APPLEBUS FILE SERVER

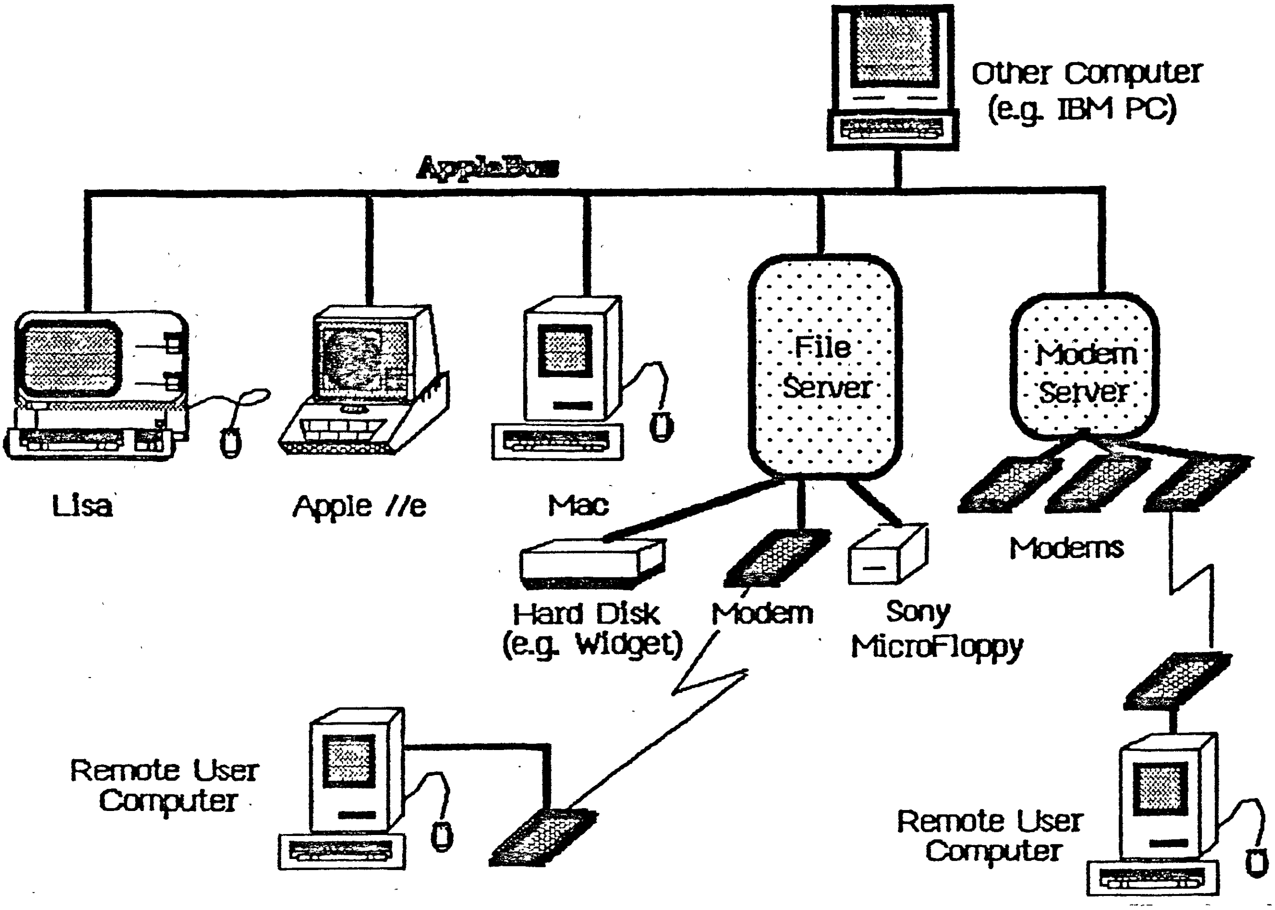
Functional/Global Description
- Preliminary Version -
Gursharan Sidhu -- November 11, 1983
(addendum to Bob Belleville's September 22, 1983 "Strawman Version")

I. SCOPE

This document discusses an AppleBus file server to provide users with a "disk sharing" facility for file storage and retrieval. User work stations will be Mac, Lisa, Apple-//, etc., computers although the server could be used by non-Apple computers (eg IBM PCs) if these are equipped with the appropriate software and AppleBus interface hardware.

