SciTech SNAP
System Neutral Access Protocol

The Future of Device Driver Technology!

Kendall Bennett

CEO & Founder

SciTech Software, Inc

Introduction & Agenda

- The “Driver Crisis”
  - Recognizing the Driver Crisis
  - The Driver Crisis worsens over time
  - Root causes of the driver crisis
- The SciTech SNAP Architecture
  - Architectural Overview
  - Key Benefits
  - Beneficiaries

Recognizing the “Driver Crisis”

- What is a device driver?
- Ideally every user would have a driver for every OS and every device
- The “Driver Crisis” is the dilemma faced by the industry due to lack of sufficient drivers
- This affects functionality, stability and performance of end user computer systems

Open Architectures Precipitated the Driver Crisis

- Vendors tried to avoid this by building closed systems (ie: Apple)
- PC architecture is open allowing multiple operating system and hardware vendors to compete in the same space
- Prices range from sub $200 to well over $20,000
- Such as vast price range means there will be many different devices installed

Operating System Vendor Crisis

- Wide selection of hardware generates a crisis for the operating system vendor
- Not cost effective for OS vendor to build drivers for all devices
- Often unsuccessful persuading hardware vendors to support their OS
- Generally affects smaller, emerging OS’es rather than larger, well established OS’es.

Hardware Vendor Crisis

- Need to support multiple OS'es presents a crisis to the hardware vendor
- Intense competition and slim profit margins makes them focus on the biggest cash cow
- Focus on performance not compatibility (benchmarks sell hardware)
- Primarily in the business the sell hardware
- Device drivers are a necessary evil to capture the initial sale

