-- --

IEEE P1003.0 Draft 13 – September 1991

Options and Variations

The ISO/IEC 9945-1 standard draws heavily upon major implementations of the UNIX Operating System, including System V and the Berkeley versions. Where a specific behavior was clearly needed (e.g., signals), only a single behavior was permitted. However, there are points where functions were considered optional and others where two different behaviors were considered acceptable. However, in many cases, a solid technical argument favoring one approach over the other was not established. In this case, two behaviors (usually System V and BSD) are defined as being permitted. This is of benefit in writing portable applications, since those that can tolerate both behaviors will run on a wider range of systems. It is also a slight disadvantage in writing such applications, since it can mean handling a wider range of implementations.

NOTE: In a related U.S. national standard, the United States Government Department of Commerce National Institute of Standards and Technology has adopted the ISO/IEC 9945-1 standard as a Federal Information Processing Standard (FIPS 151-1) for use in computer systems procurement. FIPS 151-1 requires options and characteristics that have not been required by ISO/IEC 9945-1, requiring implementations to function in a specific way in many of these cases. While buyers may want to make such requirements, the impact should be understood. Applications written for FIPS systems may only run on FIPS systems, whereas applications written for full ISO/IEC 9945-1 implementations will run on both FIPS systems and POSIX conforming systems that are not FIPS conforming. FIPS systems will run applications written either way.

FIPS 151-1 requires all three POSIX.1 options. In addition, it eliminates several additional possible behaviors, thus narrowing the variation between conforming implementations.

4.2.5.1.2 Regional Standards

None.

4.2.5.1.3 National Standards

None.

4.2.5.2 Emerging Standards in the POSIX OSE

4.2.5.2.1 International Standards

None.

4.2.5.2.2 Regional Standards

None.

4.2.5.2.3 National Standards

The IEEE P1003.4 Group is defining realtime extensions to ISO/IEC 9945-1. Draft 9 of the realtime POSIX extensions proposes standardized interfaces to the following functions:

- — Response to asynchronous events
- — Priority interrupts and scheduling
- — Preemptive scheduling
- — Memory locking
- — High-performance file system (contiguous or other)
- — Realtime timers (with nanosecond resolution times)
- — Shared memory
- — Semaphores
- — Interprocess communications (message passing)
- — Asynchronous event notification
- — Synchronous input and output.

The P1003.4 group is also specifying an interface to threads (P1003.4a).

4.2.5.3 Gaps in Available Standards

While ISO/IEC 9945-1 and P1003.4 both represent very important work, they do not yet address all of the services indicated in 4.2.4. Areas of particular shortfall include Event, Error, and Exception Management Services, some Generalized I/O Services (particularly concerning services for device drivers), and System Initialization, Reinitialization, and Shutdown Services. In addition, Security (see 5.2) and Reliability, Adaptability, and Maintainability (Fault Tolerance; see 5.4) services are not reflected in these two base standards, and some capabilities are explicitly considered to be implementation defined. For some of the services discussed here, adequate consideration is not given to the implications of multiprocessor and distributed implementations of the services and interface provided. Finally, since these are intended to be base standards (or, in the case of P1003.4, an extension to a base standard), profiles are needed in order to select appropriate features and provide appropriate combinations with other related capabilities.

4.2.5.3.1 Public Specifications

The following are public specifications that define interfaces to services for which no formal standards are currently available.

OSF/1

The Open Software Foundation (OSF) ‘‘Application Environment Specification (AES)—Operating System Component’’ (OSC).

Service Gaps Addressed:

- — Generalized I/O
- — Event, Error, and Exception

— System Init/Reinit/Shutdown D

SVID D

The AT&T System V Interface Definition (SVID), Issue 3. D

Service Gaps Addressed: D

— Generalized I/O D

— Event, Error, and Exception D

— System Init/Reinit/Shutdown D

4.2.5.3.2 Unsatisfied Service Requirements

There are two significant areas of the services described above for which no standards currently exist. One is the considerations implied by the use of multiprocessors to implement some or all of the services described herein. The other area is that of interfaces to logical device drivers.

4.2.5.3.3 Standards/Specifications Outside the POSIX OSE

4.2.6 OSE Cross-Category Services

4.2.6.1 Capability and Security Services

These services support the ability of the system to control usage such that system integrity is protected from inadvertent or malicious misuse. These protection services provide a mechanism for the enforcement of the policies governing resource usage. Note that many of the security services are implicit services; i.e., they are provided without an explicit request to the operating system. There are two distinct classes of system access with which operating system services must be concerned: physical access and logical access.

Security services at the physical level are used to protect against security compromise, given unauthorized personnel may have physical access to system hardware. Typically, the physical access is to a terminal and/or terminal/display cables; however, physical access may also include network cables, central processing units, disk drives, or tape drives. Prevention of physical access by unauthorized personnel may require different operating system services under different circumstances.

