Worldwide Developers Conference

Rhapsody Networking APIs and Services

Justin Walker

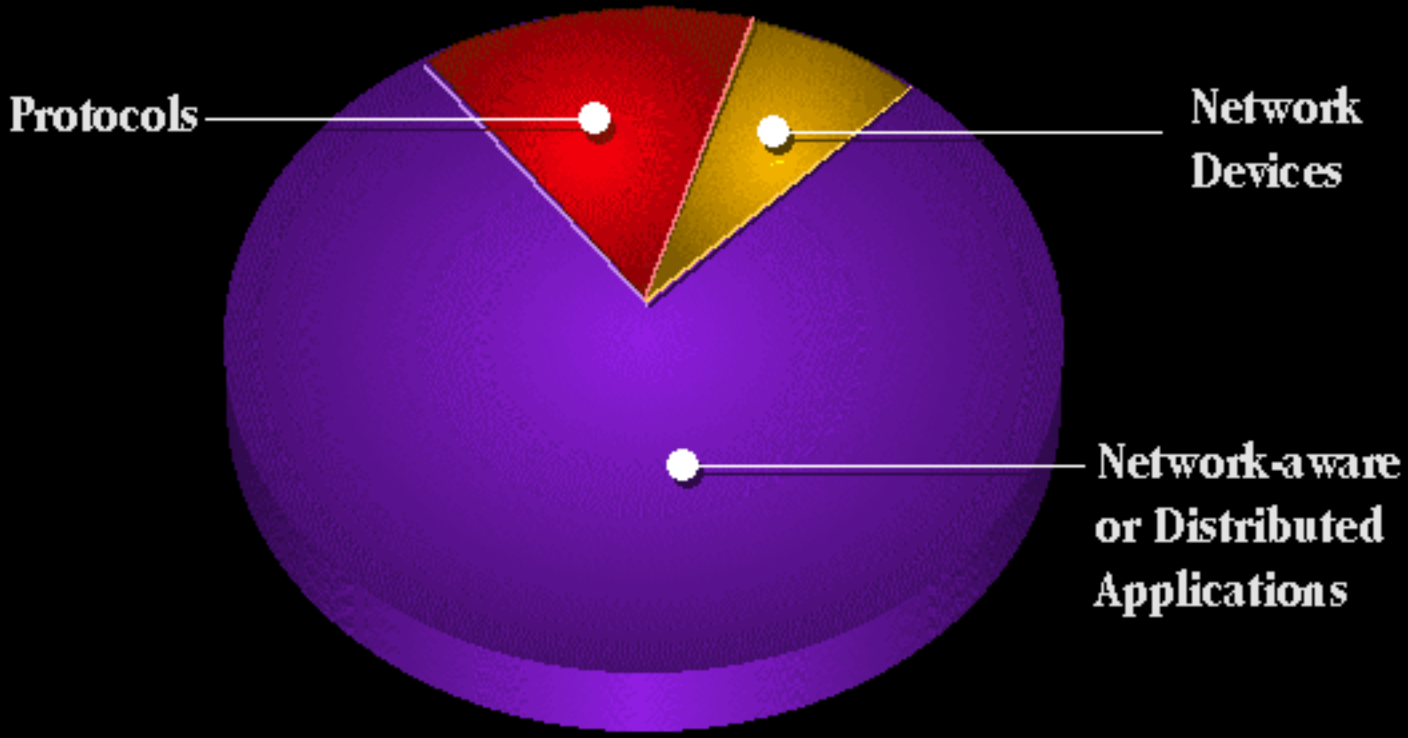
Manager
Core OS Networking

Rhapsody Networking

Combining Apple advantages with a robust infrastructure to enable the creation of unique, network-centric products

- Apple Advantages
  - Plug and Play, Configuration
- Robust Infrastructure
  - Proven Stacks and APIs
  - Time To Market, Performance
- Unique products for a network-centric world