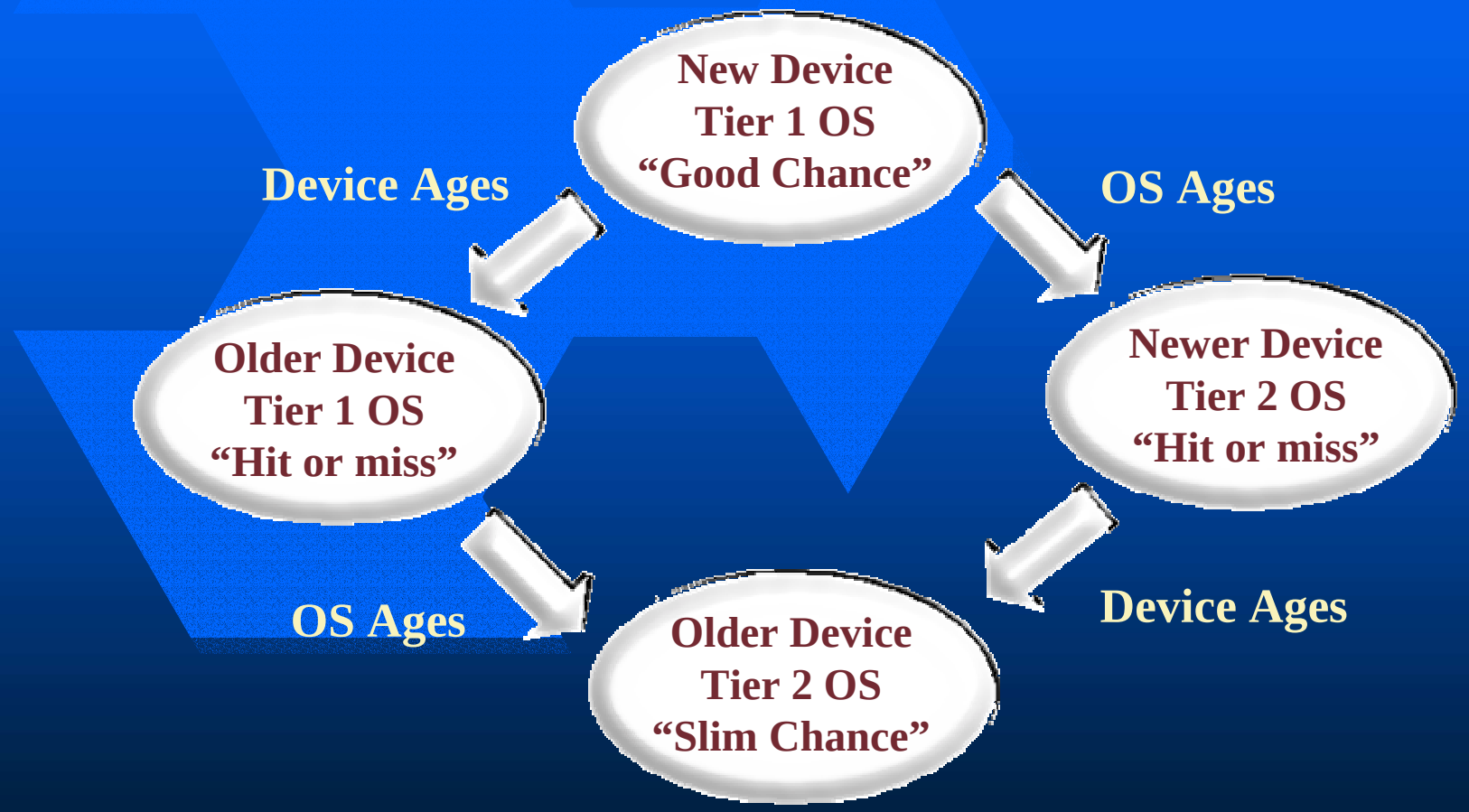
End User Crisis

- Final result is a lack of device drivers for the end user
- No choice but to complain to the hardware vendor or OS vendor
- Many times end users are informed to either upgrade their OS or replace their hardware

Driver Crisis Worsens over Time

- OS support differs depending on popularity and life cycle stage
- Over time an OS that once has good support ages to the point where support is less than adequate
- Only Tier 1 OS’es have good device support
- Presently Windows Me, Windows 2000 and Windows XP are the only Tier 1 OS’es
- Even Windows 95 and Windows 98 are Tier 2 today!