Logical access is the ability to interact with the operating system via a terminal/display. Security services at the logical level can be implemented through passwords and watchdog timers.

Capability services attach operation lists that limit a process's ability to act on resource objects. This is to ensure the resources are not misused. Access to resources can be protected by services using capability lists as well as access lists, lock/key mechanisms, global tables, or through dynamic protection structure services.

Prevention of Unauthorized Access

The system may need to be guarded from attempted access by unauthorized personnel. The point of access to the operating system that is typically of concern is through the API. Given the mode of operation (system high, multilevel, open) at which the system is operating, these services differ and have differing implications on other system services (such as reliability and naming) and system performance.

Prevention of Data Compromise

These services prevent access of data by users not authorized to the data. These services may be implemented using access lists on files (and directories) and/or encryption of data or in other ways.

Prevention of Service Denial

These services ensure that a service request will be met by the operating system in a reasonable time if the requester is authorized to use the service. These services ensure that a bandit user or process cannot cause system malfunction by monopolizing system services or resources.

Security Administration

This category involves services to allow the management of the security system, including the administration of permissions to personnel, data, and services as well as capability lists. In addition, it permits the administration access mechanisms (most often passwords and capability lists) and services that allow the system to switch modes of operation. The services will likely be accessed by the target system operator with security responsibilities through the target system operator services.

4.2.6.2 Fault Management

Many of the fault management capabilities described in 5.4 are normally the responsibility of the operating system and fall under the sorts of services described in this clause.

4.2.7 Related Standards

The following emerging standards are related to the services covered in this D
clause, in as much as they address at some level services either explicitly listed in or implied by the services found in 4.2.4:

- P1003.6 Security Interface for POSIX.
- P1003.12 Protocol Independent Interfaces (for networks).
- P1238 OSI Application Program Interfaces (initial effort is to provide at least sufficient facilities for the support of FTAM API

specifications).

4.2.8 Open Issues

[PO: PCTE and CAIS-A have been moved here largely because it is not clear what to do with them. They are not adequately accommodated by this model. They are both hybrids of operating system and database management system capabilities that seem to belong either everywhere or nowhere. They could both well be used in conjunction with a P1003.1 implementation, but they could also be implemented on other base operating systems, or implementations could even expand their capabilities to provide full operating systems. P1003.0 must decide what to do with them.]

PCTE

An effort by the European Computer Manufacturers Association (ECMA) has resulted in the definition by Technical Committee 33 of the Basis for the Portable Common Tools Environment (PCTE). This is now an ECMA standard and is referred to as Standard ECMA-149.

CAIS-A

MIL-STD-1838A (CAIS-A) was developed by the US Department of Defense to provide a common foundation for Ada Programming Support Environments. Similar in nature to PCTE (see above), it too covers many of the system services covered by 4.2.4. In addition, it provides data management services such as those discussed in 4.4 and data interchange services (specifically, a Common External Form) similar to those discussed in 4.5.

D

X/Open

[PO: The following text on XPG was given to me at the last meeting. However, I cannot tell from the text whether XPG is more like this guide or more like P1003.1. If it is the former, it does not belong here. If it is the latter, then this text should be moved into 4.2.5.3.1.2.]

X/Open is a consortium formed to create interface definitions and guidelines that will provide portability and interoperability to its members' products. Its primary output is the X/Open Portability Guide. The functional content of this guide is heavily based on the AT&T SVID. The current version is XPG3.

[PO: Documentation of open issue: what are we going to do with FIPS? For this section, the decision was made in Chicago to stick FIPS 151-1 in the section on 9945-1, but this proved quite awkward: it couldn't be placed in the regular text because it didn't pertain directly to 9945-1 (i.e., it was inappropriate to discuss it in the text that was to be devoted to ''international standards''), so it ended up in an informative note. I don't find this a satisfactory answer. I think that it properly belongs under Government/Legal Standards; the ''gap'' it fulfills is one of

profiling.

Documentation of new open issue: We are beginning to ''use our heads,'' as indicated by the discussions we had on this section in Chicago. It is essential that we capture the rationale for our decisions, particularly for our inclusions and exclusions. I have tried to do that here to some extent, but what is our overall plan for making sure that our rationales are not lost?]

4.3 Network Services

Responsibility: Charles Severance D

HLJ : This is a complete clause replacement in Draft 13. It is not further diff-marked. D D

4.3.1 Overview and Rationale

This clause describes the network services component of the application platform. It also describes the services provided to application programs and users, and it describes current and emerging standards that are standardizing these services.

Applications gain direct access to network services via the POSIX API. The network is just another system resource (albeit an important one) allocated among the competing processes.

