# MAC OS X

# A UNIX™ Geek's perspective

Jordan Hubbard
Engineering Manager, BSD Technology Group
Apple Computer

# Who am I?

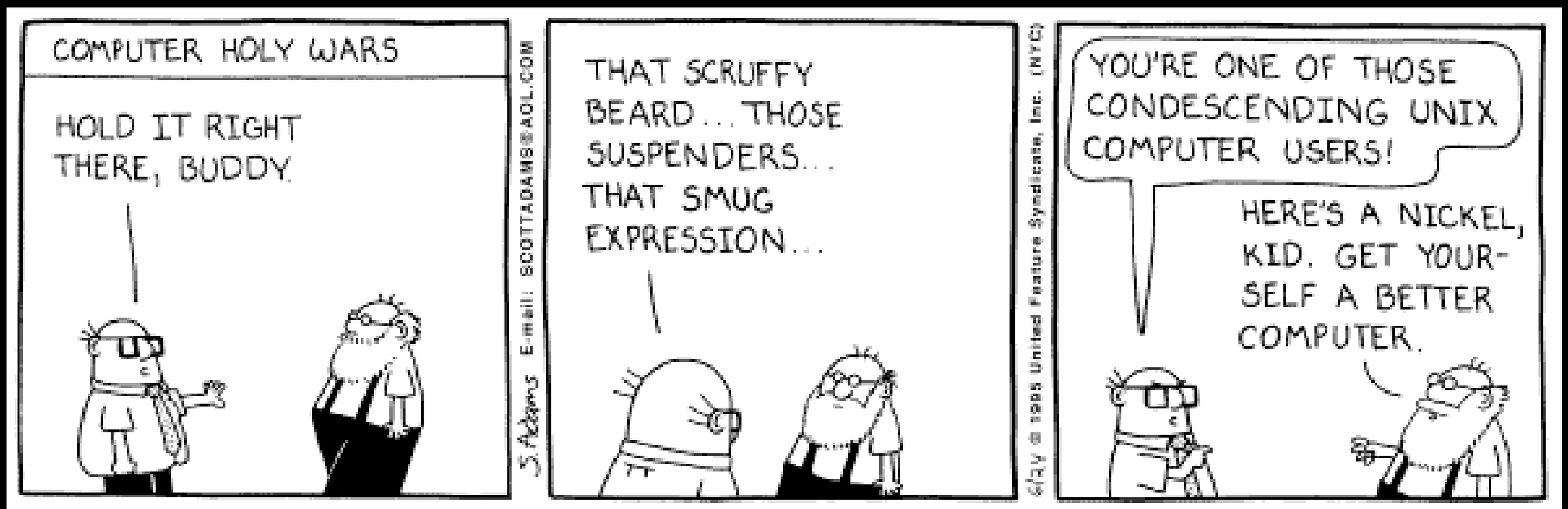
- Long-time contributor to the Open Source community
  - Volume 1 of comp.sources.unix even, which makes me old
- Background as a UI designer and big early advocate of the X Window System - wrote "awm", the first reparenting Window Manager, various toolkits and widgets, etc.
- Long-suffering administrator at U.C. Berkeley
- Co-founder of the FreeBSD project and benevolent dictator of it for abount 8 years
- Over 20 years of Unix development, but comparatively new to Apple and the Macintosh platform
  - Only came to Apple once it had a **real** OS

# Why UNIX was the right technology

- Highly "composeable" as operating systems go
  - It's an onion, not a potato
- It gave us a huge amount of open source to leverage and this was critical to the implementation process
- Instant portability for a huge number of important applications (and important users) in SciTech and other fields
- Interoperability with *BSD, Linux, Solaris and other UNIX-derivatives came almost for free

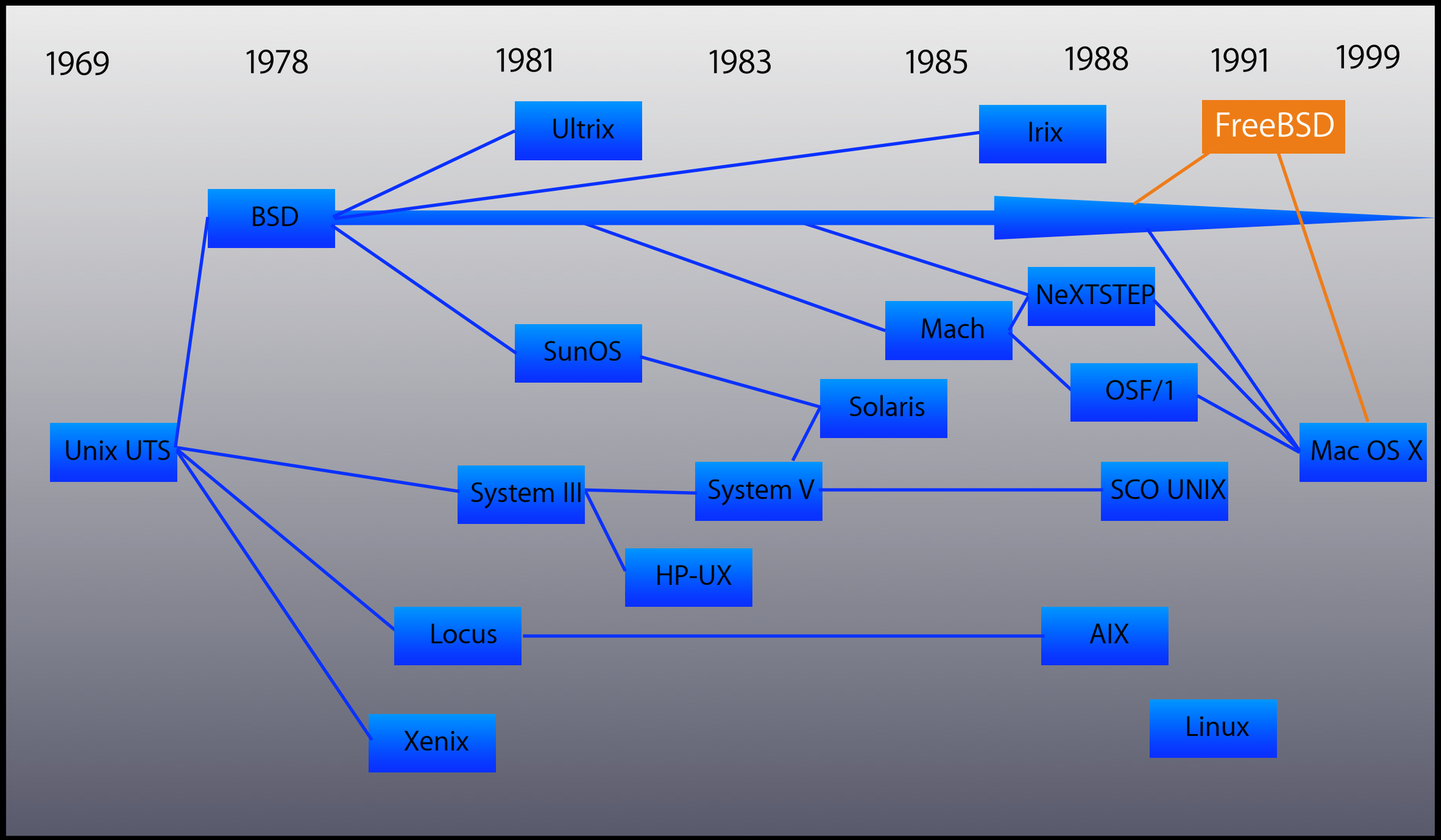

# Why UNIX was the right technology

- Development community is active, innovative and has a strong and well-established track record on OS design
- Influential in decision making