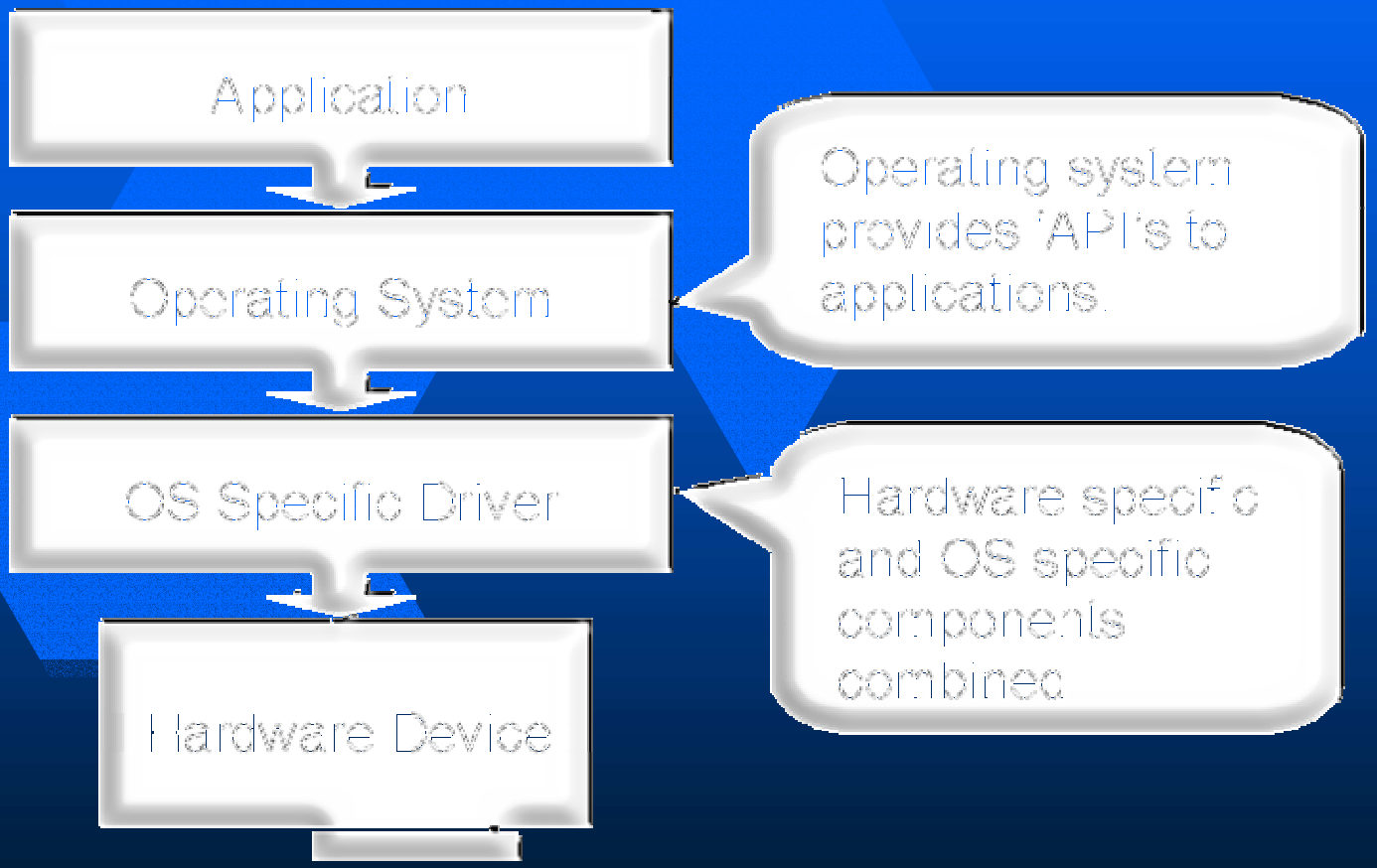
Driver Aging Over Time



Root Causes of the Driver Crisis

- Overview of Traditional device driver model
- Non-unified source code
- Code Rot
- Large Test Matrices

Overview of Traditional Device Driver Model



Non Unified Source Code

- Source code for each OS is different, even though functionality is identical
- Code is copied and ‘ported’ to a new OS from an existing OS or developed from scratch
- Bugs fixed for one OS are not necessarily fixed for another OS
- Performance improvements for one OS do not benefit other OS’es
- As the number of OS’es increases, maintaining support for all but the most popular OS’es becomes unmanageable

Code Rot

- Every time source code is compiled, it MUST be tested before release
- Lack of testing allows code to rot and cease to function in new releases
- Code Rot occurs through changes to interfaces to surrounding code (platform headers, DDK changes etc)
- Code Rot occurs due to changes in shared source code, especially in ‘Unified Drivers’

Code Rot (continued)

- Programmers working on new devices can cause Code Rot in support for old devices
- Much greater chance of code rot when support for many devices in shared in a single module
- Code Rot can occur due to changes in compilers and tools used to build drivers
- Bug in new tools can cause Code Rot
- Bugs fixed in new tools can cause Code Rot

Code Rot (continued)

- The only way to eliminate Code Rot is to completely test the resulting binary modules for correct operation EVERY time that the module is recompiled for release
- Such testing requirements leads to huge Test Matrices!