4.3.2 Scope

Network services include those capabilities required to allow application programs to execute on platforms that may be connected via a network. They cover the areas of file transfer, namespace and directory services, services in support of distributed environments such as remote procedure call, distributed time management, transparent file access, and data representation services. The application programs using these services should be able to access them via a high-level, context-insensitive or low-level, context-dependent interface.

In the open systems and distributed systems environments, interoperability is of equal or greater importance than portability. The interfaces provided by the network services must be network protocol independent and provide for this level of interoperability. The network protocols defined for both Open Systems Interconnect (OSI) and Internet Protocol Suite (IPS) for TCP/IP should provide the basis for the open networking interfaces; however, these interfaces should not preclude the use of some subsequent networking protocol in the future.

Rationale: It is important for an open system to interoperate with more systems than just other open systems. Many open systems users will have requirements to interoperate with non-ISO networks for the near future.

4.3.3 POSIX OSE Network Services Reference Model

This subclause identifies the entities and interfaces specific to the construction of an POSIX Network Environment. This environment is consistent with and extends the environment of Section 3.

As illustrated in Figure 4-4, the components of a network architecture that require standardization are divided into two groups called external environment interfaces (EEI) and application program interfaces (API).

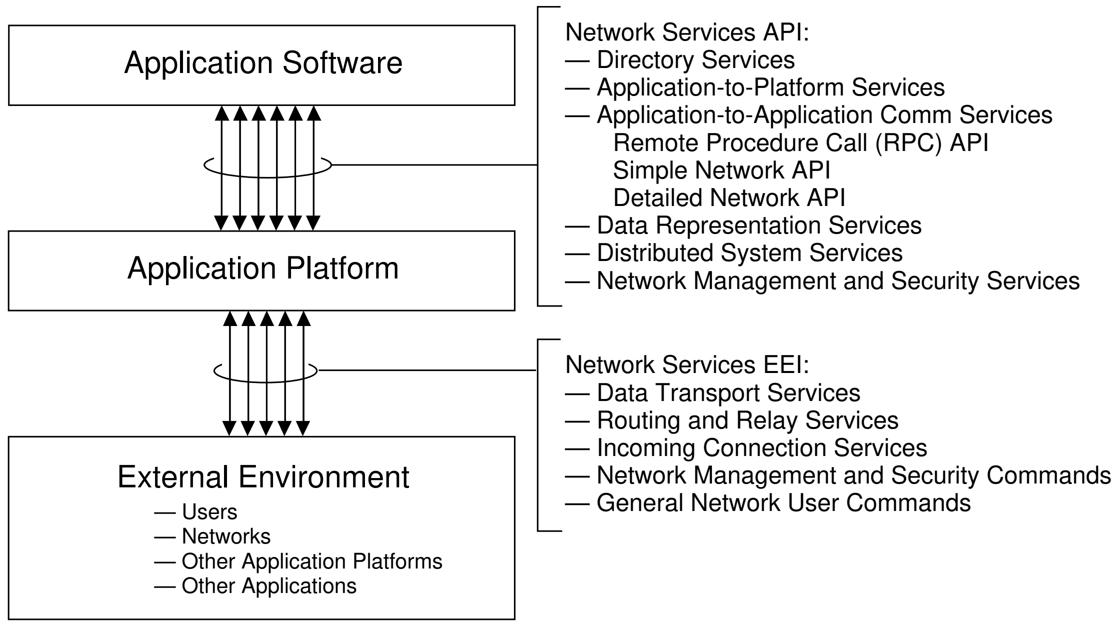

Figure 4-4 – POSIX Networking Reference Model

There may be some correspondence between services offered to the application across the API and the interfaces available at the EEI. It is quite possible for an API service to have no corresponding effect at the EEI. A good example of this is an interapplication communication service provided by the Network API between two applications on the same application platform. There may also be services available at the EEI provided by the Application Platform that are not available at the API such as remote login services.

4.3.3.1 Network Application Program Interface (API) Services

The API is concerned with the interfaces and associated standards that apply to the interface between the application and the application platform.

The services available at the API are:

- — Directory Services
- — Application to Platform Services
- — Application to Application Communication Services
- — Data Representation Services Services
- — Distributed System Services
- — Network Management and Security Services

Directory Services are those services associated with identifying and naming network elements.

Application to Platform Services provide an application with a very high level interface to networking capabilities. This interface provides applications with capabilities such as ‘‘mail this file to this address’’ or ‘‘transfer user xxx file from host yyy to the local host.’’ These services do not require the application to be aware of any of the low level network details.

Application to Application Services are the services provided by the Application Platform that allow an application to communicate with another application to exchange information. These interfaces support applications that range from having extremely simple networking requirements to the most complicated applications that must make full use of every possible network capability.