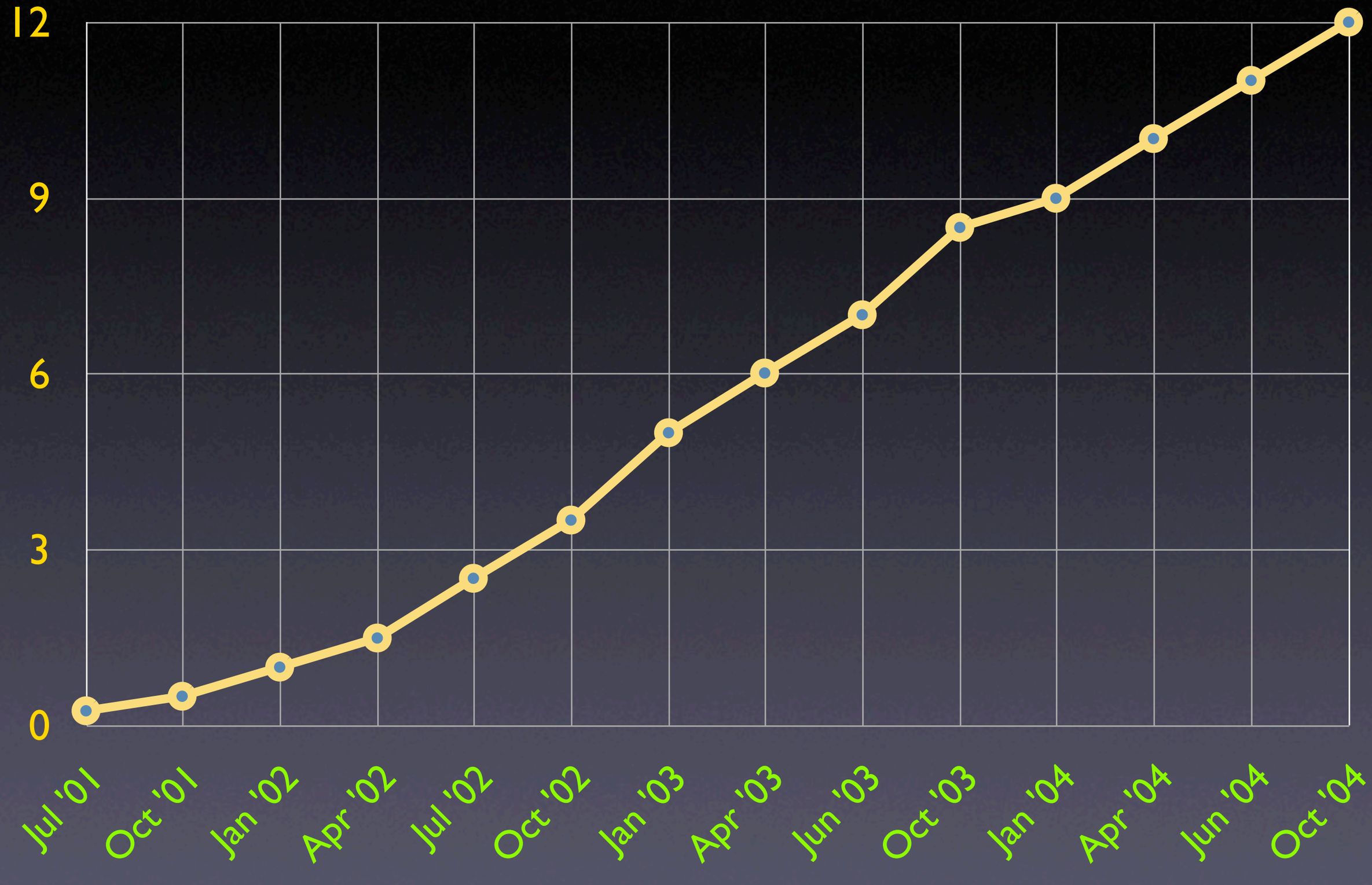



UNIX BASED

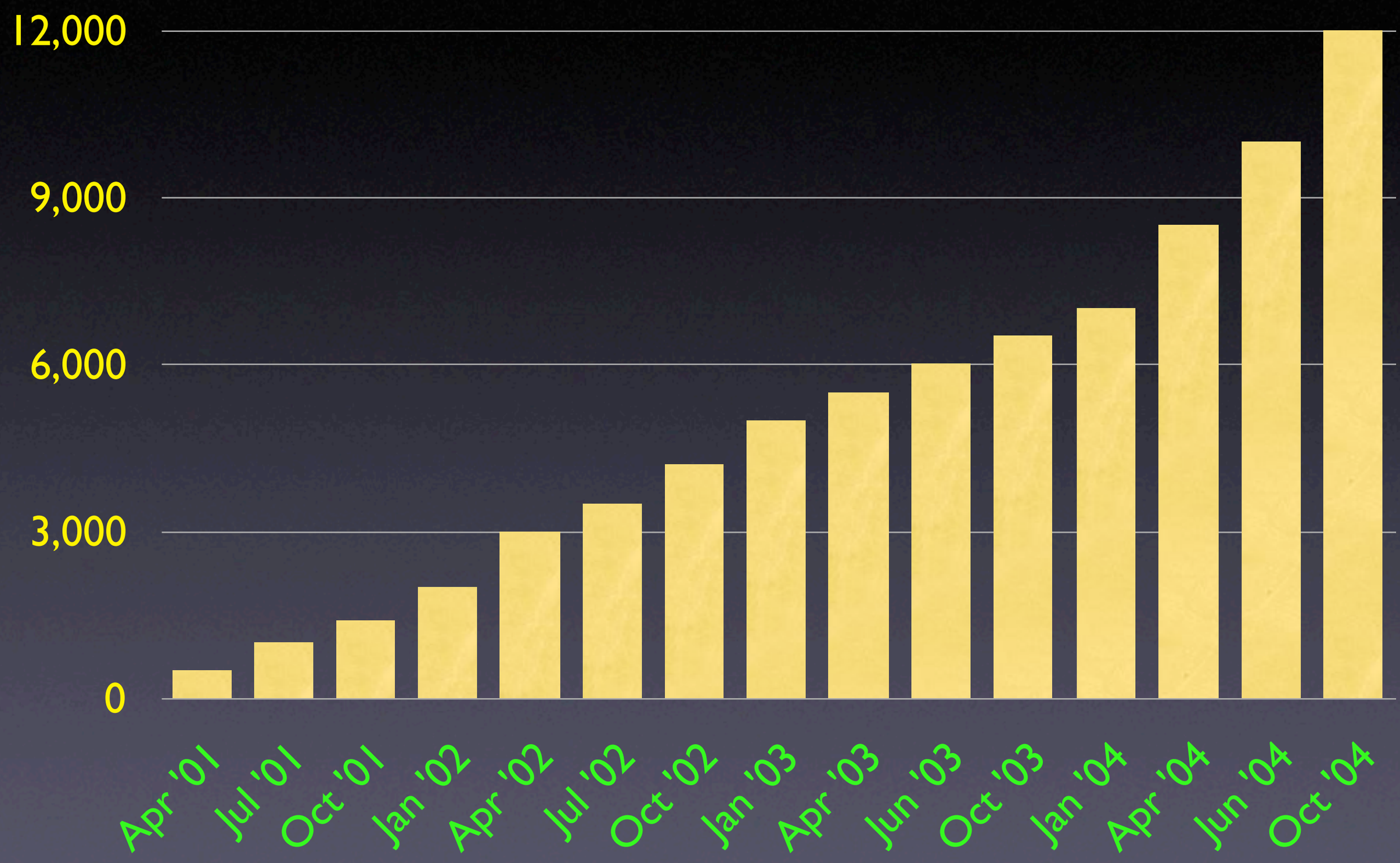
# Unix Family Tree



# Mac OS X Users: 12 Million



# Applications: 12,000 Mac OS X Native

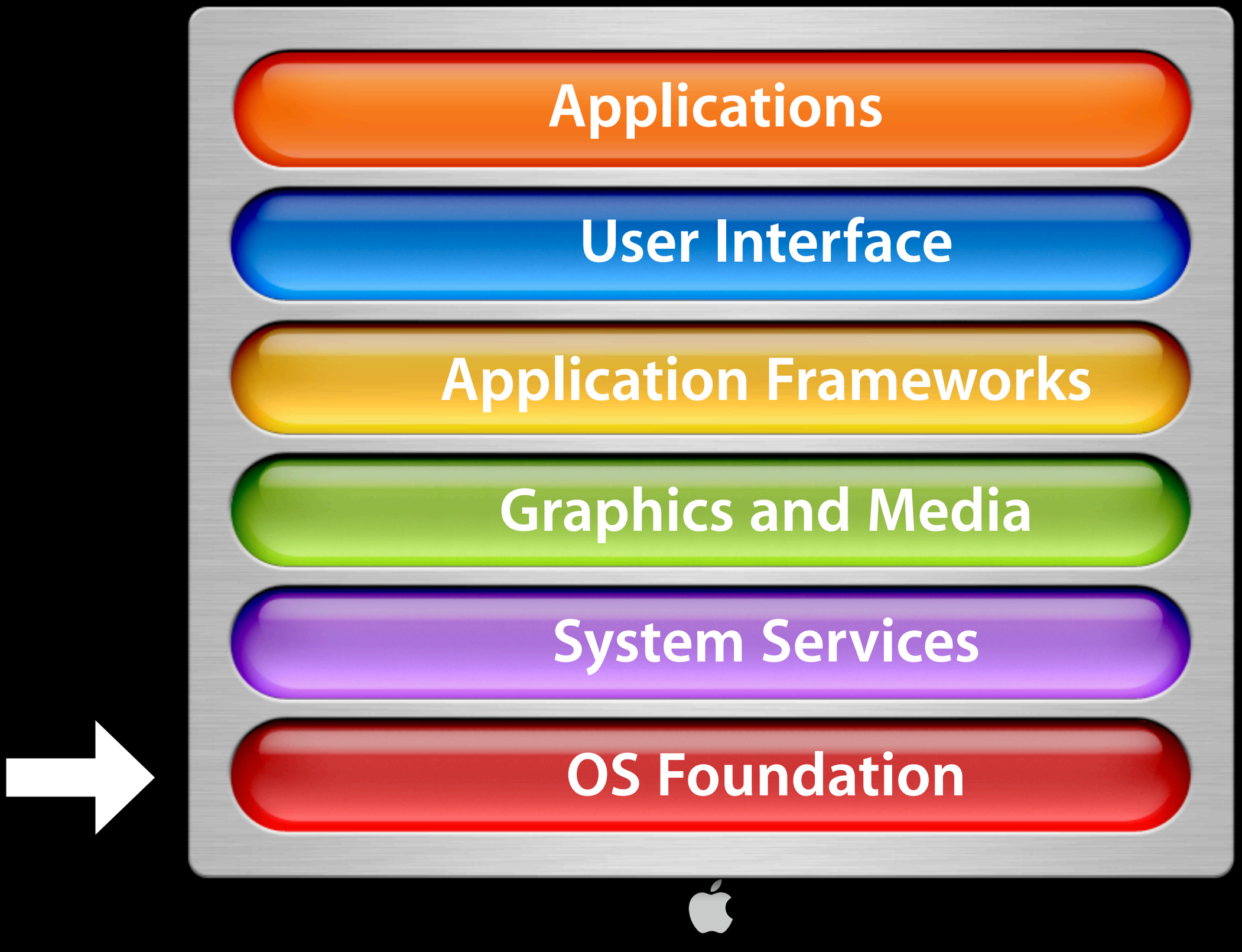


# Mac OS X is now the biggest desktop UNIX variant on the planet

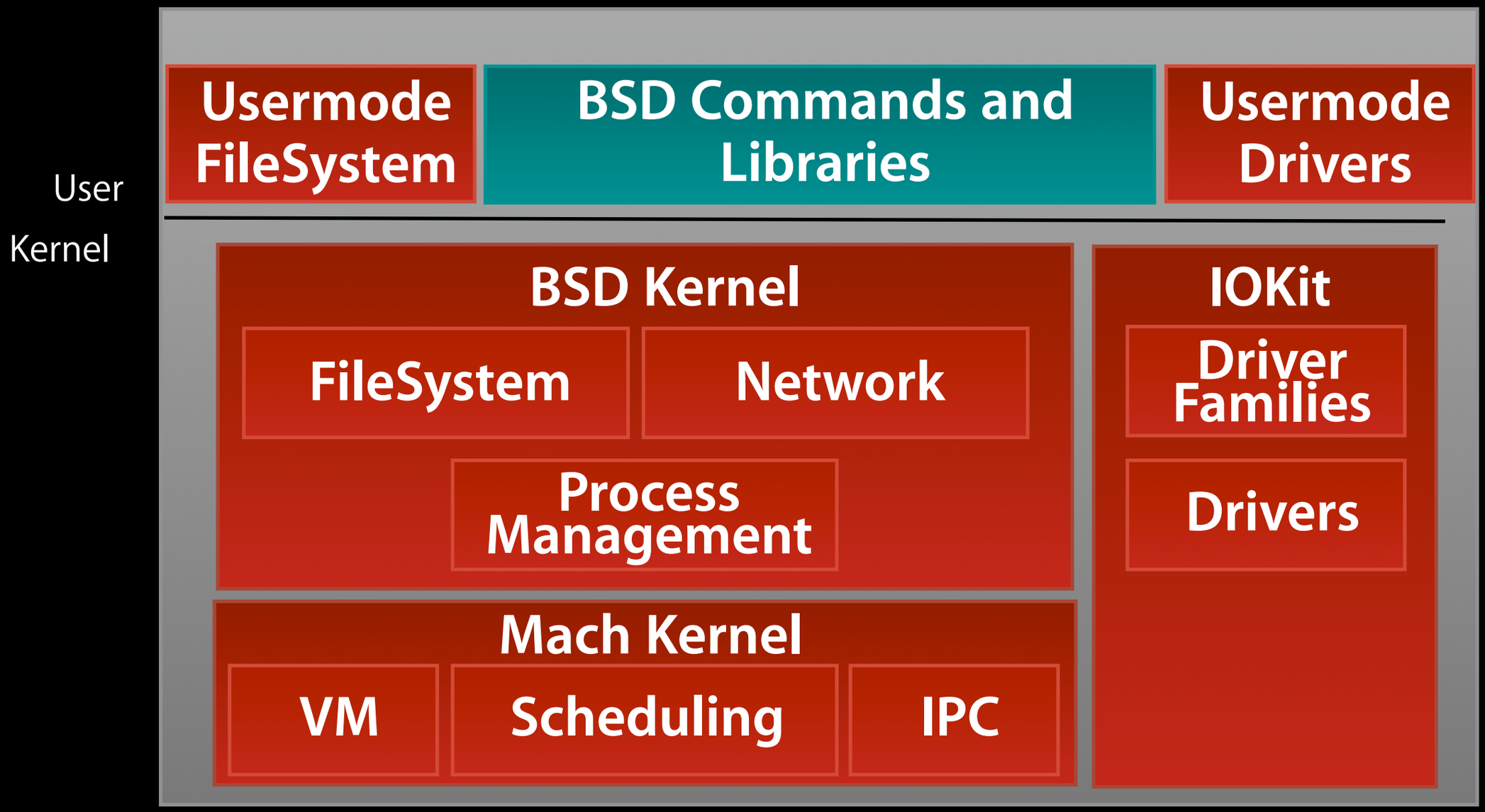
# Mac OS X 10.3

## Panther, a quick overview

# Mac OS X Architecture

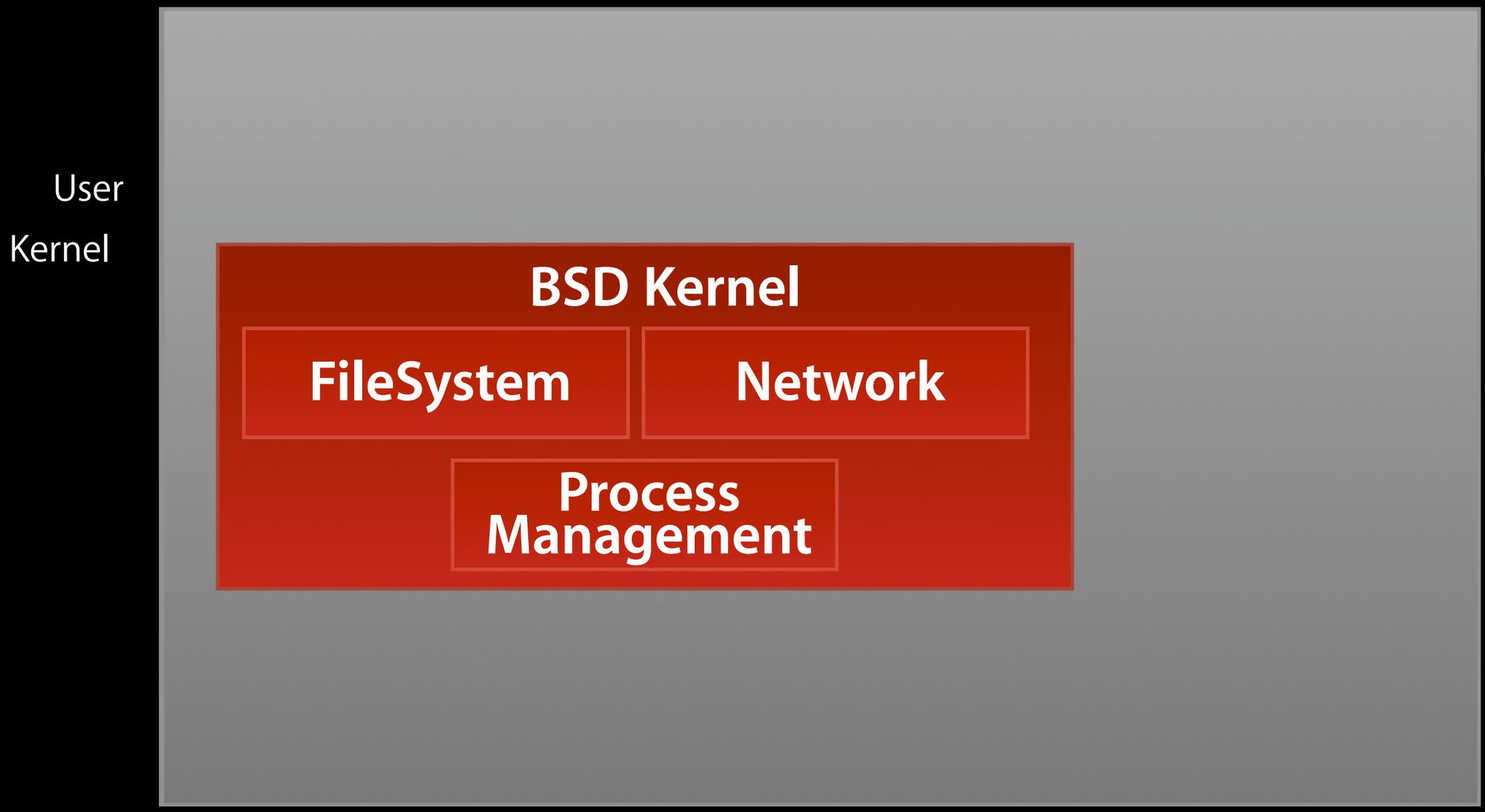


# OS Foundation




Open Source "Darwin" base

# OS Foundation



# BSD Kernel

- FreeBSD 4.8 based (networking, vfs, filesystems, etc)
- Unified Buffer Cache (different than FreeBSD's)
- Clustered I/O performance enhancements
- Local File Systems
  - hfs, ufs, iso9660, udf, fat, ntfs
- Network File Systems
  - nfs, afp, smb, webDAV, ftpfs

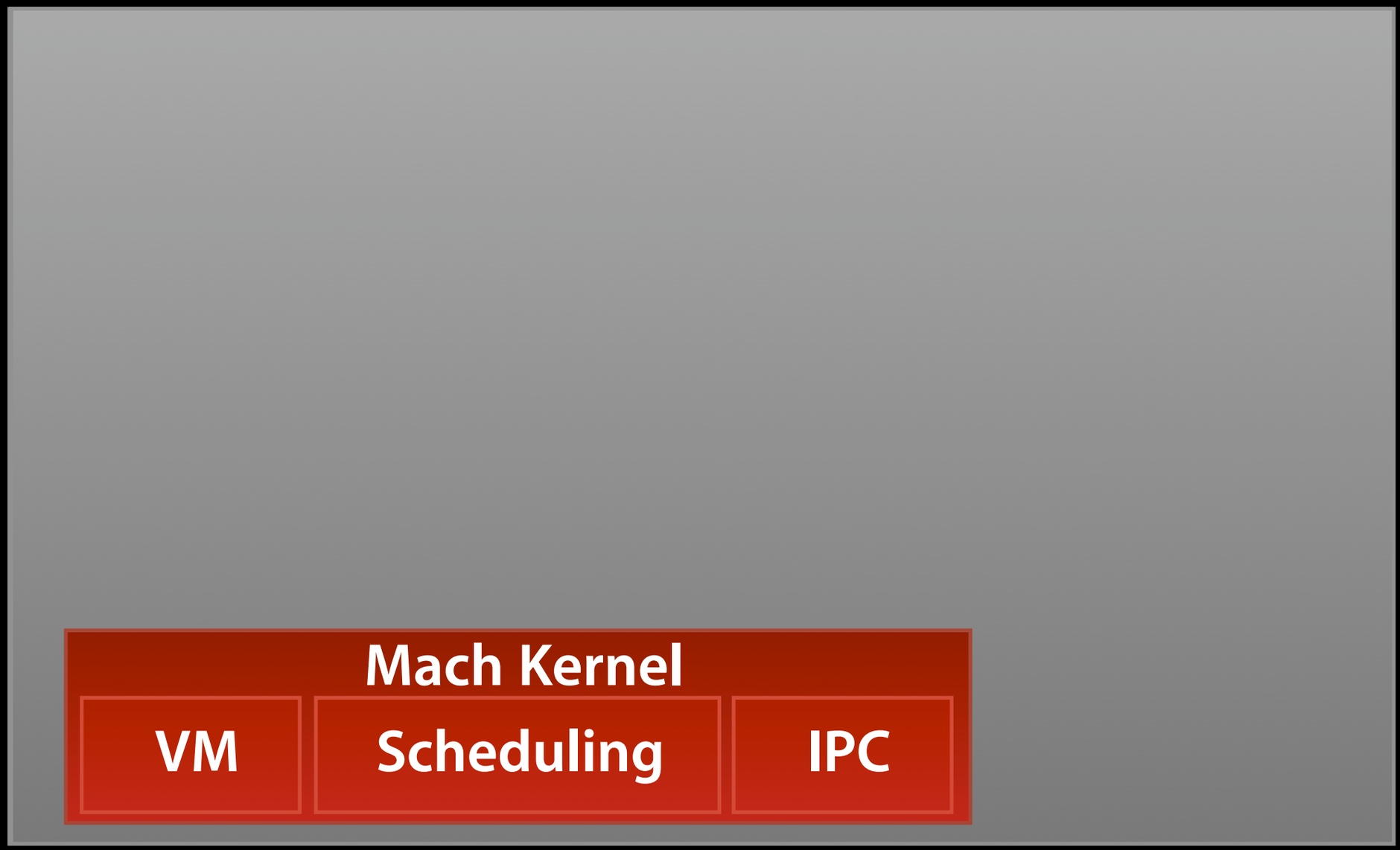
# BSD Networking

- Full IPv6 support
- L2TP/IPSec VPN client and server
- 802.1x wireless authentication (TLS, TTLS, LEAP, PEAP,...)
- Firewall based on ipfw
- Network Reachability APIs

# OS Foundation

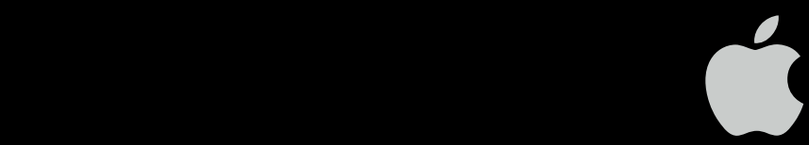
User

Kernel



# Mach Kernel

- Based on Mach 3
- VM, tasks, threads, scheduling and IPC
- Fine grain locking for SMP
- Support for > 4GB Physical memory
