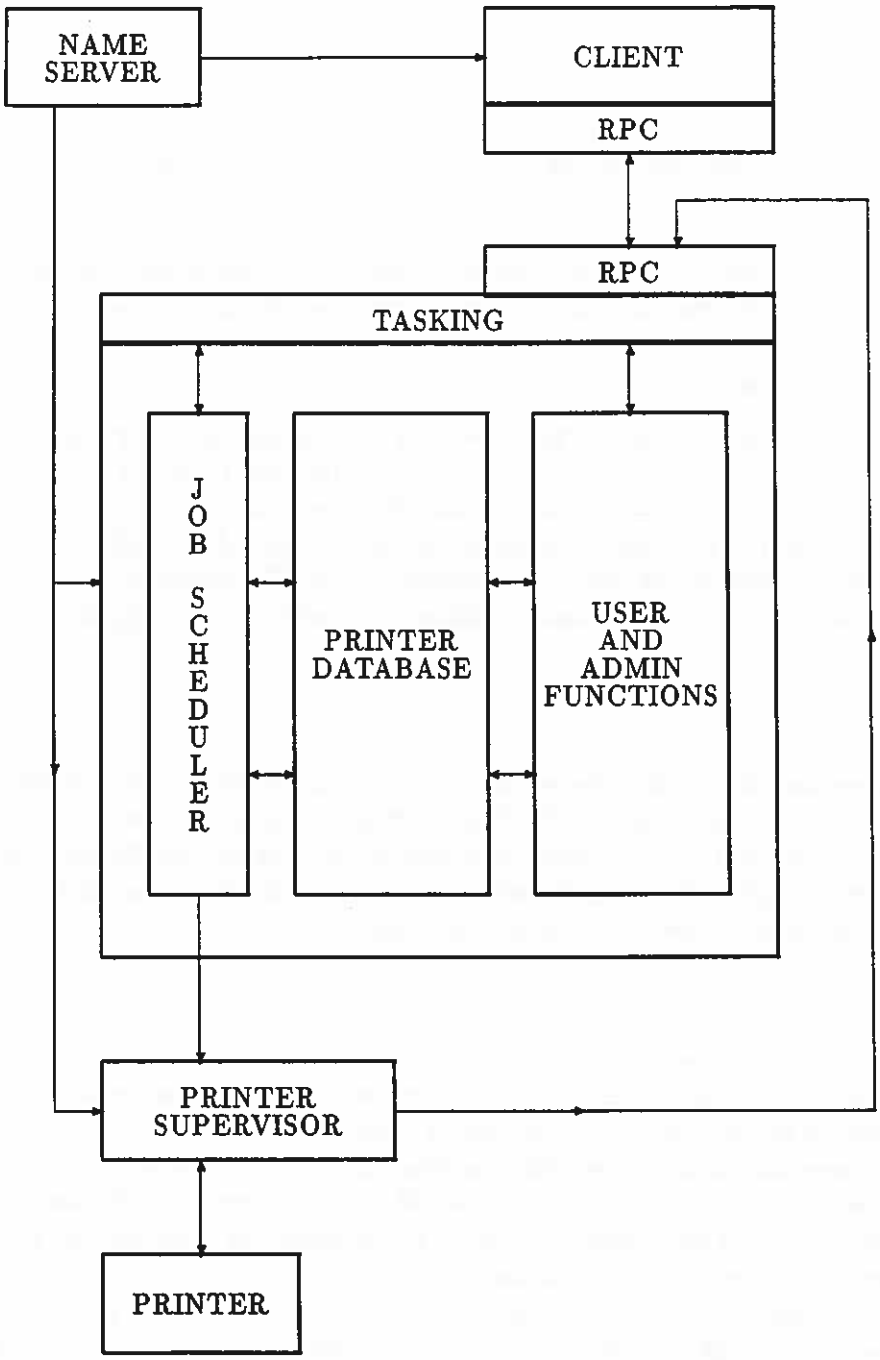


Figure 2: Palladium Print System Model

### Service Interface and RPC

The Palladium Server Interface is presented as a set of library routines (Pdlib). These routines are accessed through the RPC protocol described above. Palladium Clients make calls to the following library routines to obtain service from a Palladium Server:

1. Print
2. Modify Print Job (that was previously submitted)
3. Read Job Status
4. Cancel Job
5. List Jobs
6. Read Print Server Characteristics
7. Read Print Server Status
8. Read Printer Characteristics
9. Read Printer Status

In addition library routines are supplied which support the management of the Palladium Server. These routines take well-defined arguments and return results, rejections and events.

### File Transfer Paradigms

When a print job is submitted to a print server, the contents of the file(s) to be printed must be made available to the server at some point. There are three paradigms for print job file transfer:

1. Pass with Request (as a Print Operation parameter)
2. Retrieve by Server
3. Copy by Client

Palladium initially supports only the 3rd paradigm.

The first paradigm (Pass with Request) requires the RPC mechanism to be able to pass arbitrarily long file contents as a parameter of the print request. This paradigm also implies that such a request will return to the user after an arbitrarily long time. We could not satisfy this requirement and limit the time of the request with our current tools.