Data Representation Services provide the application with network oriented data representation services to insure the application can interchange information with other entities in the proper format.

Distributed system services provide the application with the ability to make use of multiple physical computer systems resources.

Network management and security services allow the application to control and configure the network resources.

4.3.3.2 External Environment Interface Elements

4.3.3.2.1 User Interface EEI Elements

The User interface EEI elements include the commands that users can use to perform network functions such as:

— File transfer

— Electronic Mail

— Remote printing

These commands are considered to be beyond the scope of this clause and will be covered in 4.9.

The User interface EEI elements that will be covered in this section are the commands that are used to perform network management and security functions.

4.3.3.2.2 Communication EEI Elements

The primary focus of the network EEI is the network protocols and supporting formats for network communication.

The entities in the external environment may be other application platforms or user interface equipment connected to the network using the open networking protocols. The standards at the EEI will be in several areas including:

— Physical connections

— Network protocols and formats

— Distributed systems services

The standards at the EEI will impact system interoperability but also may have an effect on application portability because certain applications may require particular types of network access to operate.

4.3.3.3 Implementation Aspects

The POSIX OSE Network reference model focuses on the requirements of application portability and system interoperability. As such, the model does not represent how systems are actually put together.

In the network area, there is much effort dedicated to the design of network standards to allow network components to be re-usable. This subclause shows how some of these network standards are related within the POSIX Network Reference Model.

Other network models are also related to the POSIX OSE Network Reference models. None of these other models are in conflict with the POSIX OSE Network Reference model. These models show much more detail in the area of how different standards work together.

4.3.3.3.1 Relationship Between the OSI Reference Model and the POSIX OSE Network Reference Model

Figure 4-5 shows the OSI seven layer model for networking as standardized by ISO [ref].

There are many aspects of network architecture that are specified by the OSI Network Model model:

— The number of layers in the model and the roles for each layer.

— An indication of which layers are logically end to end and which layers are simply to the next physical network node.

— The protocols between the layers and the protocols between the peers within the same layer. This has an impact on the actual format of the information transferred between nodes at the physical layer.

In addition, this model specifies how networks of computer systems can be assembled using the routing capabilities of intermediate nodes.

The POSIX OSE Network Reference Model has a much more limited scope than the OSI network model. The POSIX OSE reference model only looks at two interfaces to an application platform: the interface between application software and the application platform (API) and the interface between the application Platform and the External Environment (EEI). At both the API and EEI, the POSIX OSE network model describes the services that are provided to the application or external environment at the interface.

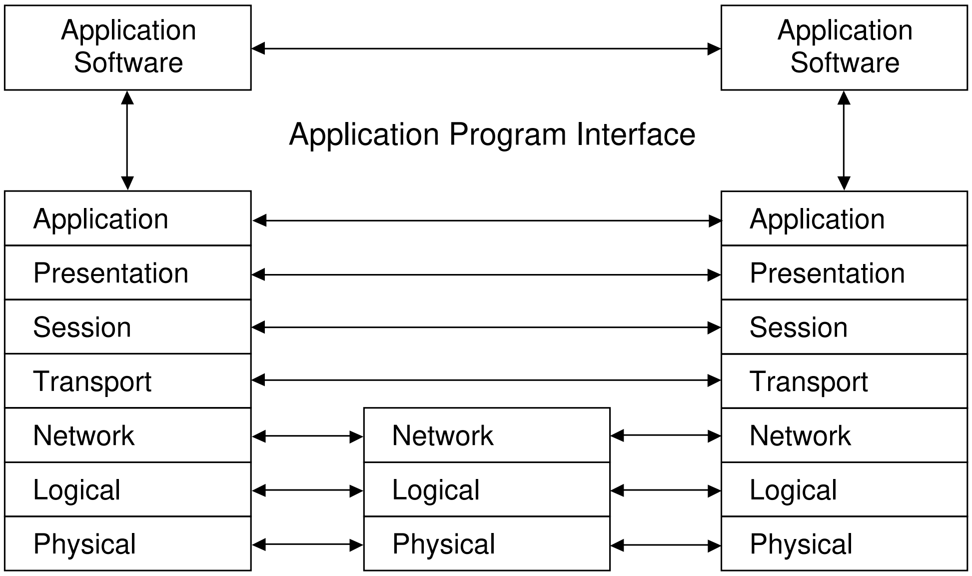

Figure 4-5 – ISO Networking Reference Model

Figure 4-6 shows how the two models could be related for one of the POSIX OSE Network Reference Model Application Programming Interfaces and three of the External Environment Interfaces. The POSIX OSE reference model focuses on a single application platform rather than a whole network. The POSIX Network API includes the API provided to interface to the application layer of the OSI model.