- [fairly] Light-weight threading model makes aggressive threading more practical
- Real-time scheduling
- Event driven application programming model (via Mach ports)



# OS Foundation

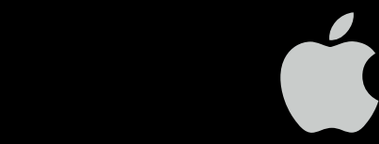
# IOKit

- Written in conservative C++
- OOP device family and instance model
- Support for user space drivers
- Dynamic plug and play
- Handles all device property information and provides convenient introspection via ioreg(1) and friends
- Sophisticated power management

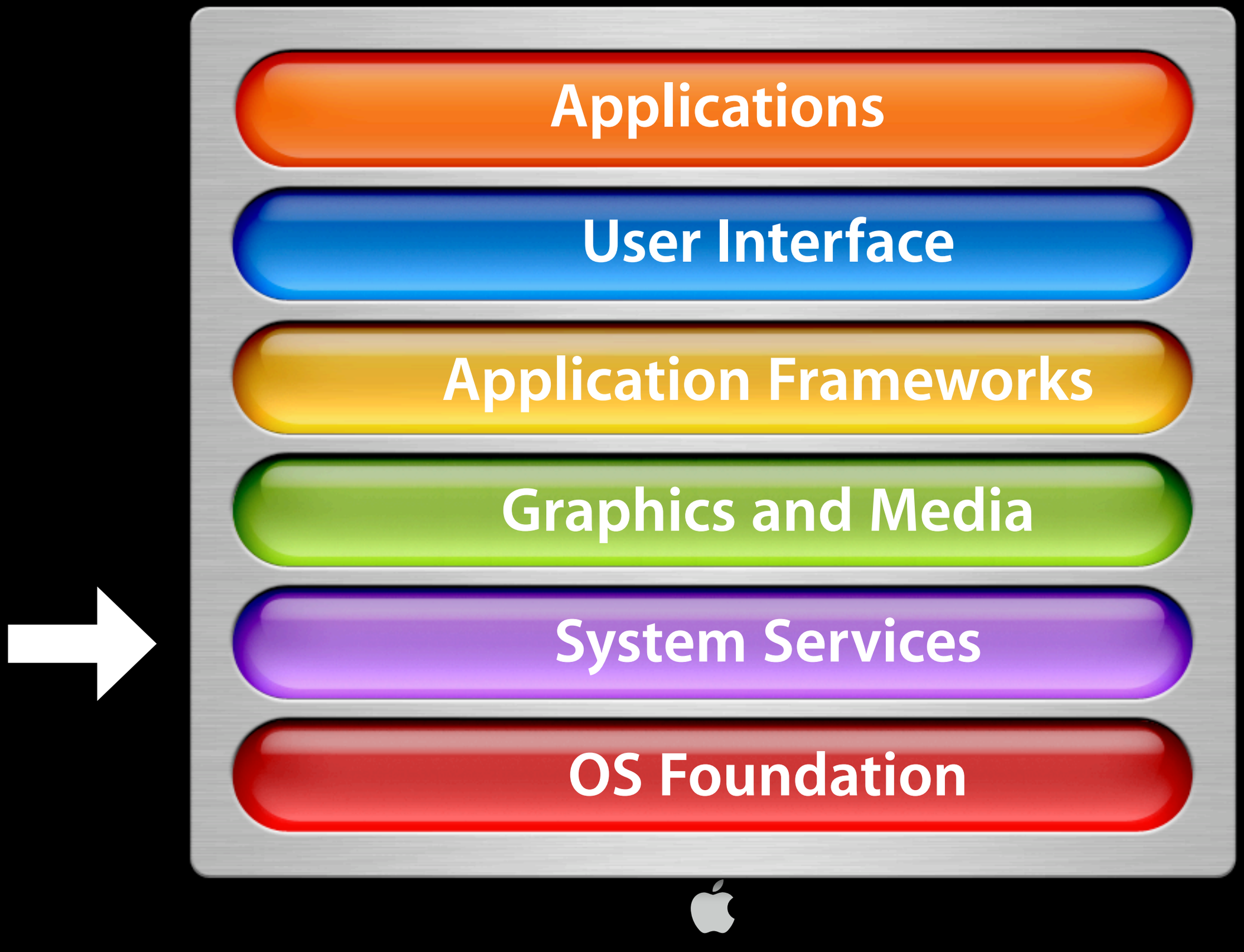
# OS Foundation

# Commands and Libraries

- Standard commands and libraries from FreeBSD 4.8
- A full suite of scripting languages
  - perl, tcl, python, ruby, php
- Every standard shell
  - bash, csh, tcsh, zsh, etc
- Standard editors
  - pico, vi, emacs (the only one you actually need)
- Standard C compiler suite
  - gcc, g++, Objective-C [version 3.3]

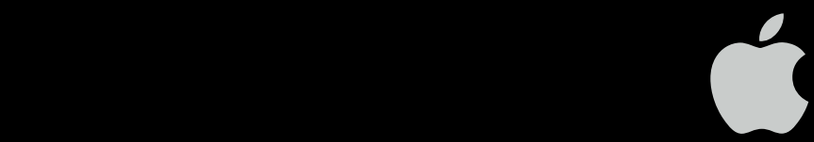


# Mac OS X Architecture



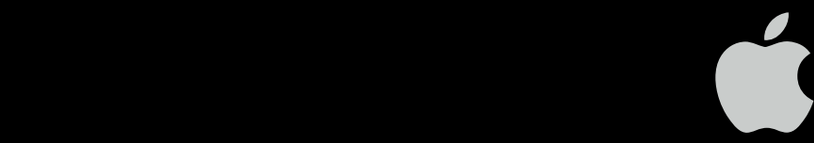
# Open Directory

- Flexible plug-in architecture
  - Supports legacy flat files
  - Supports OpenLDAP
  - Supports Active Directory
- Open Source
  - http://developer.apple.com/darwin/projects/opendirectory/