The second paradigm requires global naming and access to files in the distributed environment. In addition it requires authentication and/or authorization forwarding, to allow a print server to access a user's file. These requirements could not be easily met during the design of Palladium.

The third paradigm requires only a common file transfer mechanism, and the ability to notify the server when the client has completed the file transfer. The Athena/Palladium environment supports both of these requirements. Palladium has added an RPC library routine for passing chunks of data to the server to support this paradigm.

### Palladium Server Database

The Print Server Database contains high-level definitions of logical printer capabilities (e.g. names of media supported, finishing features supported, etc.), definitions of resources (e.g. sizes and colors of media, location of stapling, etc.), and print job request infromation. The print job information consists of the text to be printed and the parameters which will control the processing of the print request. The database also contains capability and location information for the supervisors which the server controls. This information is used in validating and scheduling a print request.

### Palladium Printer Supervisor

The Printer Supervisor provides a static process to drive a specific printer (or set of related printers) for the printing of one print job at a time. The Printer Supervisor is entirely responsible for interpreting the job request. This includes interpreting the content of the file(s) and reconciling the instructions provided in the print request parameters with the contents of the file. The supervisor has complete control over the printing of the job, including the loading of fonts, forms and other electronic resources into the printer, as well as the transfer of the file(s) to the printer. The Printer Supervisor is the equivalent of the printcap setup code of the Berkeley Print Daemon plus the 'child' code of the Daemon.

### Supervisor Interface

The Printer Supervisor provides the same RPC-based interface that the Print Server provides to clients. This allows the Printer Supervisor to be accessed by either a Print Server or by individual Print Clients. The Printer Supervisor may be configured to either: allow exclusive access by a single Print Server or allow access by 'all comers'. The latter configuration allows a print supervisor to be installed on a workstation to drive a personal printer attached to that workstation. The same interface used by the client to access shared printers through a Print Server can be used by the client to access a local (potentially non-shared) printer. The only difference to the client is the address advertised in the name server.

### Palladium Compatibility Gateways

The Athena Palladium implementation is completely independent of current System V and BSD Unix Print Spooler implementations. Interoperability with existing implementations can be provided through gateways between the Palladium client and traditional *lpd* daemons.

## Conclusions

The Palladium distributed print system has been designed to use existing software technology to construct a truly distributed printing system that can support current and future printing devices. We are currently implementing and testing the design.

The incorporation of centralized management into the system will allow for easier and more effectively management of print services as advancing technology and user requirements place greater demands on them. It is hoped that basing our development work on an emerging standard will

permit our work to be easily augmented and extended by Project Athena and other members of the MIT network community.

## Acknowledgments

We would like to acknowledge the efforts of the Palladium design team, Pradeep Chetal, Dave Gray, Jack Woychowski, and Gerhard Messmer, who contributed significantly to the design of the print system.

## References

[PostScript87] Adobe Systems, Inc., "PostScript Language Reference Manual," Addison-Wesley, Reading, Massachusetts, (April 1987)

[Athena85] E. Balkovich, S. R. Lerman, and R. P. Parmelee, "Computing in Higher Education: The Athena Experience," *Communications of the ACM*, 28(11) pp. 1214-1224, (November 1985)

[Campbell83] R. Campbell, "4.2BSD Line Printer Spooler Manual," Dept. of Electrical Engineering and Computer Science, Univ. of Calif., Berkeley, (July 1983)

[Nectar88] E. C. Cooper, "The Nectar Threads Package," Computer Science Department, Carnegie Mellon University, (May 1988)

[Hesiod88] S. P. Dyer, "The Hesiod Name Server," in *Usenix Conference Proceedings*, (Winter 1988)

[ECMA88] "Standard ECMA - Document Print Service Description and Print Access Protocol Specification," Preliminary Draft, European Computer Manufacturers Association, Group TC35-TG5, (September 1988).

[PdTeam88] Handspicker, et al. *The Athena Pd Design Document* MIT Project Athena, (September 1988)

[HRPC88] J. Sanislo, "An RPC/LWP System for Interconnecting Heterogeneous Systems," in *Usenix Conference Proceedings*, (Winter 1988)

[SMS88] M. A. Rosenstein, D. E. Geer, Jr., P. J. Levine, "The Athena Service Management System," in *Usenix Conference Proceedings*, (Winter 1988)

[Kerberos88] J. G. Steiner, C. Neuman, J. I. Schiller, "Kerberos: An Authentication Service for Open Network Systems," in *Usenix Conference Proceedings*, (Winter 1988)

[Zephyr88] T. DellaFerra, *et al.*, "The Zephyr Notification System," in *Usenix Conference Proceedings*, (Winter 1988)

[Apollo87] T. H. Dineen, P. J. Leach, N. W. Mishkin, J. N. Pato, and G. L. Wyatt, "The Network Computing Architecture and System: An Environment for Developing Distributed Applications," in *Usenix Conference Proceedings*, (June 1987)