Because the OSI portions of the Application Platform External Environment Interface depend on the format, protocol, and services of what is produced at the physical level of the OSI reference model, the EEI technically depends on all seven layers the OSI model plus the services added on top of the application layer such as platform provided services or network management services.

This guide will not call out all of the standards that specify the protocols, formats, and options for all seven layers of the ISO network standards. Instead, this guide will focus on the services provided at the EEI using all of those formats and protocols.

Figure 4-7 shows an API interface to only layer seven of the OSI Network interface, which is intended to be the primary API for accessing network services. It is possible to define APIs that interact directly with any of the seven layers. There are a number of pragmatic reasons to provide APIs that access layers below layer 7. The cost of using one of these lower layer APIs is that the applications may sacrifice portability and/or interoperability.

It is important to note that while these APIs are represented as a part of a layered network architecture, from the point of view of the application interacting with the application platform, this layering is not critical to the use of the services. From the application perspective, there are simply three different types of

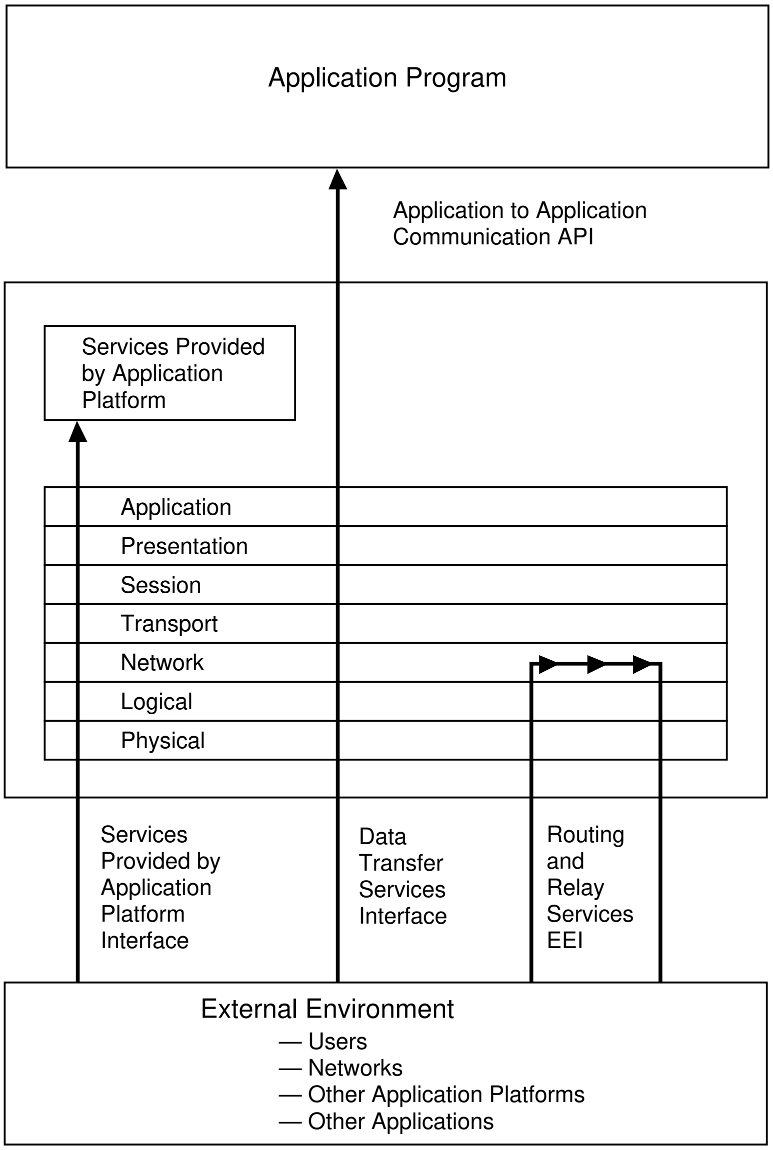

Figure 4-6 – Relationship of ISO and POSIX OSE Network Reference Models

network services, each with a different set of capabilities and requirements. Whether or not there is any actual layering or code common to the three services is implementation dependent.