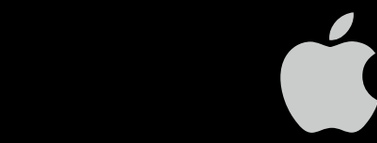
# Security Server

- Full CDSA (Common Data Security Architecture) implementation
- Plugin-based authentication
- Implements keychains for easy access
- It's not OpenSSL
- Open Source references:
  - http://developer.apple.com/darwin/projects/security/
  - http://sourceforge.net/projects/cdsa/

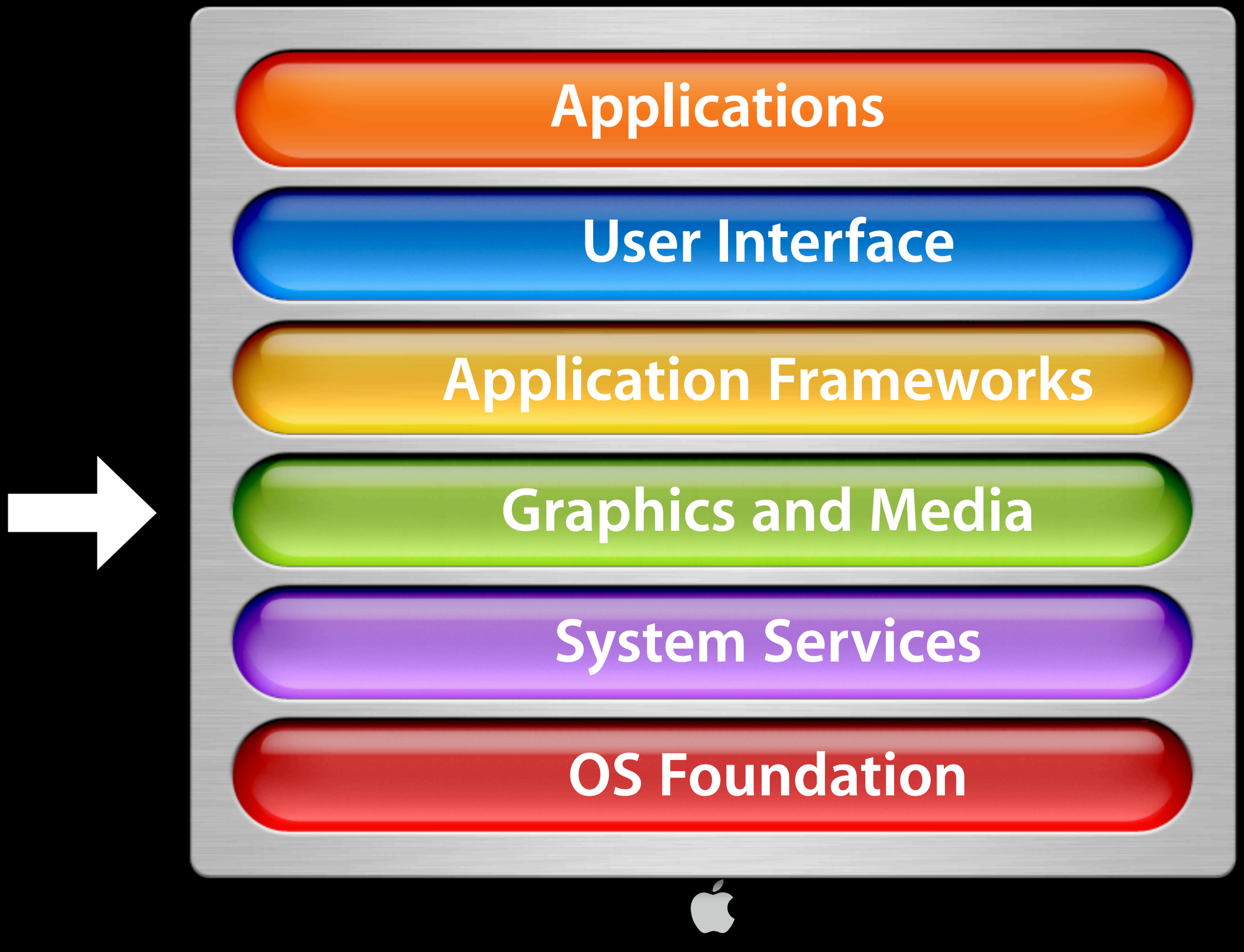


# Rendezvous

- Service registration
- Service discovery
- Easy ad-hoc networking via .local namespace
- Also available for FreeBSD, Solaris & Linux (and a number of misc devices)
- Open Source references:
  - http://developer.apple.com/macosx/rendezvous/



# Mac OS X Architecture



# 2D Graphics: Quartz

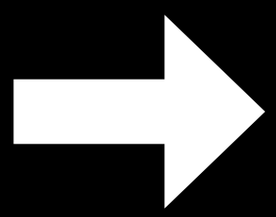
- PDF-based imaging model
- Leverages GPU
- CUPS "WYSIWYG" printing
- Python bindings
  - CoreGraphics APIs
  - QuickTime images
  - PDF, RTF, HTML

# 3D Graphics: OpenGL

Industry Standard
3D Technology


Keeping the world safe from DirectX

# Mac OS X Architecture



Applications

User Interface

Application Frameworks

Graphics and Media

System Services

OS Foundation

Finder File Edit View Go Window Help

Inbox

Delete Reply Forward New Junk Get Mail Search

From Subject Date Received

Galen Stackwell Greetings Today

3 John Smith Summer Vacation Yesterday

John Smith Trip

Kathryn Lemay Sun

Galen Stackwell Goi

Michelle Re:

Ed Thompson Gre

Galen Stackwell Last d

Mike Status

Juliette Walsh Need

From: Kathryn Lemay
Subject: **Summer Vacation**
Date: July 21, 2003
To: Todd Taylor

Documents

home

Mac OS X Panther

Network

My Archives

iDisk

Home

Applications

Documents

Desktop

Pictures

Project folder

Project folder Letters Downloads

Stuff Work in progress Meeting notes

logo.jpg Reports Updated list

Images to process Adrien Picture of Beach

25 items, 13.5GB available

Video Chat with Aki

# Also supports X11

- Based on XFree86 4.4
- Implements X11R6.6
- Includes basic X apps e.g. xterm, xcalc, xedit, etc
- Hardware OpenGL rendering
- Native Aqua and X11 applications run side by side
- Or you can run it in Full Screen mode

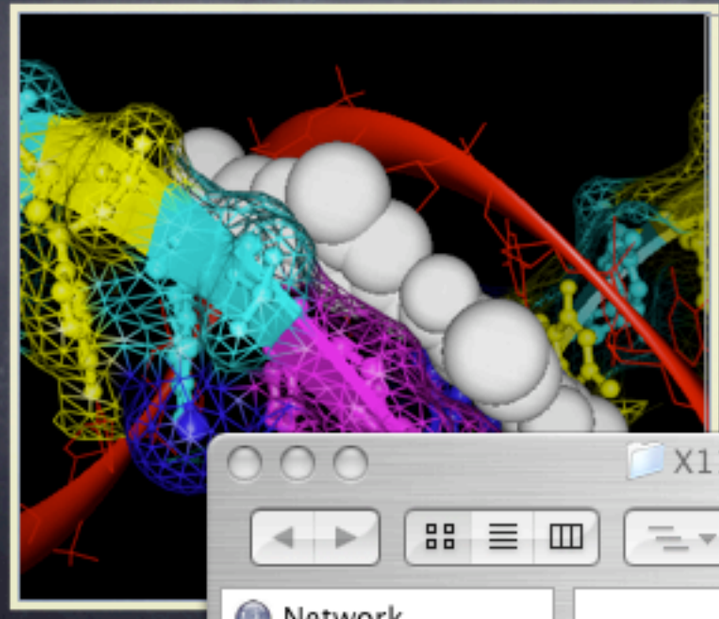
X11 Applications Edit Window Help

Fri 3:55 PM Apple

Netropsin Presentation.key

New Delete Play Themes View Masters Text Shapes Table Chart Inspector Fonts

DNA with Netropsi

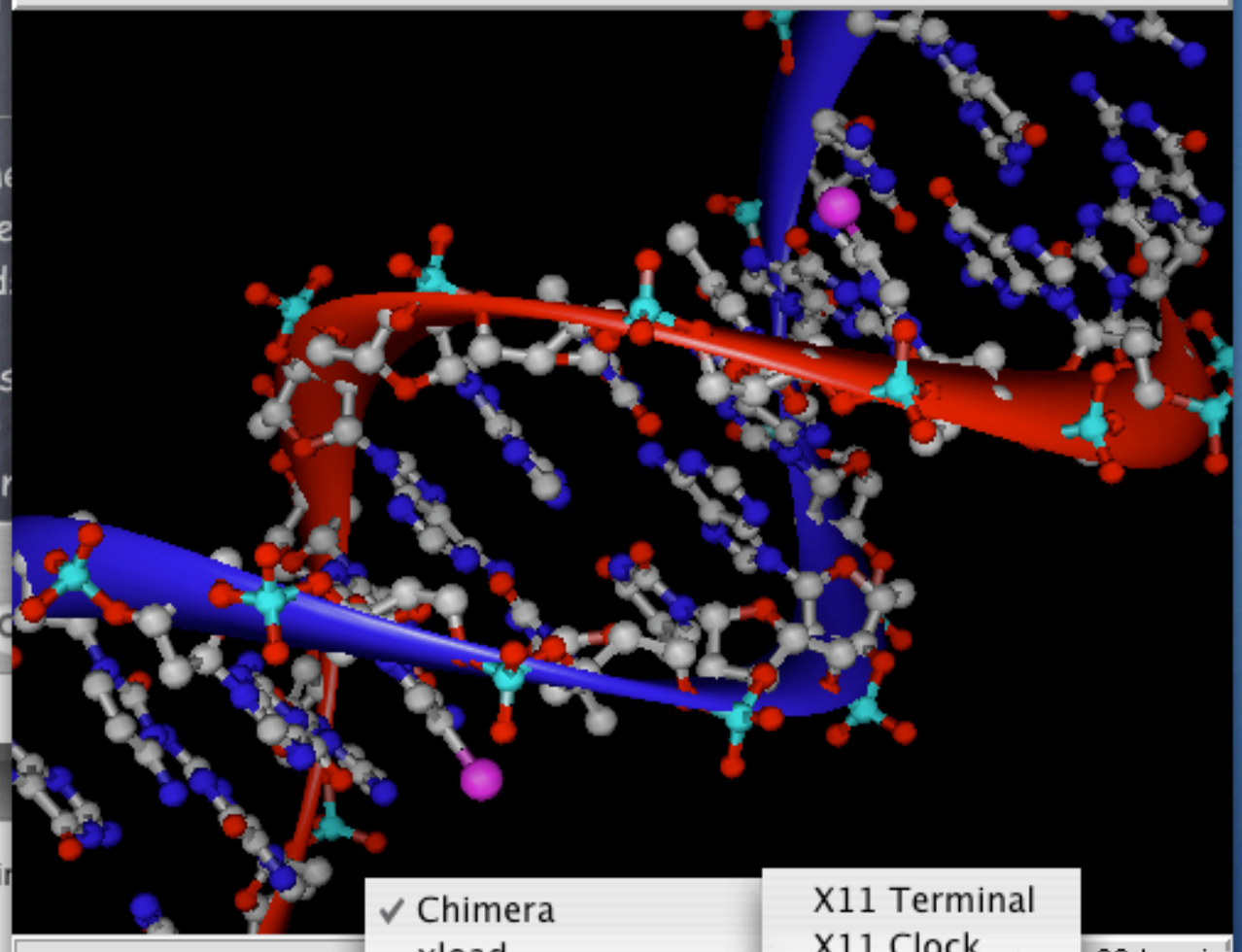


- Netropsin ne
  between the
  DNA strand
- Strand B dis
  affinity for
  chemical bor