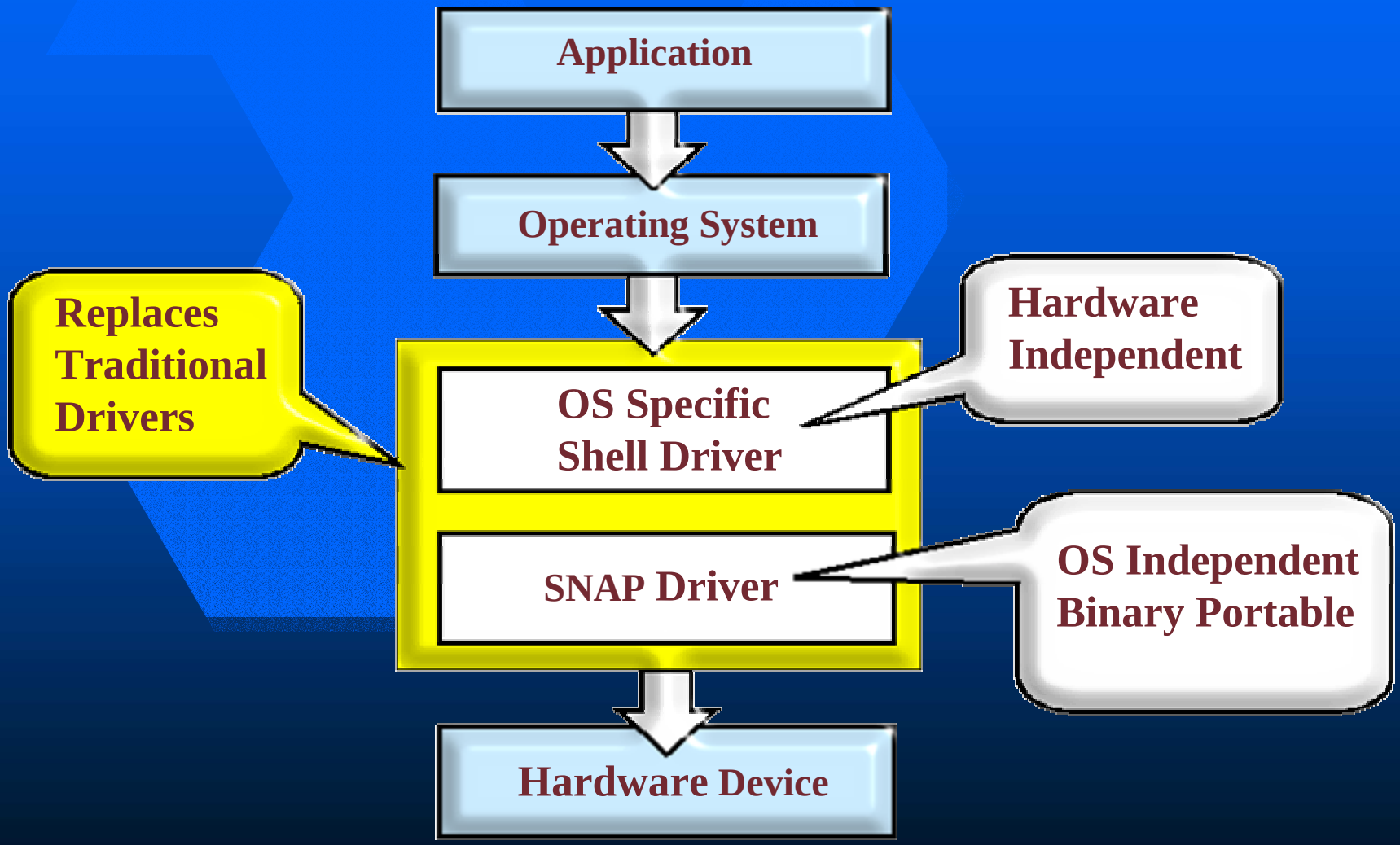
Large Test Matrices

- Required to avoid Code Rot
- Proper testing and verification requires time consuming and complex testing procedures
- Vendors neglect older hardware in order to avoid Code Rot
- With ‘Unified Drivers’ vendors either do not release support for all chipsets, or quality suffers through lack of testing

The SciTech SNAP Architecture

- SciTech SNAP Device Drivers Have Two Main Parts:
  - The Operating System Device Class
  - The Device Dependant Element
- Traditional Device Drivers Have One
  - Two driver components are combined to form a single, logical device driver.