In addition to providing expanded file storage space, the file server is a mechanism for user sharing and exchange of files/documents. As indicated, the server could be accessed from remote systems not directly connected to the AppleBus, via modem connections over telephone lines to a port on the server or on a different Communication/Modem server on the AppleBus. Access will also be possible, at a later date, through other networks interconnected to the AppleBus.

This is a preliminary document that spells out the global structure of the file server and the service it provides. Details on related server, work station, and technical design issues will be provided in other companion documents.



II. DISK SERVERS

Disk servers can be grouped into several classes:

(1) Information System Servers:

Servers that provide data-base/information system service (outside the scope of our consideration here).

(2) Servers that Expand the OS Disk Space:

The server's disk (or a part of it) is made available over the network to the work station, whose operating system can thus access an expanded disk space. Such servers fall into two groups depending on the type of access allowed:

. Block Servers (Disk/volume block read/write access)
. OS File Servers (Open/Close/Create/Delete/Lock/Unlock files,
and Put/Get/Read/Write file data)

These servers though versatile are not discussed any further in this document.

(3) Servers for User File Storage/Retrieval:

These let users store (and retrieve) complete files on the server's disk. The operations provided are Create/Delete/Store/Retrieve/Lock/Unlock/Rename a file.

Lisa and Mac users deal with the concept of an object (tools, folders, etc.) and not of a file. Objects can consist of several files and are put on the disk and retrieved from it for use. This extends easily to disk servers that let the user store/retrieve objects on their disk in a User File Archive.

III. FUNCTIONAL DESCRIPTION

The User File Archive provides the most "personal" network extension of the personal computer by placing the user in charge. It fits in naturally with the user interface/interaction metaphors of the Lisa/Mac systems and with other mouse-based user interfaces (eg Apple-// systems with the mouse). It is also the simplest kind of disk sharing service to build. The basic function supported by this system is storage and retrieval of documents/objects.

There are certain functions this server does not provide. It does not allow access to parts of files, eg. page access. It does not solve the problems of incompatibility (between the different types of work stations) in the internal structure/format of files. (Solutions can be built on "top" of the file service in one of many ways, but this is a non-goal of the server effort).

Yet, the potential for the server is great and we describe a fairly complete functional repertoire; for the restricted prototype and first release versions see the section on schedules and dependencies.

(A) The User File Archive and Its Metaphors:

The file archive is a collection of (one or more) file servers on the network, each appearing to the user as a "File Cabinet" with several "File drawers".

Documents/folders/tools are stored in File drawers and retrieved from them. Access to File drawers can be controlled as with a "lock and key" system using passwords. Documents/objects can be retrieved from a file drawer to the user's system and then be used as a local object. An added feature, not found in regular file cabinets is that some drawers have a "built-in copier": a document can be copied from a drawer while leaving the original document in it.

Each of these metaphors has an implementation equivalent:

File Archive	--	all file servers on the network
File Cabinet	--	a specific file server
File Drawer	--	a group of files on a file server
Object/Document	--	a file (or logically related grouping of files) in a file drawer.

(B) User Functions and Manager Functions:

File service functions are grouped into user functions and management and maintenance functions.

(i) User Functions:

User functions are:

1. Open the File Archive: list all file cabinets (file servers);
2. Open a File Cabinet: list the file drawers in a file cabinet;
3. Open a File Drawer: list all objects in the file drawer;
4. Store an Object from user system's desktop to a File Drawer;
5. Retrieve/Copy an Object from a File Drawer to user system's desktop;
6. Delete a file/object from a File drawer;
7. Rename a file/object in a File Drawer.

Another function, "Move", is a combination of retrieve and delete.

For the restricted prototype (and possibly the first version) file cabinets will have only one file drawer (flat file structure) and a file archive will consist of only one file cabinet (server). The additional structure of several file cabinets in the archive with several file drawers per cabinet will be added subsequently.

It should be clear how a user interface that fits in with the icon/metaphor approach of Lisa/Mac follows from this structure (details in a companion doc).

(ii) Maintenance/Management/Configuration Functions:

These fall into the following categories:

- Installing the Server: -- create delete file/drawers
  -- set/modify passwords/access modes
  -- alter space allocation of file drawers
- Backup Operations
- Diagnostic and Repair Operations

We want to keep the initial system simple so we propose building these functions in stand-alone software run with the server in offline mode. At this time a terminal (computer in terminal emulation mode) is connected to the non-AppleBus serial port of the server and the software is loaded from the server's floppy disk. Standalone operation eases enforcement of security, allows server diagnostic/repair/maintenance when the net is down, and eliminates the need for having AppleBus software resident at this time (the server will then of course be off the network).