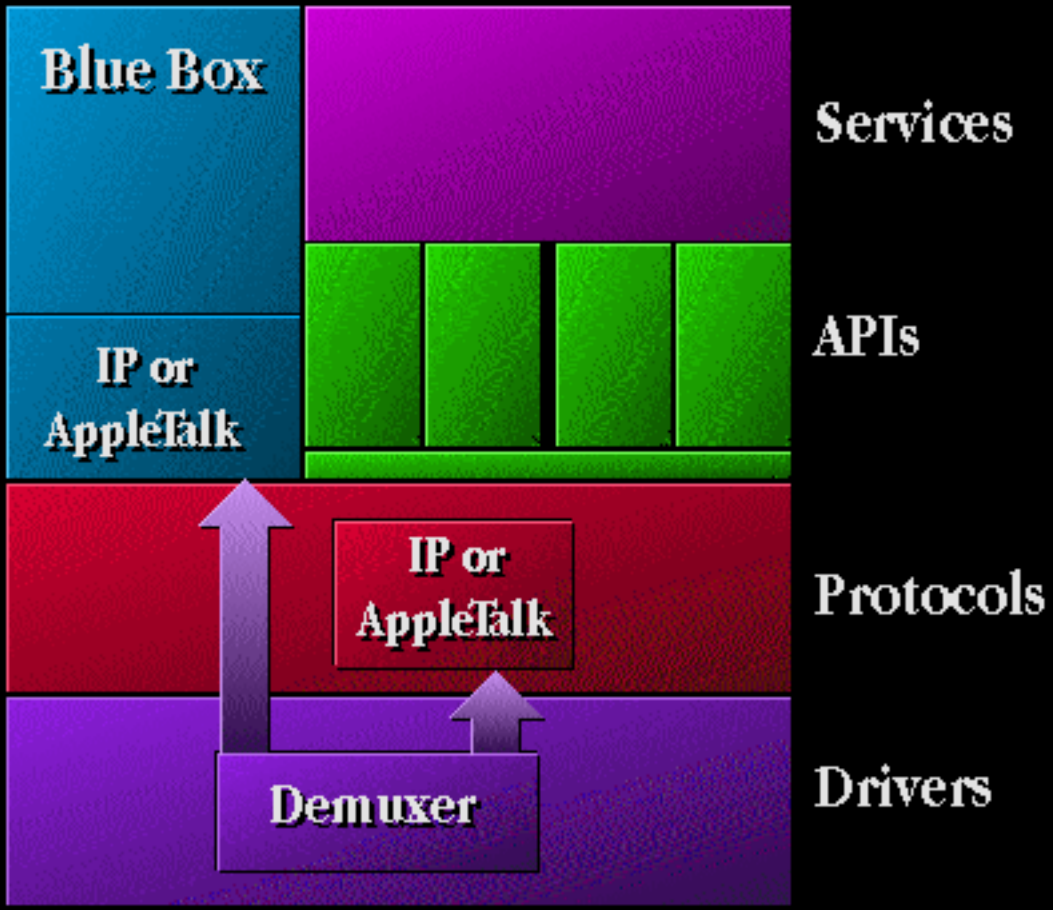
Developer Community



Topics

- Network Devices
- Protocol Stacks
- Applications and Application Services

Network Architecture

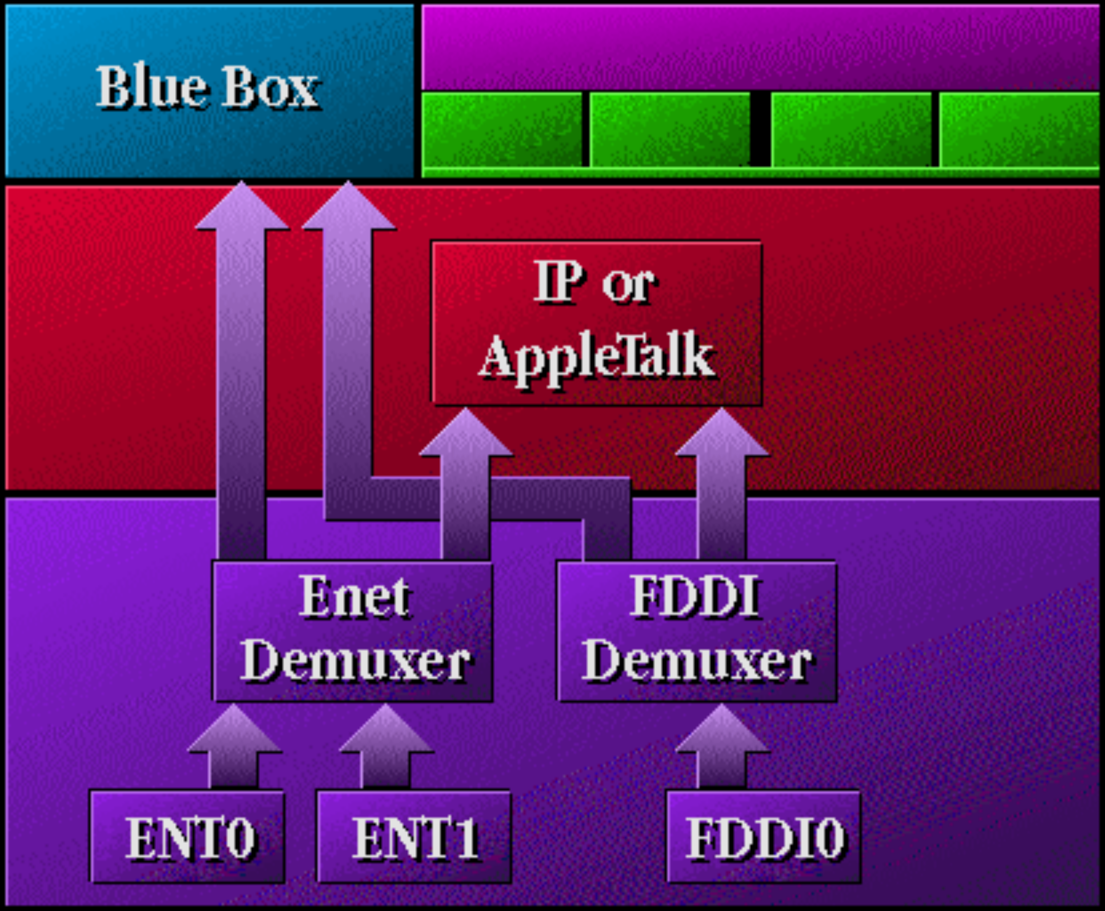


Network Devices

IOKit Network Drivers

- Object Model
- Plug and Play Configuration
- Multi-protocol support
- Modified API from DriverKit

Network Devices



Network Devices

Media Support

- Ethernet, FDDI, …
  - Ethernet (10, 100) Now
  - FDDI Soon
- PPP
  - IPCP Now
- New media support

Network Protocols

Protocols Supported

- Primary Protocol Support: TCP/IP, AppleTalk
  - Plug and Play, Ease of Use
- In addition: Netware, SMB
- Protocols can be added (Mach LKS)

Network Protocols

- TCP/IP Stack: BSD 4.4
  - Current with most IETF RFCs
- Socket APIs for TCP/IP
- AppleTalk Stack: ANS 700
- AIX APIs for AppleTalk

Network Protocols

The TCP/IP Stack

- IP Routing
- IP Multicast Support
- Multihoming, IP Aliasing
- Raw Sockets (protocol, device)