4.3.3.3.2 POSIX Network Standards Efforts

The current POSIX approach to networking focuses on producing Application Program Interface (API) specifications. Most of the network connectivity specifications at the External Environment Interface are well covered on other

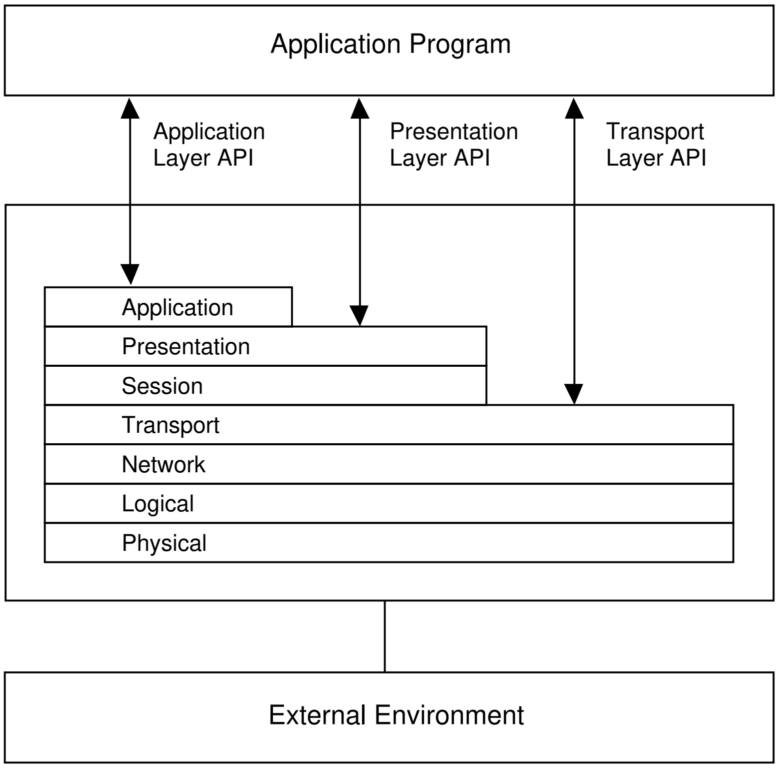

Figure 4-7 – Multiple POSIX OSE APIs to Different OSI Layers

standardization areas such as ISO (OSI networking) and the RFC process (TCP/IP).

One important aspect of the POSIX networking approach is that it is not focusing solely on producing standard APIs for OSI Network services. The POSIX Simple Network Interface (P1003.12 SNI) is explicitly designed so to be implemented transparently on a wide variety of networks. At the current time the possible list includes:

- OSI Application Layer
- OSI Transport Layer
- TCP/IP
- Other networks including proprietary networks

The current POSIX API standardization efforts include:

- P1003.12 — Simple Network API
- P1003.12 — Detailed Network API
- P1003.17 — Directory Services API

— P1238.0 — OSI Application Layer API (ASCE)

— P1238.1 — OSI Application Layer API (FTAM)

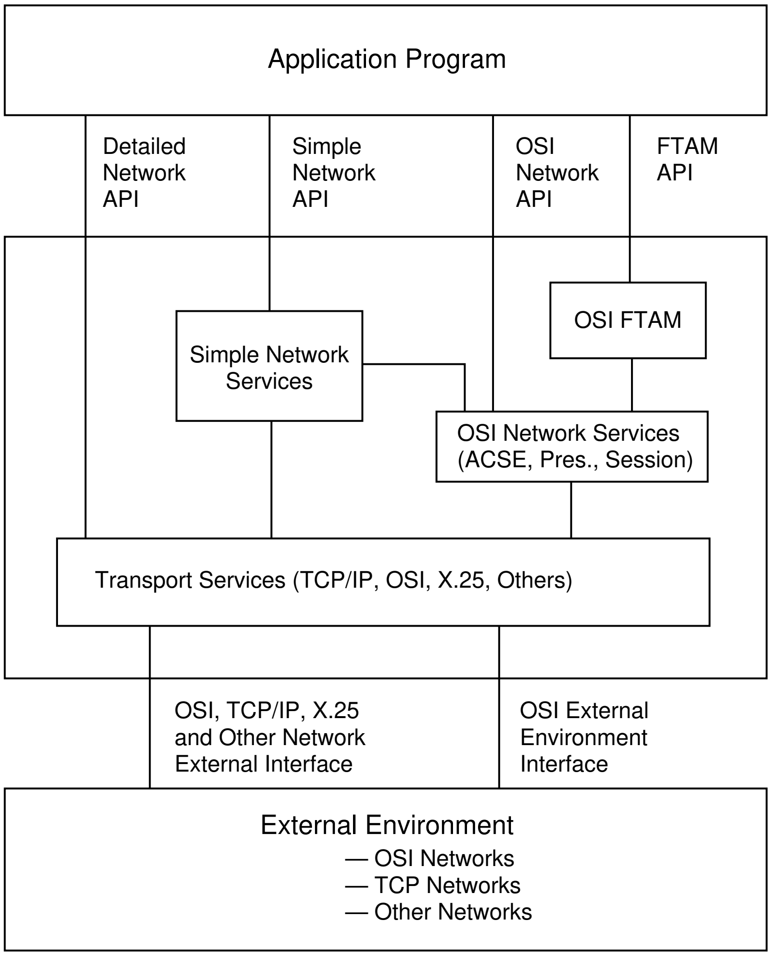

Figure 4-8 – Basic Network Services Model

Figure 4-8 shows how the basic network services can be related. The Simple Network Services API is designed so that a Simple Network Services Implementation can be done using the services available using the Detailed Network Interface API. An application can use the Detailed Network Interface to access multiple network transports but there may be differences between networks visible at the API. Applications that need to be portable across different types of network transports should be written using the Simple Networking Interface.