66%

Chimera

File Select Actions Tools Favorites Help



32 bon

X11 apps

Network
Panther
Desktop
apple
Applications
Documents
Movies
Music
Pictures
X11 apps

cinepaint
openoffice
vim

6 items, 11.04 GB available

✓ Chimera
xload

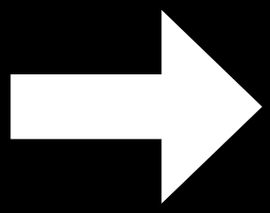
Applications

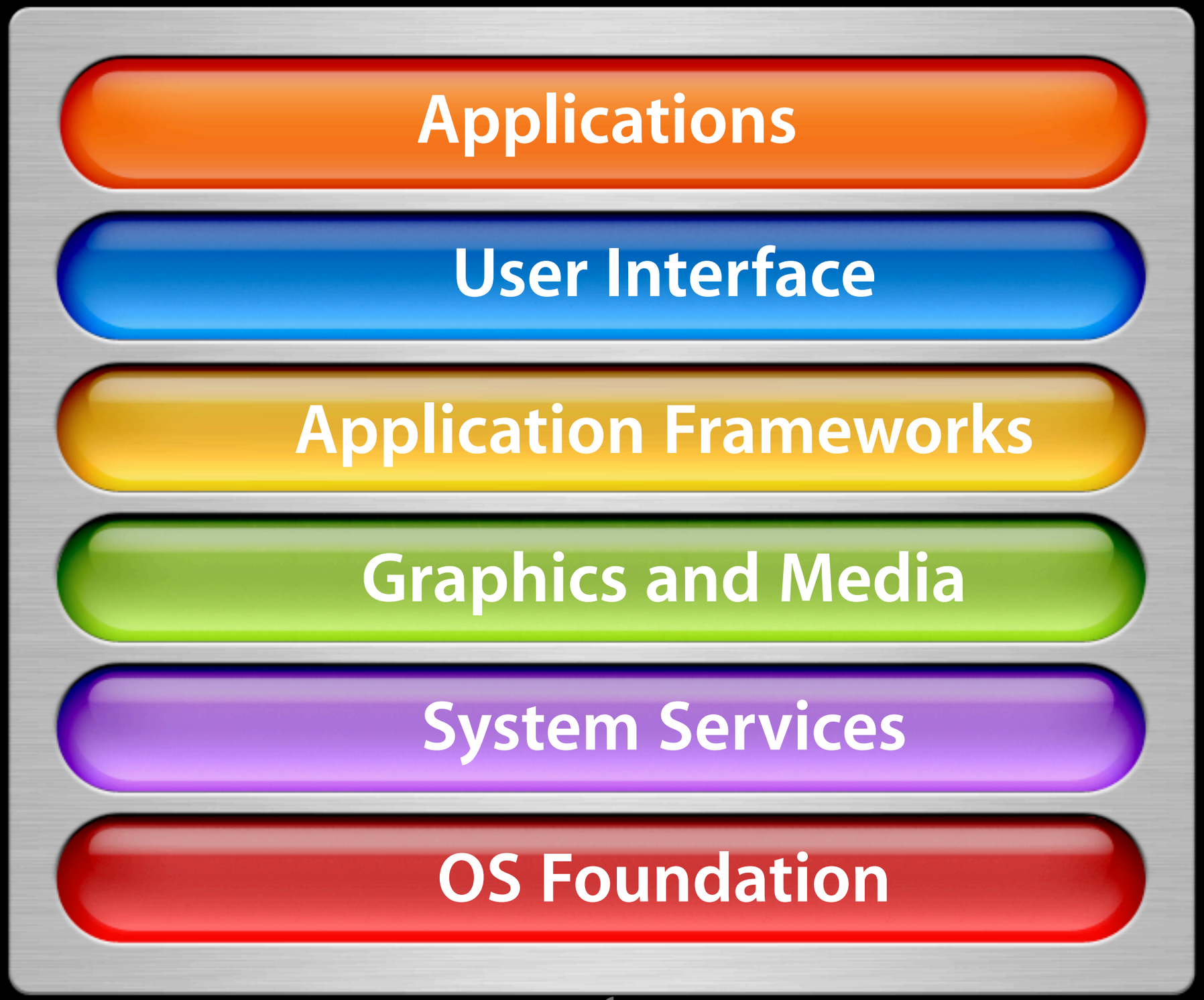
Show In Finder
Hide
Quit

X11 Terminal
X11 Clock
GLX Gears
GTK Gimp
Motif NEdit
Tk Getleft

Customize...

# Mac OS X Architecture

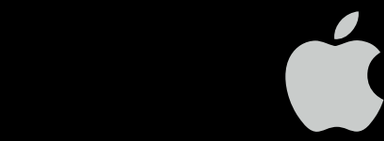




# Most of the important ones...

- Microsoft Office
- Photoshop
- Quicken / Quickbooks
- Quark Xpress
- Macromedia Director and Macromedia Studio
- ... and many many more, either here or coming soon

# The challenges of UNIX...

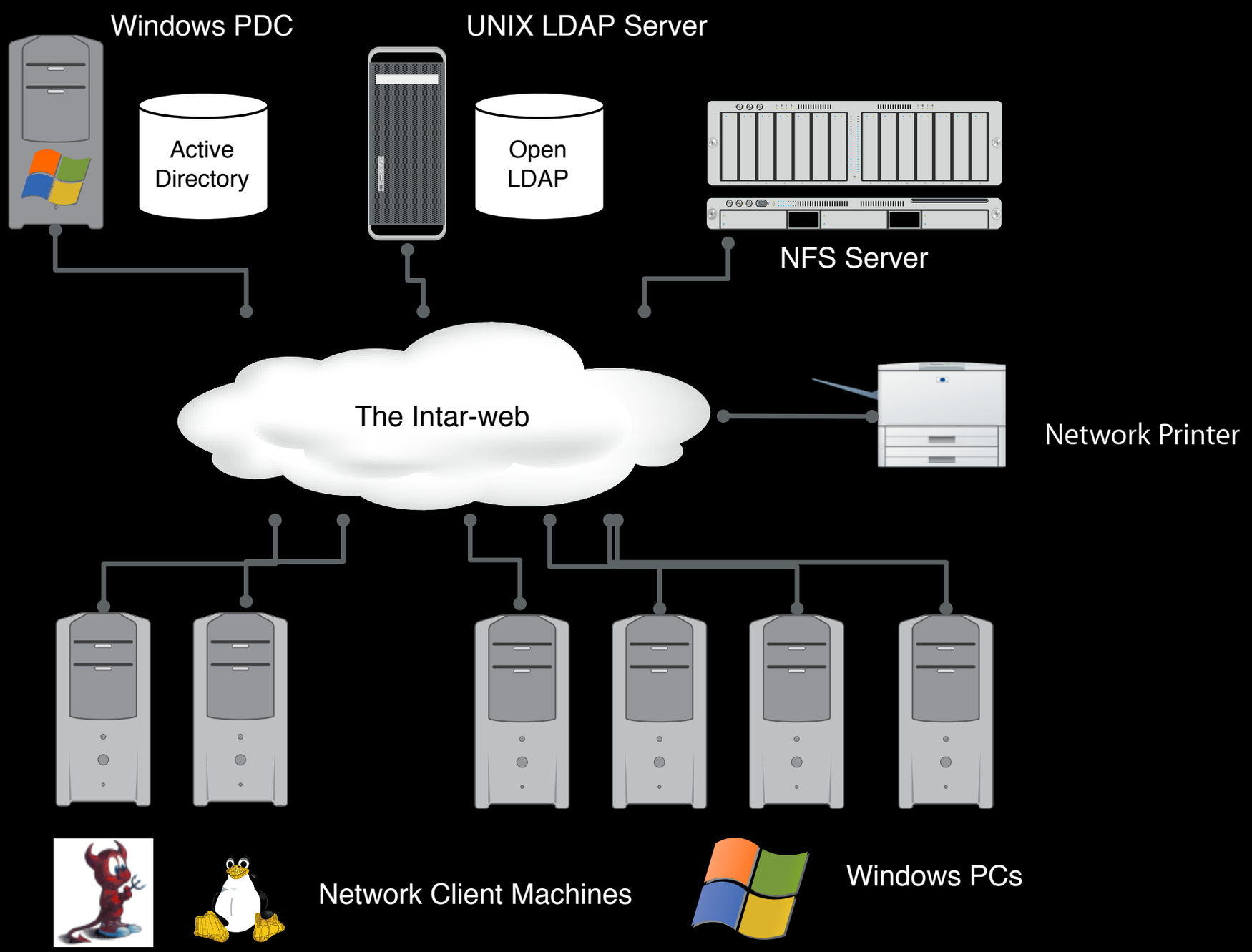


# Challenge: Authentication

Sorry, this is **not** a user information database:

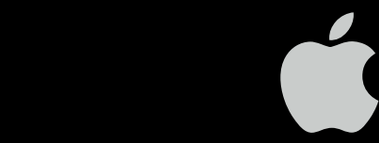
nobody:*:-2:-2:Unprivileged User:/:/usr/bin/false
root:*:0:0:System Administrator:/var/root:/bin/sh
daemon:*:1:1:System Services:/var/root:/usr/bin/false
smmsp:*:25:25:Sendmail User:/private/etc/mail:/usr/bin/false
lp:*:26:26:Printing Services:/var/spool/cups:/usr/bin/false
postfix:*:27:27:Postfix User:/var/spool/postfix:/usr/bin/false
www:*:70:70:World Wide Web Server:/Library/WebServer:/usr/bin/false
mysql:*:74:74:MySQL Server:/var/empty:/usr/bin/false
sshd:*:75:75:sshd Privilege separation:/var/empty:/usr/bin/false

# The present looks a lot more like this ...



# Challenge: Authentication

- The traditional UNIX group model is obsolete
- The uid is obsolete and insufficient - prepare for the GUUID (and privacy concerns)
- Smart Cards (and their successors) are in the future
- Kerberos everywhere: A good solution, but still some integration work to do

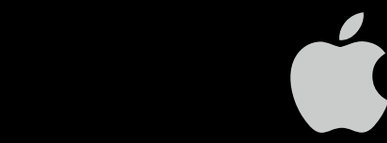


# Challenge: Authentication

- ACLs: Easier to implement than to use
- ACL interoperabilty - fact or fiction?
- The resource fork is back! POSIX Extended Attributes:
  - A challenge for the command line
  - A challenge for NFS and non-EA aware local File Systems

# Challenge: API Stability

- Telling people to just recompile their code is **NOT** an evolutionary API strategy:
  - APIs need to be clearly classified (supported, unsupported, unstable, marked for death, etc) in header namespace and doc
  - Shared library version numbers aren't proving to be sufficient
  - Current linker toolchain may not be sufficient either

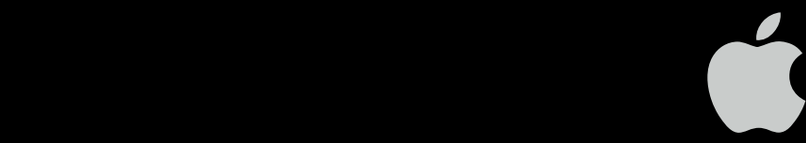


# Challenge: API Stability

- Restricted Kernel APIs are essential:
  - Developers like to poke into the innards, but this can strongly inhibit innovation
  - "Just recompile" not even often an option in this application space
  - Things like /dev/kmem are evil and should die (and will someday in Mac OS X)
- Proper kernel abstraction can help **both** the OS vendor and its 3rd party hackers

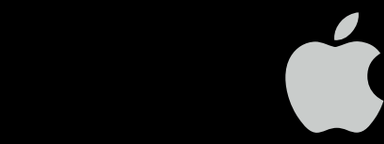
# Challenge: Administration

- Still too many weird configuration files and formats (~/Library/Preferences could be taken further)
- Service control and management is crude
- Remote administration and machine cluster administration still has a long way to go
- Logging / Auditing (for firefighting) are haphazard



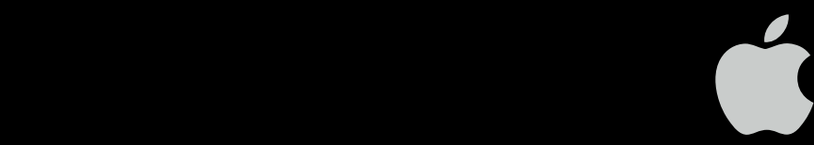
# Challenge: UI + Applications

- The X Window System still sucks as a UI portability solution
  - The X UI toolkit world is still balkanized
  - Complex desktop apps can't use it anyway
- High level APIs - Libc isn't, but it's the only common denominator we have (for now)
- "Scripting" languages (and Java) are one possible portability bridge



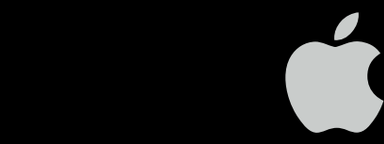
# Challenge: Hardware evolution

- Integer performance and clock rate increases are slowing down with die-shrinks and other issues
- Floating point performance is becoming a more significant battleground
  - Comparatively little compiler tuning and hand-optimization can have significant effects
  - Continuous work on exploiting Altivec in progress at Apple
- GPUs are also becoming viable as general purpose computational engines



# Challenge: OSS community

- Apple has done a great job leveraging open source, but there are things we want to improve:
  - More effective 2-way collaboration. Not just "pull" but "push"
  - Greater visibility into the OS dev process (particularly with bug reporting)
  - More timely source drops which always match current OS and update version
  - More "co-production" with OSS community, where and when it matches their mission