If it is considered necessary, an on-line access attribute modification service can be added later. Other maintenance and management functions can be made to work in a on-line fashion as well (over the network) in later versions.

(C) Privacy/Access Control Issues:

This will not be implemented in the restricted prototype version of the server, but can be added later.

In the next version we will provide access control with the following features:

- Access control will be at the level of file drawers. Users will need a password in order to "open a file drawer";
- A drawer can be made private to a user, if the password (key) for that drawer is known only to that user;
- A drawer can be shared by several users by providing them with its password (key);
- A drawer can be made public (no lock on the drawer) by not associating a password with it.

In this version any user with the password to a drawer will have unrestricted use of the contents of the drawer, ie. will be able to copy, delete, rename, store any file in the drawer.

A more extensive access type control will be added in later versions. Then:

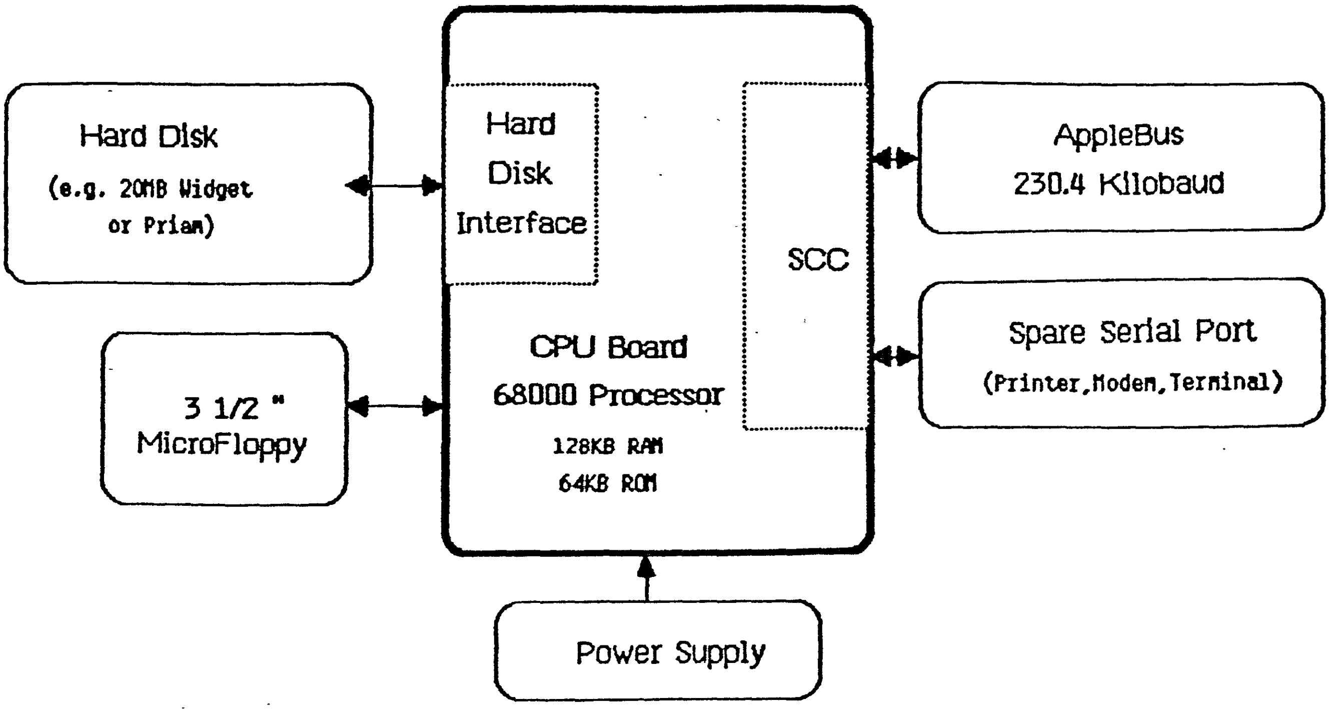
- Drawers can be made read-only, unrestricted read-write, controlled read-write;
- For controlled read-write drawers, when a user "pulls" a file from the drawer, it becomes inaccessible to any other user of the drawer until it is returned to the drawer. This can be achieved by removing the file from the server disk, or by locking it when a copy is sent out to a user. The file is unlocked when that user returns it (or a modified version) to the drawer.