It is important to note that while the SNI API and DNI API standards have been designed so that the SNI Services can make use of the DNI API to access transport services, it is not a requirement that every implementation of SNI Services be written using the DNI API to access transport services. From the point of view of the application program, it is only important that the application platform provide an API for both the SNI and DNI services. This interface between the SNI Services and the Transport Services is an example of a Systems Integration Interface as described in 3.6.

Another example of Systems Integration Interface that is the subject of discussion in the POSIX Network area is the interface between the OSI Network Services and the transport services. This may or not be required to be the DNI API. This is an example of an interface that should have no impact on user application portability but may have great impact on the ability to procure the different types of network services from different vendors.

The area of Directory Services (P1003.17) is also specified so to be able to make use of different types of Directory Services including:

— X.500 Directory Services

— TCP/IP Directory Services

— POSIX P1003.7 System Administration and Management Services

Figure 4-9 shows how the Directory Services are related to the other network services. All of the APIs and SIIs from the previous figure have been eliminated to reduce the number of interfaces shown on the figure.

4.3.4 Service Requirements

The service requirements for the network component of an open system are very wide ranging. Many of the other components of the application platform make implicit or explicit use of network services.

Much standardization effort has gone into the aspects of networking that are available at the external environment interface. Effective networking standards at the external interface are fundamental to providing system interoperability.

The service requirements for both the API and EEI are described in this section.

CRS : Comment to mock ballot reviewers: This section is supposed to state the services required by an application at the API and the requirements that users and networks place on the external interface of the computer system. We have found that an effective way to state these requirements in a general way in to use the phrase ''ability to ....'' Comments are welcomed to add service requirements to these sections. We know there are general service requirements that could be added. Comments adding service requirements (especially those using our terminology) are greatly appreciated.

4.3.4.1 Application Program Interface Services

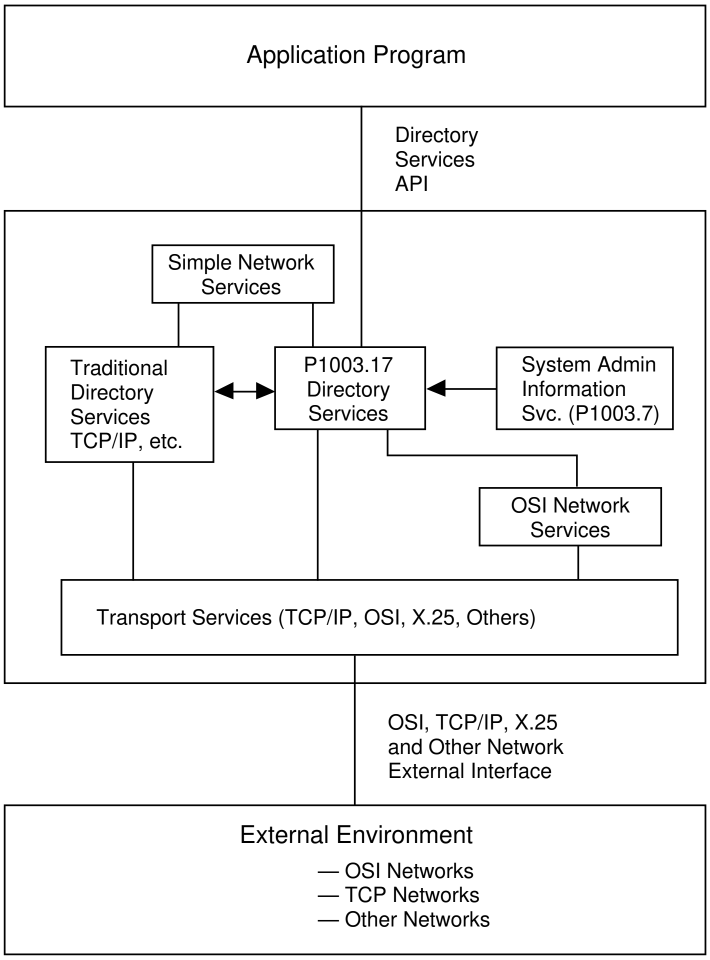

Figure 4-9 – Directory Services Architecture

4.3.4.1.1 Directory Services

CRS : Rob, any additional service descriptions would be appreciated. You know this area much better than I do.