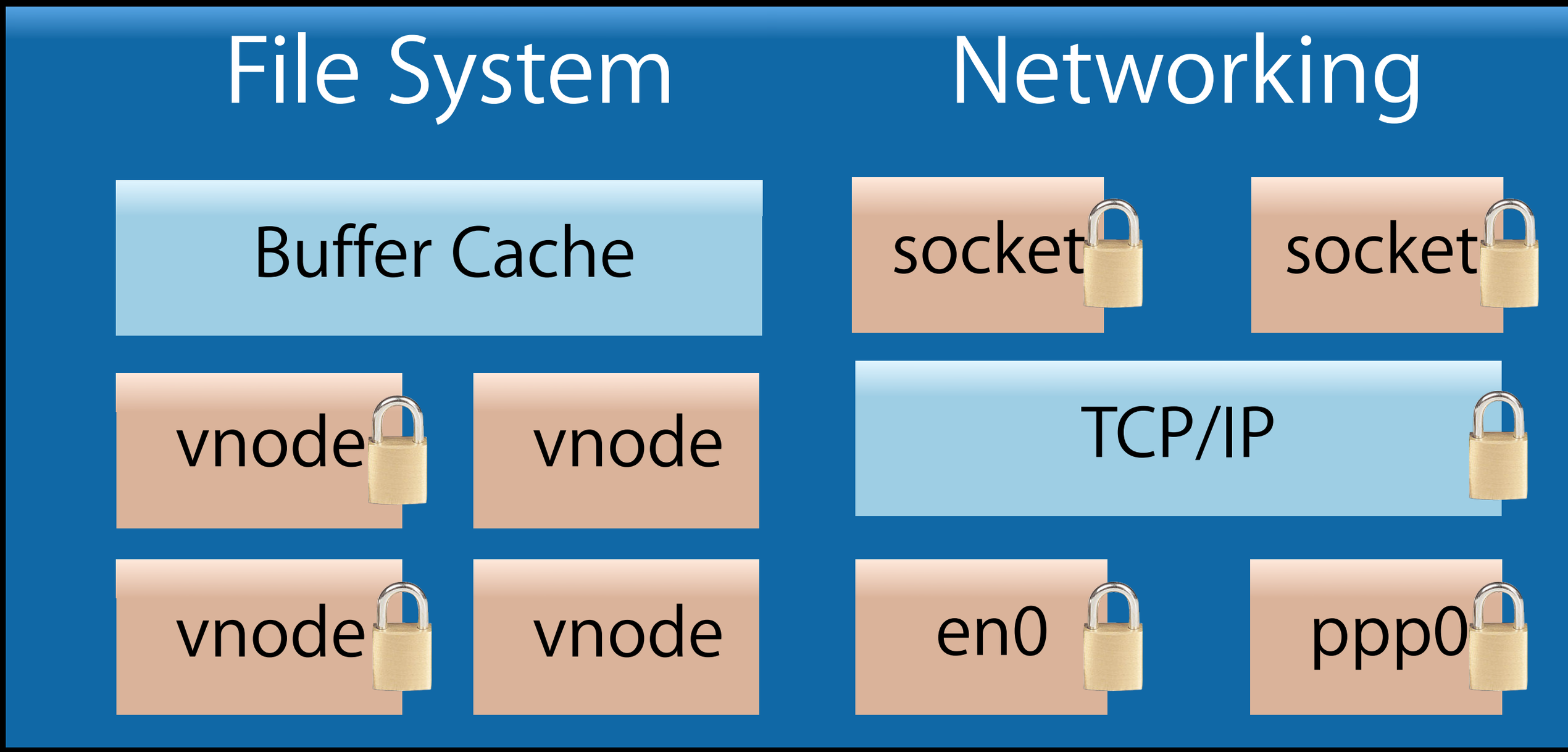
# Tiger - A selective preview

Some UNIX challenges we are addressing...

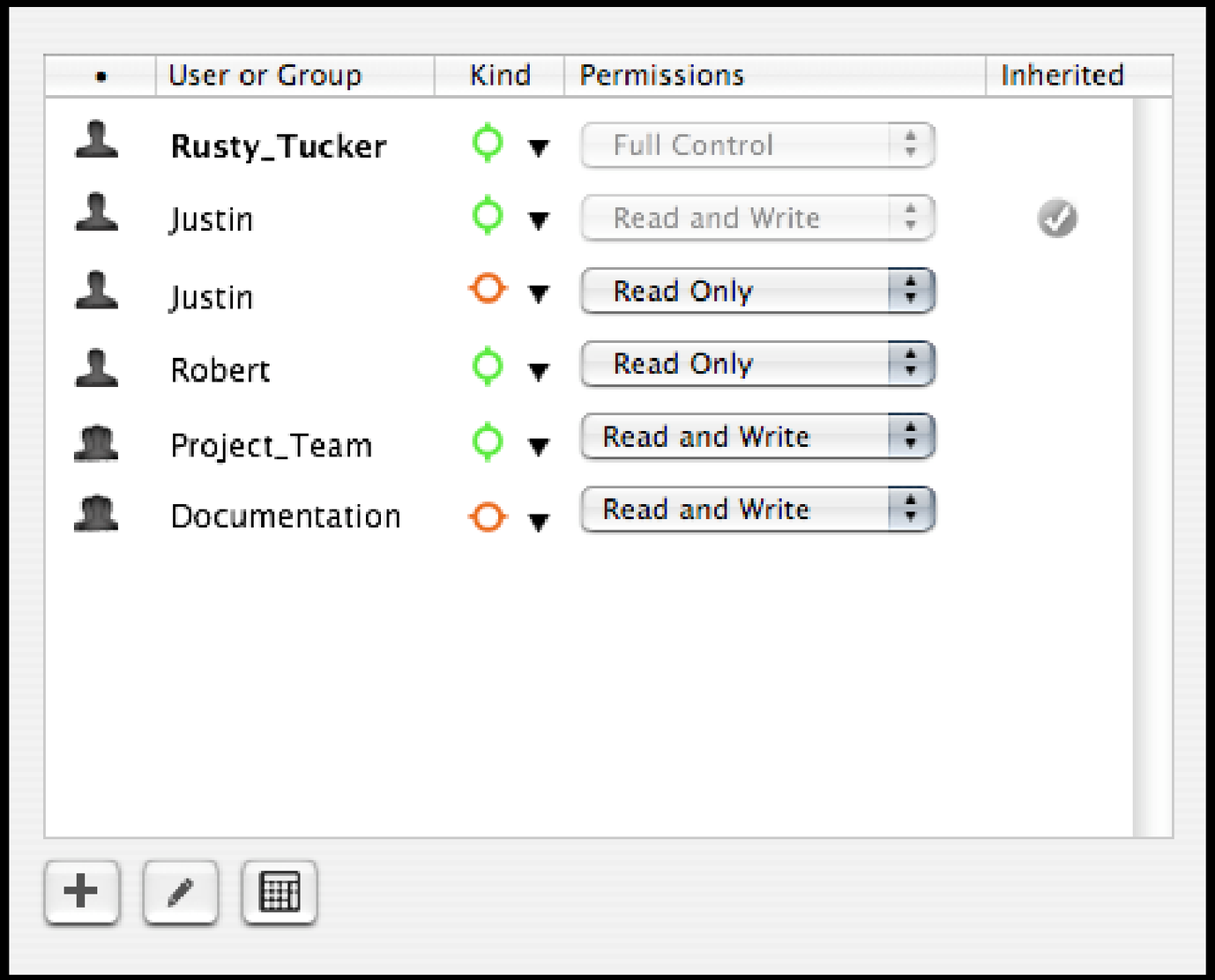
# 64-Bit Features

- 64-bit addressing for user tasks
  - Up to 16 exabytes of addressable virtual memory
- Designed for large data set applications
  - Scientific applications
  - Rendering engines
  - Server applications
- Improves performance for memory-intensive applications
- Xcode 2.0 supports 64-bit development

# Finer-grained Kernel Locking



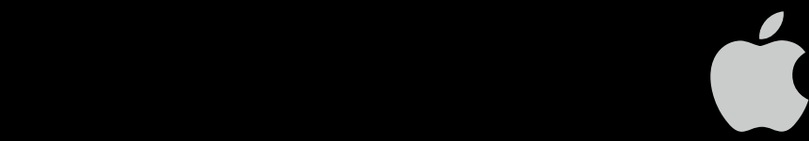
# Access Control Lists




- Conceptual ACL
  - List of Access Control Entries (ACEs)
  - Group or User
  - Permissions granted or denied
- Each ACL is bound to a file system object
  - File
  - Directory

# HFS+ Metadata (EA) Support

- Command line support
  - cp, mv, ditto
  - Remote copy engines: scp, rsync
  - Archivers: tar, zip, cpio
  - Editors: vim, emacs, pico
- No need for “enhanced” tools (rsyncX, tar_hfs, cpMac)
- Tiger ↔ Tiger should “just work”
- Investigating Tiger ↔ Non-Tiger



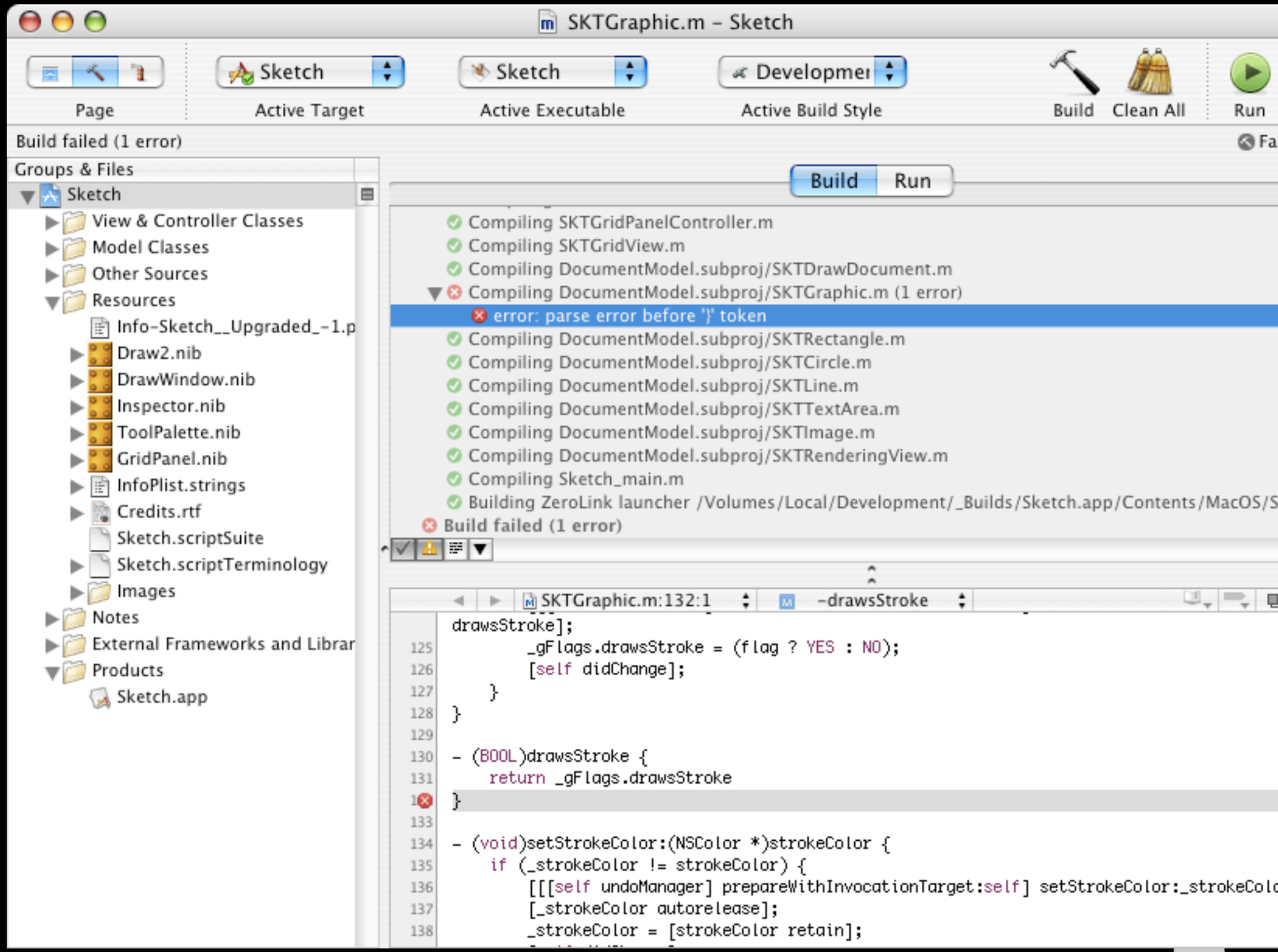
# Xgrid 1.0 in Tiger

- Distributed computing for the rest of us
  - An easy way to submit and run any number of computational tasks on an ad-hoc cluster of Macs
  - Xgrid handles the hard work of:
    - connecting nodes into a cluster
    - managing a queue of jobs and subtasks
    - Monitoring node availability
    - scheduling the tasks on the nodes
    - copying executables and input data to nodes
    - staging output data and collecting results
  - Security can be handled via ad-hoc mutual authentication or managed via Open Directory