Network Protocols

The AppleTalk stack

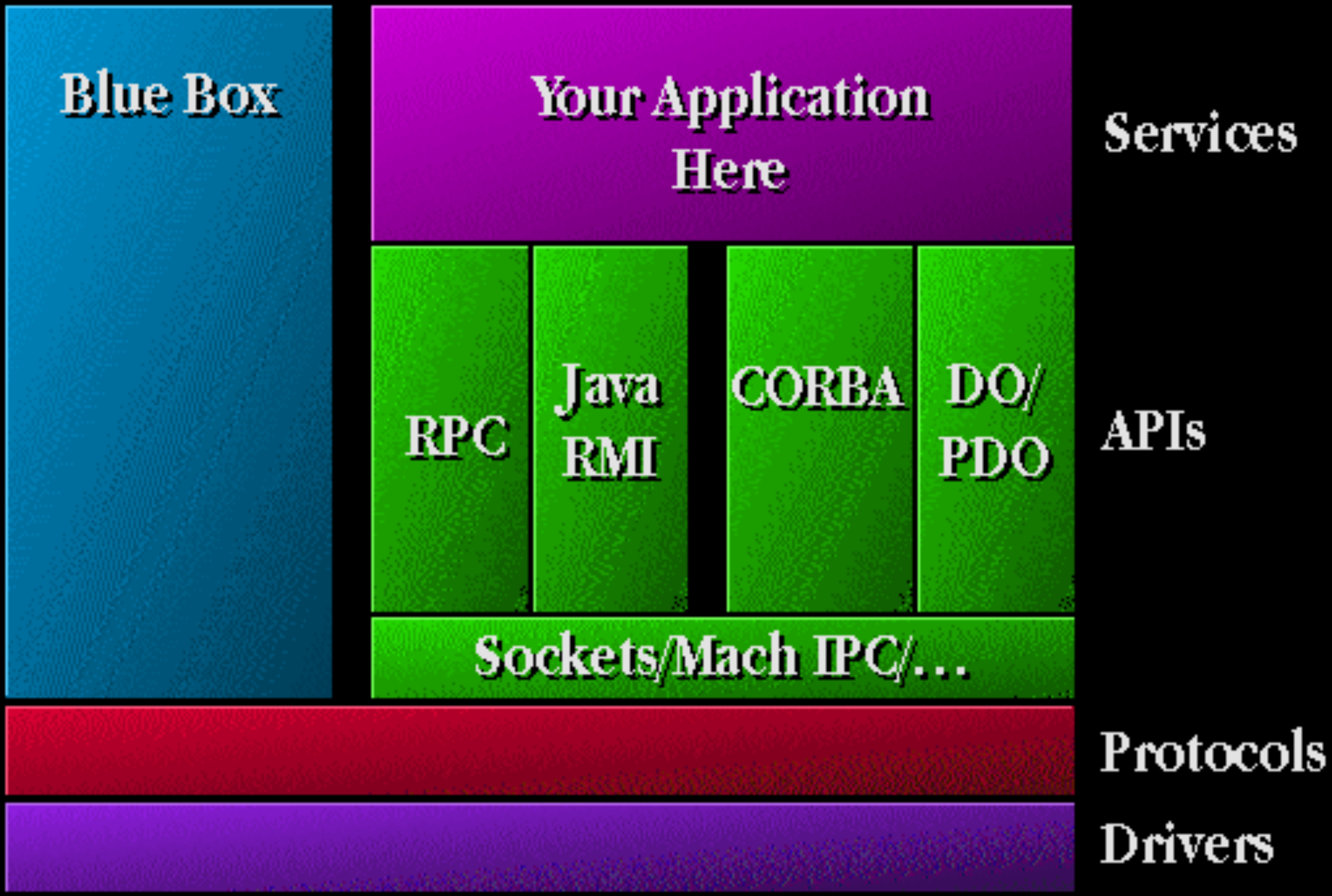
- Apple Network Server Code Base
- High Performance
- Routing (RTMP, AURP)
- Multihoming
- MP efficient on ANS (we know where the locks go)

Network Protocols

Network Services

- The usual suspects (Bind, NIS, …)
- NetInfo
- NFS (Versions 2, 3, NQNFS)
- Multicast Routing