Directory services allow an application to find the names and addresses of objects and services available to the application. These services include:

- — Ability to look up the name to be used to access a particular service
- — Ability to look up the address of a named object

4.3.4.1.2 Application to System Services

These are the services requested by the application that are performed by the Application Platform on behalf of the application without the application actually communicating directly with another application. Many of these services may actually connect to some remote application but the details of the connection are left up to the application platform.

These services will be provided by a relatively simple high level API. These services include:

(1) File transfer

(2) Remote execution of commands

(3) Mail delivery

(4) Remote login

(5) Remote printer access

(6) Network status

— Ability to access remote or local systems using remote procedure calls (RPC). When this type of access is provided, nearly all of the details of the network connection and interaction are masked from the application.

4.3.4.1.3 Application to Application Service

There are three areas of application to application service requirements:

— RPC Services

— Simple Network Services

— Detailed Network Services

The RPC services allow an application to register with the network application platform as the provider for a particular RPC Service. Once the service has been properly registered, other applications can transparently request services using a subroutine call. The details of communicating the service request to the application that is registered to provide the service and the return of the response to the requesting application are handled transparently by the Application Platform.

Applications making use of RPC services may not even be aware that the service are being provided via an RPC mechanism.

The Simple Network Services are application to application services provided using a simple set of interface routines. These will allow a wide variety of networking applications to be written that do not need to exercise control their network access at a very complex level of detail.

In addition, these services should be provided over a wide variety of network transport mechanisms. Applications written exclusively using the simple services should be portable across a wide variety of networking environments.

Applications written using the simple network services may not be able to make use of unique advantages of a particular physical networking scheme. To make use of these network-specific features the Detailed Network Services must be used.

The service requirements for the simple network services are intended to be the minimum requirements to write a large subset of network applications.

The Simple Network Services sacrifice the capability to control every detail of the network services in the interest of portability across networking environments and applications simplicity.

The Detailed Network Services API allows the application to control over much more detail of the network services. In addition, using the Detailed Network Services an application may be able to make use of unique networking capabilities available in particular networking environments.

4.3.4.1.3.1 RPC Services

These service requirements include:

- — Ability to register as an RPC service provider
- — Ability to wait for incoming requests
- — Ability for an application using RPC services to control parameters such as timeout

CRS : Someone who is an RPC whiz should beef this up with more detailed RPC services

4.3.4.1.3.2 Simple Network Services

The services provided at the simple network interface are:

(1) Name resolution

(2) Connection oriented services

- — Ability to indicate willingness to accept incoming connections
- — Establishing and destroying connections
- — Data transfer over connections
  - Read
  - Read with timeout
  - Write
  - Write with timeout
- — Simple error handling
  - Connection dropped notification

- Connection read failure
- Connection write failure

— Ability to close a connection

- Unconditionally
- Only after all data has been received

(3) Connectionless services

— Ability to indicate willingness to accept incoming requests

— Ability to send requests

- With acknowledgment
- Without acknowledgment
- Specified timeout

— Ability to receive requests

- Wait unconditionally
- Wait with timeout

— Ability to query as to whether any requests are available

— Simple event notification

- Lost request
- Request acknowledgment

— Simple error handling

- General network failure

(4) Support for server applications

— Ability to register as the provider for a service

(5) Simple status inquiry

— General network availability

4.3.4.1.3.3 Detailed Network Service Requirements

The services provided at the Detailed Networking Interface include all of the service requirements in the Simple Network Service Requirements plus the following:

(1) Ability to query the network services to get detailed information about network configuration and status

(2) Ability to specify performance metrics

(3) Ability to control routing

(4) Ability to select between different network protocols

(5) Ability to negotiate capabilities

- Required capabilities
- Optional capabilities
- Ability to determine the results of the negotiation

(6) Capability to deliver information with different priorities

(7) Ability to request and process extended event notification

(8) Ability to request and process extended error recovery including allowing the application to completely control error recovery.

(9) Ability to make full use of network resources for performance critical applications

This should provide the application with the ability to completely control connection oriented services and connectionless services.

4.3.4.1.4 Data Representation Services

- Ability to access all of the data representation and format conversion services to allow an application to communicate with a wide variety of computer systems.

4.3.4.1.5 Distributed System Services

The services provided in this area include:

- The ability to identify available resources in a distributed system
- The ability to dynamically make use of the resources in a distributed system.
- The ability to access files regardless of the physical location of the files.
- The ability to have reliable time services across all of the resources of the distributed system.

4.3.4.1.6 Network Management Services

The services provided at the Network Management API are:

(1) The ability to manage

- Network objects
- Network relationships
- Network security

(2) The ability to monitor and report on