# Two Ways to Use Xgrid in Tiger

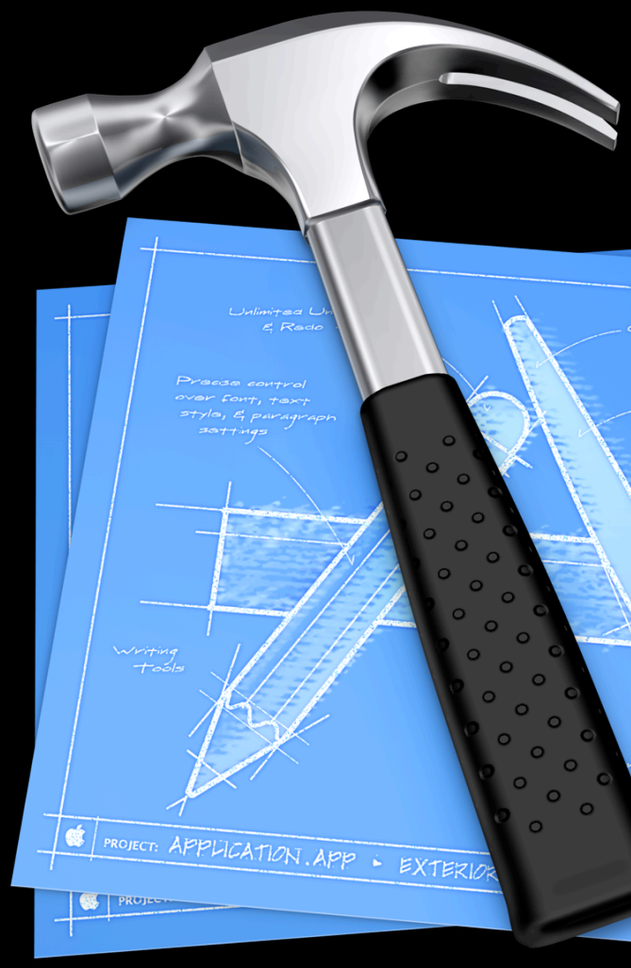
- Use the "xgrid" tool
  - Factor computational code into command-line executable
  - Use Xgrid to distribute work and collect results
  - (Recommended for current projects)
- Integrate with your application using Cocoa API
  - Distribute tasks if grid available
  - Monitor status of work
  - Retrieve results from Xgrid controller
- See http://www.apple.com/acg/xgrid/ for more info

# Xcode 2.0


SKTGraphic.m – Sketch
Page
Sketch
Active Target
Sketch
Active Executable
Developmer
Active Build Style
Build
Clean All
Run
Build failed (1 error)
Groups & Files
Sketch
View & Controller Classes
Model Classes
Other Sources
Resources
Info-Sketch__Upgraded_-1.p
Draw2.nib
DrawWindow.nib
Inspector.nib
ToolPalette.nib
GridPanel.nib
InfoPlist.strings
Credits.rtf
Sketch.scriptSuite
Sketch.scriptTerminology
Images
Notes
External Frameworks and Librar
Products
Sketch.app
Build
Run
Compiling SKTGridPanelController.m
Compiling SKTGridView.m
Compiling DocumentModel.subproj/SKTDrawDocument.m
Compiling DocumentModel.subproj/SKTGraphic.m (1 error)
error: parse error before '}' token
Compiling DocumentModel.subproj/SKTRectangle.m
Compiling DocumentModel.subproj/SKTCircle.m
Compiling DocumentModel.subproj/SKTLine.m
Compiling DocumentModel.subproj/SKTTextArea.m
Compiling DocumentModel.subproj/SKTImage.m
Compiling DocumentModel.subproj/SKTRenderingView.m
Compiling Sketch_main.m
Building ZeroLink launcher /Volumes/Local/Development/_Builds/Sketch.app/Contents/MacOS/S
Build failed (1 error)
SKTGraphic.m:132:1
-drawsStroke
drawsStroke];
_gFlags.drawsStroke = (flag ? YES : NO);
[self didChange];
}
}
- (BOOL)drawsStroke {
return _gFlags.drawsStroke
}
- (void)setStrokeColor:(NSColor *)strokeColor {
if (_strokeColor != strokeColor) {
[[[self undoManager] prepareWithInvocationTarget:self] setStrokeColor:_strokeColo
[_strokeColor autorelease];
_strokeColor = [strokeColor retain];

# Xcode

- The fastest way to create Mac OS X applications
  - GCC 4.0
  - Optimized for Power Mac G5
  - High performance development technologies
    - Fix and continue
    - Predictive compile
    - Zero Link
    - Distributed build
  - Shark and CHUD performance tools
    - G4 and G5 optimizers with every system
    - Included at no additional cost



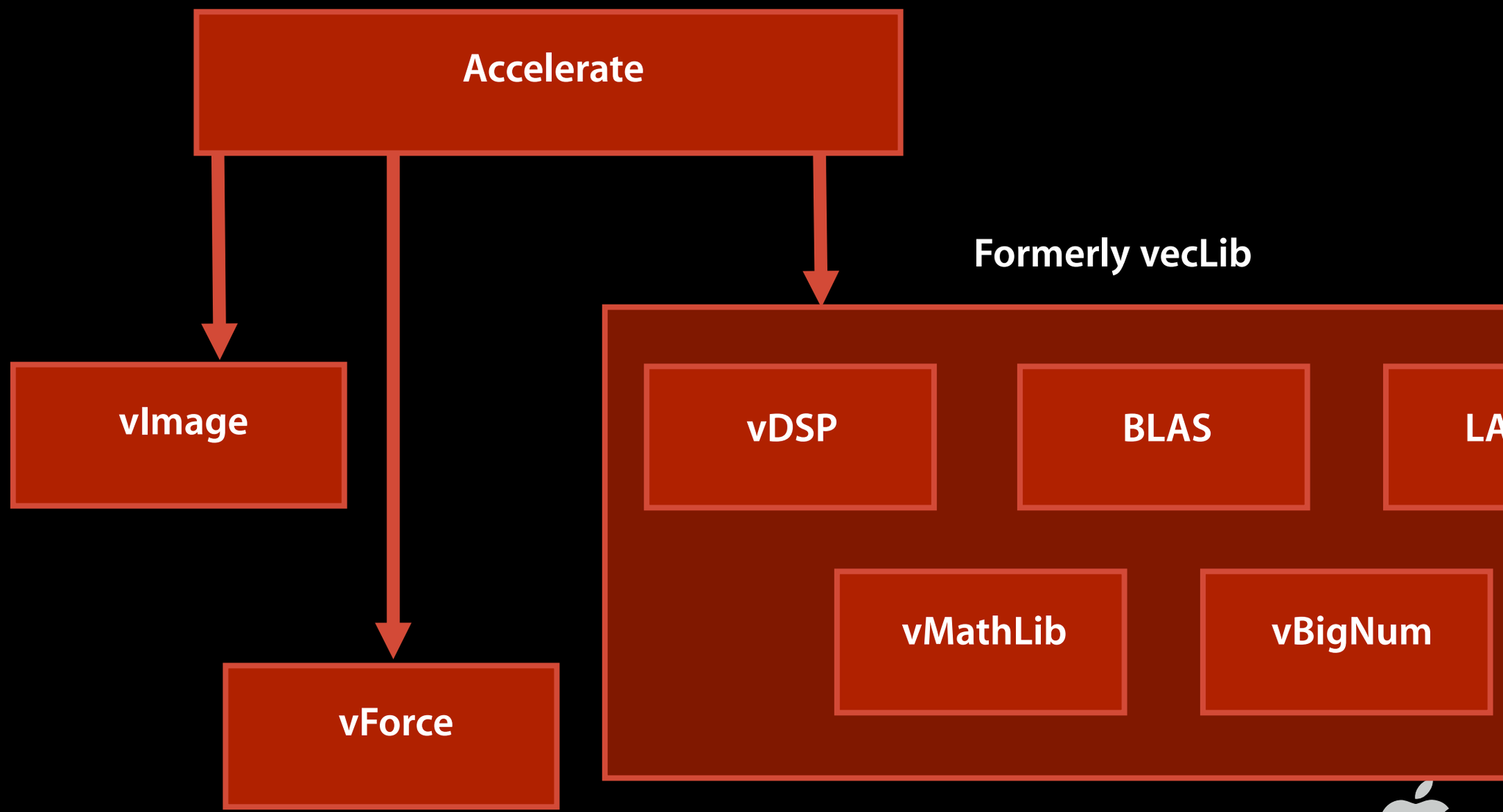
# Tiger Performance Math APIs

- At every level, usability is straightforward
  - Library APIs internally dispatch for G3 vs. G4, G5,
  - One binary safely runs on all platforms
  - libm links by default (just like libc)
- For “long double” and “complex” APIs:
  - libmx.a (”-l mx”)
- For vForce, BLAS, LAPACK, vDSP, vImage:
  - “-framework Accelerate”

# G5-Tuned Libm

- Leverages new features of G5 processor
  - Algorithms recast at instruction level to exploit 2 FPUs
    - Careful attention paid to dispatch group formation
    - Careful attention paid to Load/Store hazards
  - Hardware square root
  - Faster on G4 too!
  - New libmx.a for complex double and long double math

# The Accelerate Framework in Tiger



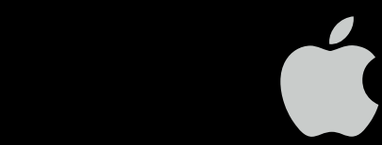
# Vector Libraries

- Robust library for low-effort performance enhancement
  - Digital signal processing: 1-D, 2-D FFTs [vDSP]
  - BLAS Levels 1, 2, 3 (ATLAS tuned, selectively SMP aware)
  - LAPACK, linear systems and eigenvalue problems
  - Tuned 4x4, 8x8, 16x16, 32x32 matrix multiplies
  - Heavy use of Velocity Engine throughout for single precision
  - Using from C

```
#include < Accelerate/Accelerate.h>
cc someMath.c -O2 -framework Accelerate
```

# Launchd

- Merges the functionality of init, mach_init, xinetd, cron and System Starter
- Understands legacy configuration files (via translating parsers) in addition to new plist configuration files
- Much more flexible rules for determining when and why to launch a service
- Provides a single interface for registering, starting, stopping and interrogating services
- Takes almost all the hard work out of writing a network or Mach IPC based service



# ASL - The Apple System Logger

- Fully backwards compatible with syslog
- Supports arbitrary output plug-ins for storing log data as well as client and server side filters
- Unified log message format and encoding
- Command-line tool for controlling logging behavior, searching and pruning log messages
- Both client and server-side log threshold control
- Still evolving: Will eventually consolidate all log data produced on Mac OS X