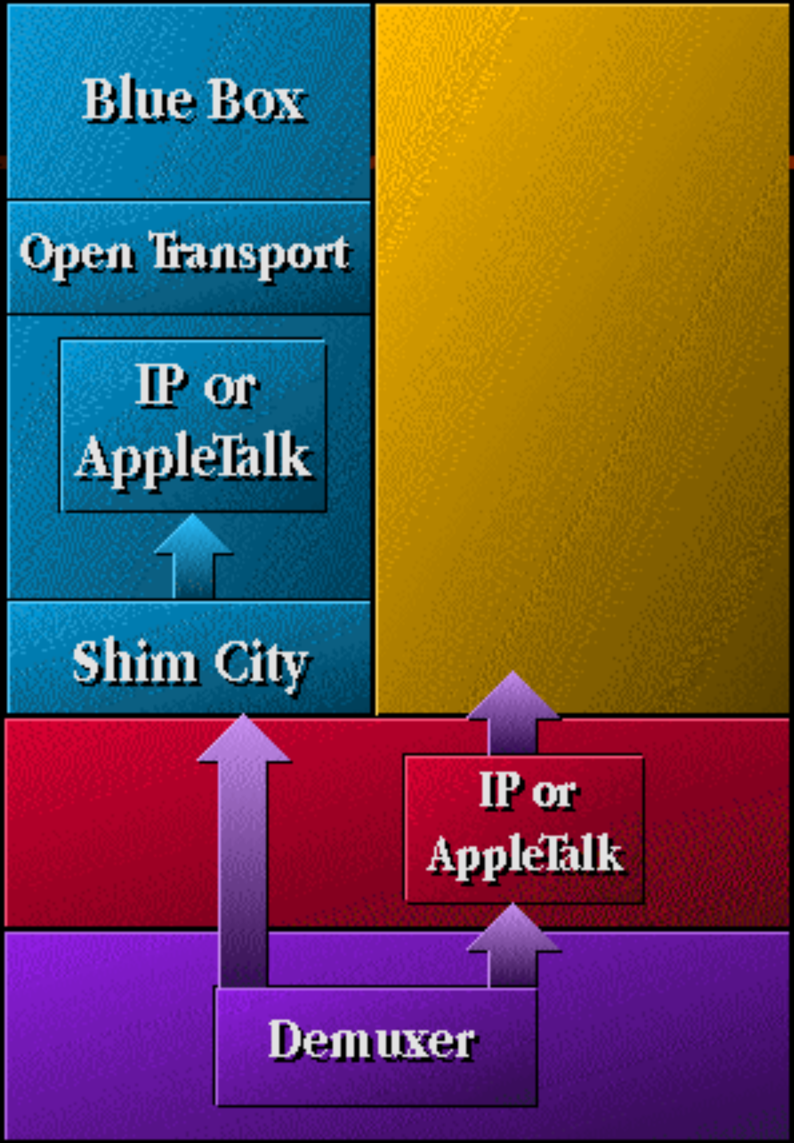
Network Apps, APIs, and Services



Network Apps, APIs, and Services

- The Blue Box
  - One Special Application
- Network APIs and Services

The Blue Box



Protocols

Drivers

Blue Box Support

"Full" Mac OS networking support

- Native OT above the hardware/driver layer
- Support in IOKit
  - A demuxer for packet delivery
- Protocol Configuration
  - TCP/IP: one address required
  - AppleTalk: independent stacks

Network APIs

Platform-specific and cross-platform APIs—your choice

- Mach IPC APIs
- Socket API for TCP/IP (Classic BSD)
- AIX AppleTalk API
- Objects
  - PDO
  - RMI
  - CORBA

Network APIs—the Low Level

Low-level, cross-platform or platform-specific

- Mach IPC
  - Cross-platform
- Sockets/AppleTalk
  - Platform-specific

Rhapsody Network APIs— The Object View

Blaine Garst

Yellow Box Dude

Network APIs— Objects and the High Ground

Distributed Programming is still hard

- Problems
  - Security
- Solutions

Network APIs— Objects and the High Ground

Distributed Programming is still hard

- Problems
  - Security
  - Availability
- Solutions

Network APIs— Objects and the High Ground

Distributed Programming is still hard

- Problems
  - Security
  - Availability
  - Scalability
- Solutions

Network APIs—
Objects and the High Ground

Distributed Programming is still hard

- Problems
  - Security
  - Availability
  - Scalability
  - Reliability
- Solutions

Network APIs—
Objects and the High Ground

Distributed Programming is still hard

- Problems
  - Security
  - Availability
  - Scalability
  - Reliability
- Solutions
  - $$$$$$$$$$

Network APIs— Distributed Programming

Many approaches

- Fat clients
  - TP Monitors
  - Custom solutions
- Thick clients
  - Distributed programming
- Thin clients
  - Web technologies

Distributed Programming

Fat client: Divide and Conquer!

- Use databases for data
  - Data in RDMS, flat files, wherever
- Custom logic in “object” servers
  - Build object “schema” with EOModeler
- Vend via Client application
  - Custom UI (custom/standard widgets)
  - “Associations” synchronize views

Distributed Programming

Thick client

- RMI
- CORBA
- Distributed Objects

Distributed Objects (a.k.a. PDO)

- Extensible
  - Substitute underlying transport
  - Supply security layer
  - Substitute naming layer
- Integrated support in ObjC language
  - oneway, in, out, inout keyword support
  - byref, bycopy object copying directives
- Transparent programming
  - Exceptions, objects, etc. flow unimpeded
  - Garbage collection

Distributed Objects

Client and server (!!) coding example

- id client = [NSDistantObject proxyWithName:@“ideas” host:@“apple.com”];
- [client postSuggestion:@“buy NeXT”];
- id conn = [NSConnection connectionWithRoot:[Ideas new] name:@“ideas”];
- [[NSRunLoop currentRunLoop] run];

Web Programming

Thin client

- Browser based clients
  - Relieves worry about application distribution
- Server based computation
  - Somewhat scalable
  - Little to no transactional state

Web Programming: WebObjects

Object Framework for composing dynamic web pages

- 3+ tier architecture
- Web clients
  - Web servers (Netscape, Microsoft, Apache)
  - Object Servers (EOF)
  - Database Servers
- JavaScript, Java applets, Frames, etc., support

Q&A

Worldwide Developers Conference