IV. HARDWARE, COSTS AND PERFORMANCE

The file server will be built with a single board CPU based on a 68000 CPU and with 128Kbytes of RAM and 64Kbytes of ROM with the logical configuration of the following figure. A DMA (asynchronous) hard disk interface would allow the performance of the server to increase tremendously. This of course would have to be balanced against the additional cost.

(Hardware cost estimates based on Bob Belleville's proposal are as follows)

Item	Cost	
Hard disk (Widget 20 Mb)	$300	250K per year (Wolfgang Dirks expects this to go downto $250 by end 1984)
Sony 3 1/2" drive	$ 80	
Apple][power supply	$ 36	
Main digital board	$172	same as current Mac
Case	$ 13.50	J. Manock & L. Zsidek
Shipping container, manuals, etc.	$ 20	
Direct Cost	$621.50	
+ 1% labor	$ 7.00	
+ 7% overhead	$ 43.50	
TOTAL COST	$671.00	



File Server Logical Configuration

V. SOFTWARE MODULES

Details of the internal modular design of the software are still being worked out. The following software systems are needed to have a complete and useful file service system available on AppleBus. These are:

- File Server Control Program (including AppleBus and disk drivers):
  Will be loaded from the 3 1/2" floppy and executes in RAM. Additional software (serial port driver plus support protocol software) will be needed for remote access via the server's modem port;

- Work Station Access Software:
  An application program (tool) for file storage/retrieval plus associated AppleBus and protocol drivers. This will have to be done for each of the following work stations (Mac first):
  -- Mac
  -- Lisa
  -- Apple-// and/or Lolly

- Server Configuration/Maintenance software:
  Standalone software run on the server to configure/install it, to create access control information, etc.

- Backup system for the server (details TBD):
  Incremental backup to the 3 1/2" floppy

- Diagnostic modules for work stations and server (details TBD)

On the server we propose to use a standard existing operating system: the Mac OS and support structure, with a driver for AppleBus (final version to be provided by Larry Kenyon), and a driver for the server's hard disk. For the work stations the software will be in the form of a tool (utility program) built on the corresponding OS. For Mac work stations, this involves close interaction with the Finder (on the Lisa with the Filer/Desk-Top-Manager).

VI. PROTOCOLS

Currently the AppleBus data link and transport protocols are not defined. By using a layered approach we can go ahead with the implementation of the file server using File Server specific protocols on top of very simple non-standard transport and data-link protocols. This will allow us to implement and test the file server specific aspects without waiting for a final resolution of the broader protocol issues. An assumption on protocol structure is that (see figure below) that the file server protocols will work independently of the underlying transmission medium/system (for example AppleBus and/or dial-up asynchronous lines). To start with we will use the AppleBus data link protocol proposed by Ron Hochsprung and will move to the final data link protocol when Larry Kenyon publishes it.

VII. IMPLEMENTATION STAGES, SCHEDULE AND DEPENDENCIES

The File Service will be implemented in three stages:

(1) Restricted Prototype:

Objective is to test out the file transfer part of the implementation with limited attention to user interface aspects.

Server to be built on a standard Mac (or Lisa) using the external (additional) Sony 3-1/2 inch floppy as the server disk. Service features:

- Flat file structure (one file drawer per server);
- No access control (no passwords or access mode);
- Server will support up to 4 simultaneous users;
- Simple text based user interface;
- User functions:
  - Open the file drawer/file cabinet (ie. list files in the drawer)
  - File transfer (storage/retrieval) to from Mac systems (files copied from the server are deleted from disk)
  - File deletion
  - File renaming
- Minimal maintenance and diagnostic functions needed for development purposes only;
- Use current AppleBus datalink protocols (defined in Ron Hochsprung's proposal);
- No remote access service over asynchronous dialup lines.

The implementation schedule (approximate) is as follows:

Now	--	receive one more Mac system from Mac division (have one now, need the other one RIGHT AWAY!);
Now	--	need AppleBus cables, connectors, terminators, for the Mac systems right away;
by end Nov 83	--	have an AppleBus with a fixed bus master and two Macs and two Lisas exchanging packets reliably and have a functioning AppleBus monitoring tool on a Lisa and a Mac system;
Dec 1, 83	--	need one external Mac microfloppy drive;
Dec 83 - Jan 84	--	complete software for prototype file server and Mac access software;

(2) First Release Version:

The objective is to build a simple but useful product in our first release, saving advanced or esoteric features for later versions. The major attention here will be to:

- finalize the product's specification based on the prototype experience;
- bring the software up on the final server hardware and with a hard disk on the server;
- enhance the user interface to be icon-based;
- build the maintenance, management and diagnostic tools

including the backup utility;

- time permitting, add the multi-drawer and access control features (but this is optional);
- enhance server capability to support up to six simultaneous users;
- modify drivers and protocol software to the finally accepted protocols;
- to subject the software to extensive testing to establish robustness;
- generate the user and internal documentation for the product;
- build in the remote access via dialup connection feature into the server.