# UNIX command/library updates

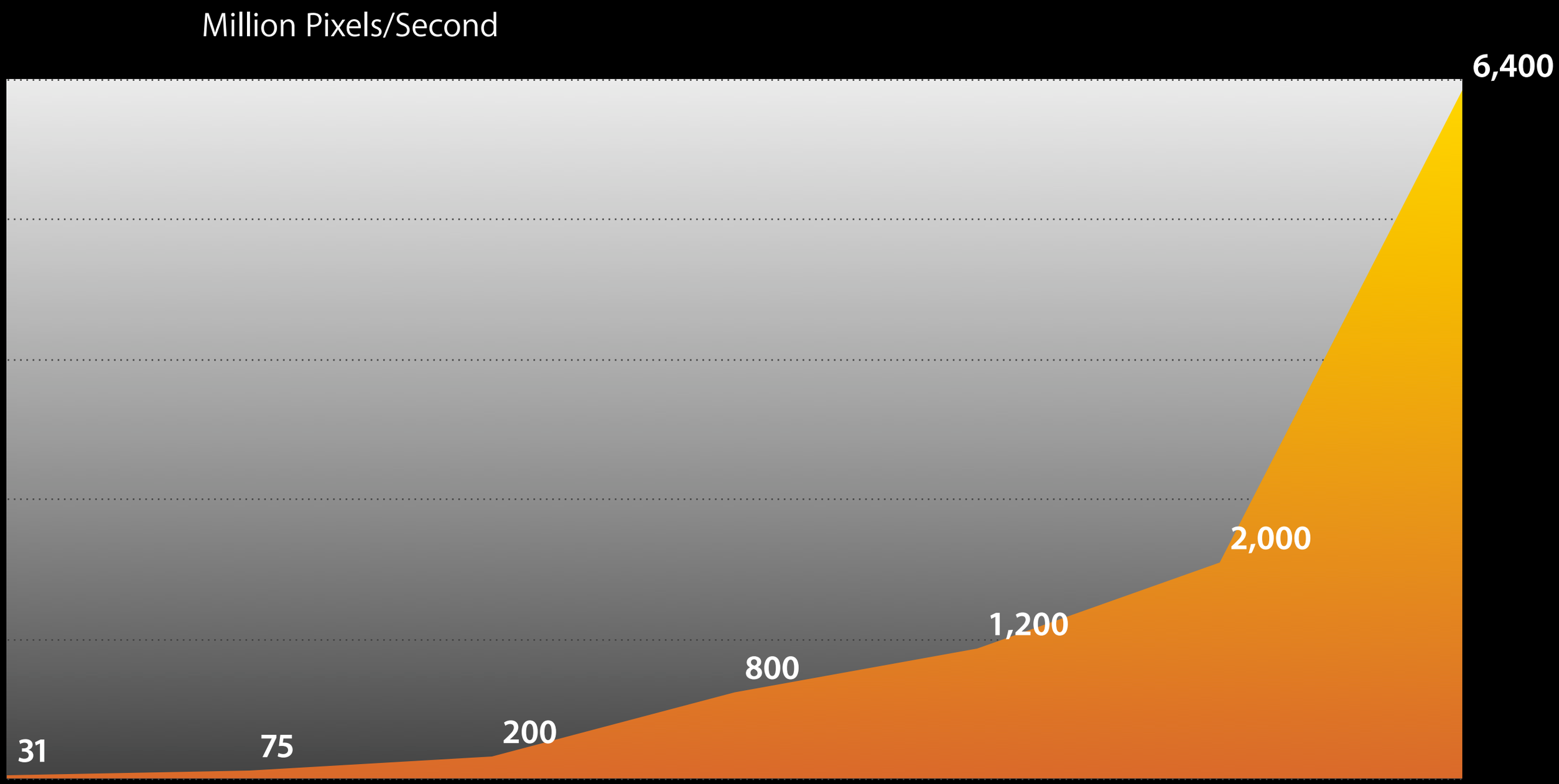
- Commands and libraries updated to FreeBSD 5.x
- Perl, python and ruby all updated to latest versions
- Tcl updated and Aqua Tk added for portable UI programming
- Tkinter and wxWidgets added for portable Python UI
- Looking at UI portability solution for PERL too
- The AT&T Korn shell is now bundled - Solaris users will find this useful

# UNIX/Linux compatibility

- dlopen() and friends are now native (and preferred API for dealing with dynamic loading)
- poll() is now native
- kqueue() support much improved
- Even more SYSV compatibility (ipcs, iprm, et al)
- Improved pthread support
- MUCH more compatibility with the UNIX03 spec in headers and libraries

# Core Image

# Graphics Processing Power




Source: NVIDIA

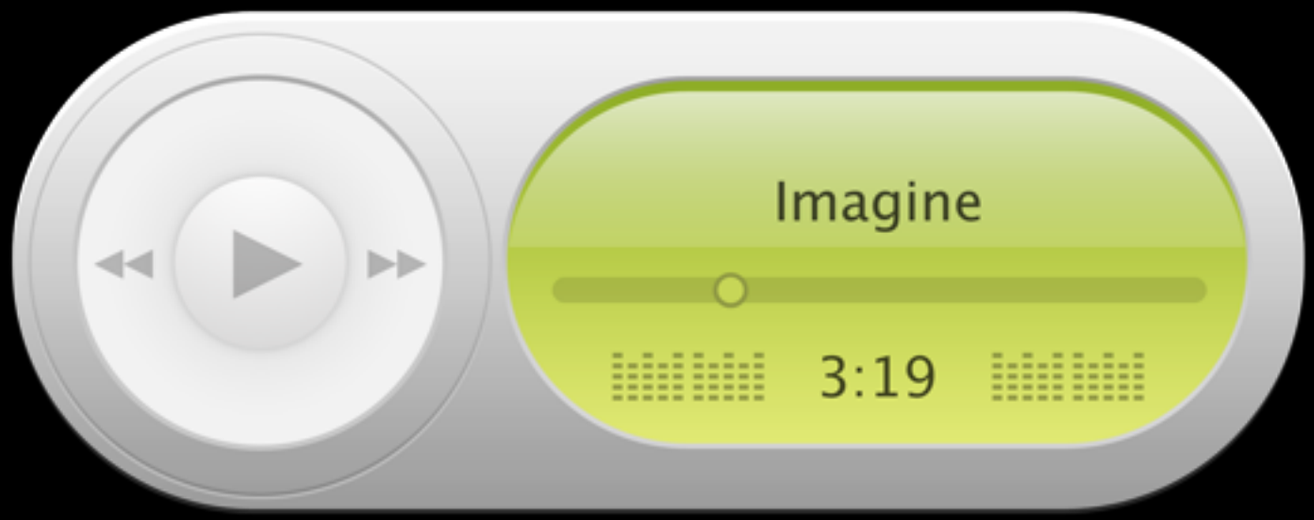
# Core Image

- Hardware-accelerated real-time image processing and rendering
- Per-pixel programming
- Floating-point precision
- Support for Core Video
- Effects and transitions
- Image Units

# Included Filters

**Focus Filters** • Gaussian Blur • Motion Blur • Zoom Blur • Unsharp Mark **Col**
**Filters** • Color • Controls • Color Matrix • Exposure Adjust • Gamma Adjust • Hue
Point Adjust **Color Filters** • Color Invert • Color Monochrome • Color Posterize
Sepia Tone **Compositing Filters** • Addition • Maximum • Minimum • Multiply
Source In • Source Out • Source Over **Distortion Filters** • Bump Distortion
Distortion • Glass Distortion • Glass Lozenge • Torus Lens Distortion • Twirl Dist
Distortion **Generator Filters** • Checkerboard • Constant Color • Lenticular Halo
Stripes • Sunbeams **Geometry Filters** • Affine Transform • Crop • Perspec
**Gradient Filters** • Gaussian Gradient • Linear Gradient • Radial Gradient **Halftone**
Screen • Dot Screen • Hatched Screen • Line Screen **Stylish Filters** • Blo
GloomPixellate • Spot Light **Tile Filters** • Affine Tile • Op Tile • Parallelogram Tile •
• Triangle Tile • Triangle Tile **Transition Filters** • Copy Machine • Dissolve • Flash

# Dashboard



Mail File Edit View Mailbox Message Format Window Help

Weather Movies Dictionary Buddy List News Sports Stickies World Clock Yellow Pages

SAN FRANCISCO


| | | |
|---|---|---|
| NASDAQ | 3450.05 | + 4.50 |
| NYSE | 14535.44 | + 3.98 |
| AAPL | 42.05 | + 7.36 |
| MSFT | 18.23 | - 1.34 |
| PIXR | 92.23 | + 3.32 |
| AMZN | 5.23 | + 0.12 |

1d 3m 6m 1y 2y 3y

28 26 24 26

Mar Apr May Jun Jul Aug

edit

reminders:

don't forget to

flowers for mo

Fri. OCT

| S | M | T | W |
|---|---|---|---|
| | | 1 | 2 |
| 6 | 7 | 8 | 9 |
| 13 | 14 | 15 | 16 |
| 20 | 21 | 22 | 23 |
| 26 | 27 | 28 | 29 |

Appointments:

1:45PM
Lunch with Win

6:30PM
Dinner with Kat

5973

m+ m- mc mr ÷

7 8 9 ×

4 5 6 −

1 2 3 +

0 . c =

**Apple Computer Inc.**

main 1-800-MY-APPLE

work 1 Infinite Loop
Cupertino CA 95014
United States

10 / 127 Name

Shazam!

0:36

# Dashboard Widgets

- Exposé-like access
- Web widgets
- Accessory widgets
- Widgets built in
- Easy to build with Webkit



# Automator

# Automator

- Automation of repetitive or complex tasks
- No programming required
- Interactive or fully automated
- 100+ Actions for Finder, iLife, Mail, Address Book, iCal, and more
- Developers can add actions
- Reusable automations
- Leverages the power of Mac OS X technologies

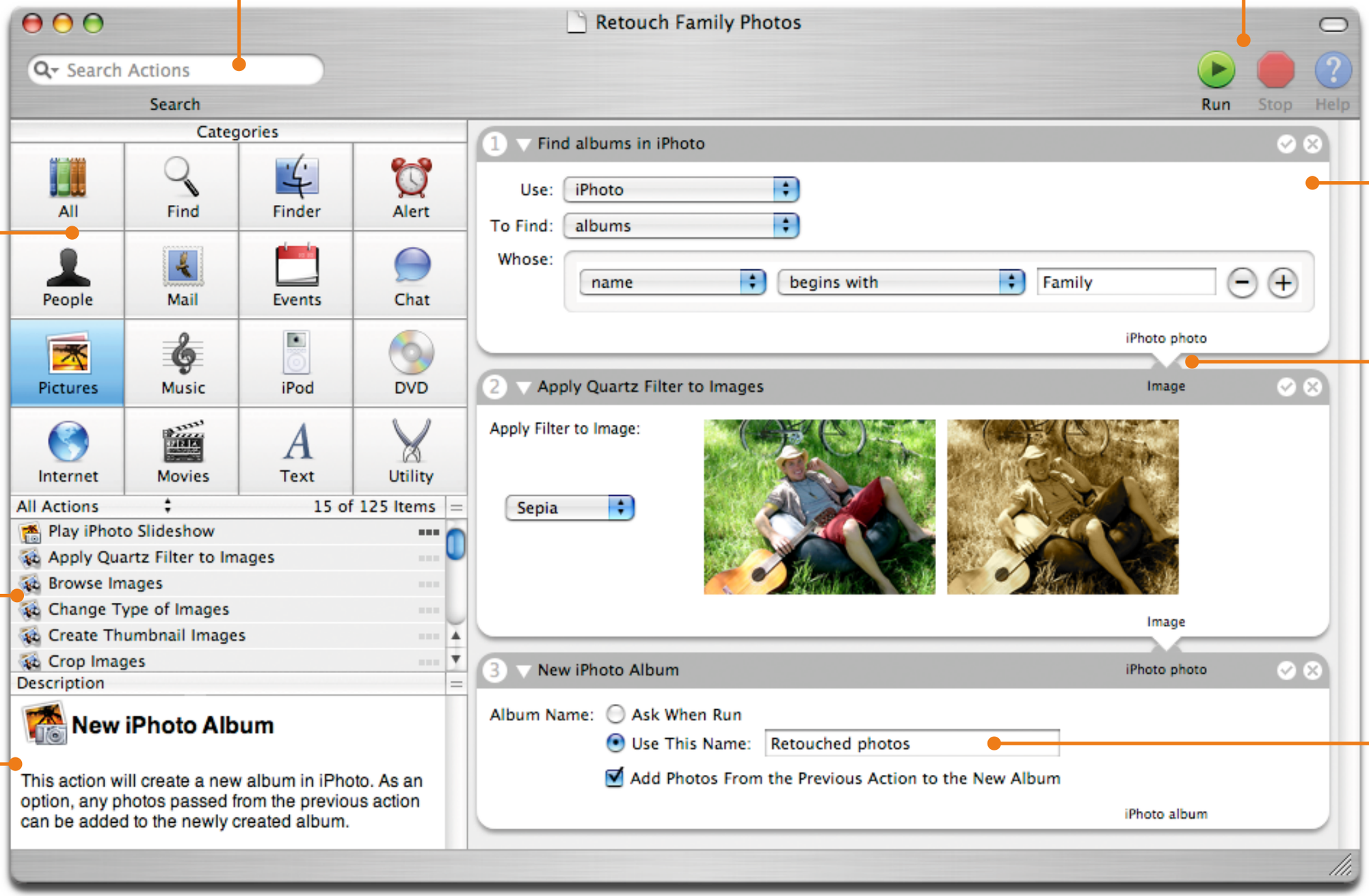
Search for actions

Control your program

125 actions organized by category

Available actions

Selected action description



# First Half of 2005

# Future Challenges

- Package management and ports collection
- Sandboxing things for security
- Unified system administration interfaces
- Make things more friendly to clustering
- Continue to increase performance (both overall and for specific applications)
- Balance the needs of the desktop and the enterprise

# Q&A