An approximate implementation schedule for this is as follows:

Early Jan 84	--	receive prototype server hardware;
end Jan 84	--	complete product specification including maintenance, management and diagnostic tools;
Feb 84	--	write and test hard disk driver and server file system modifications plus the AppleBus drivers for the final protocols;
Feb 84	--	write preliminary versions of documentation;
Mar-May 84	--	build and test v. 1 of server software
Mar-May 84	--	build Mac workstation user software
	--	build Lisa workstation user software (needs enhanced Filer interface);
April 84	--	review and modify documentation.
April-May 84	--	build and test Configuration and Backup software for the server
June 84	--	software release for final testing

This is an aggressive schedule involving the parallel implementation of several pieces of software. Delays in delivering the server hardware and/or delays in coming to a protocol structure decision could hold things up.

(3) Later Versions:

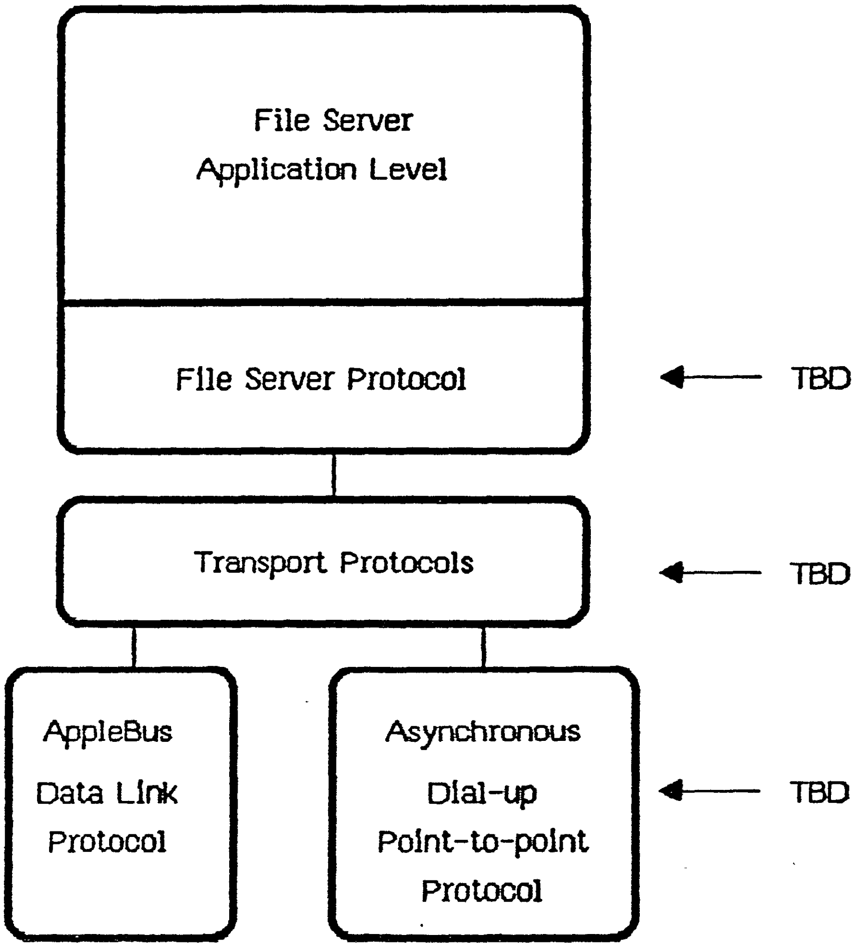
These would be undertaken in response to feedback from users regarding potential inadequacies and/or the need for new features. If multi-drawer, multi-server, access control is not included in the first release product then these enhancements could be provided at this time.

VIII. OPEN ISSUES

There are various issues that are still open and closure needs to be brought about on these. We note the following at this time:

- Protocol issues as noted above
- The use of the server ROM
- Which hard disk should the server have and with what interface? Disk size affects the requirements on the backup device
- Will the server CPU board be a modified Mac or the Lisa technology based GLM?
- Enhanced interface to the Lisa desktop manager/filer

We have charted a development path to proceed without letting these issues slow us down.



Protocols Structure