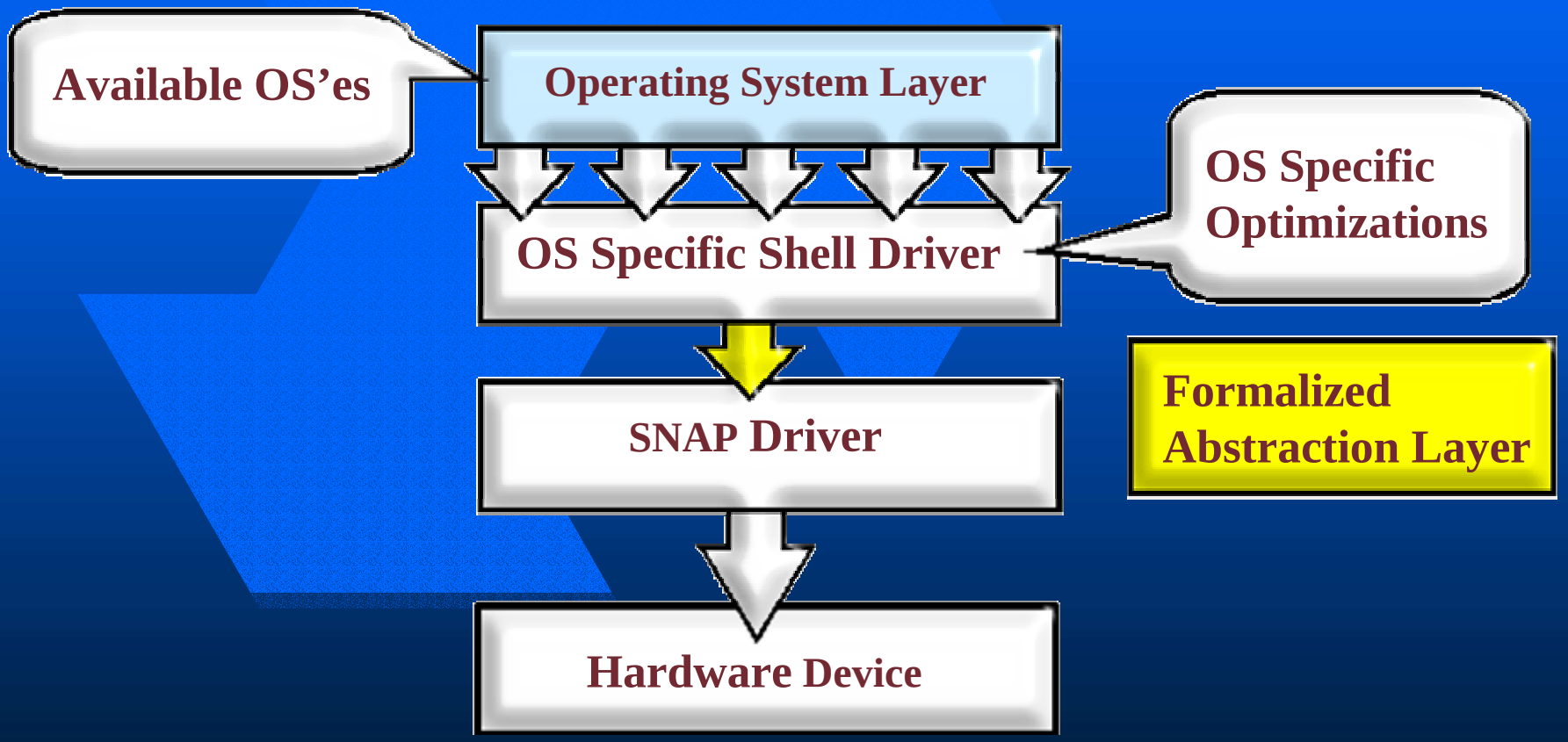
The SciTech SNAP Architecture



SciTech SNAP Binary Portable Drivers

- SciTech SNAP Binary Portable Driver
  - Can be utilized on any OS within a single processor family i.e. x86
- The Benefit Of Binary Portable Drivers
  - Allows reuse of tight highly optimized code
  - Enhancements are easily deployed on multiple OS’es

SciTech SNAP Multi OS Support



Processor Independent Code

- Developed Using The Highly Portable C Programming Language
- Porting To A New Processor Is Fast And Easy
- Easily Built And Tested Using A Single Development Environment.

Key Benefits

- Maximum Performance
- Rapid Device Driver Development
- Reduced Costs
- Cost Effective Device Support For New OS’es

Maximum Performance

- SciTech SNAP Delivers
  - Real world benchmarks have shown that SciTech SNAP drivers perform as good or better than drivers hard coded to a particular operating system and hardware device
- Optimizations Developed Once
  - The optimizations can be used by the widest variety of device/operating system combinations and there are more opportunities for spending additional time on further optimizations

Reduce Development Costs

- Traditional Development Methods
  - *1 device x 7 operating systems x 2 months = 14 man-months*
- With SciTech SNAP
  - *1 device x 1 month + 7 operating systems x 1 month = 8 man-months*

Rapid Support for New OS'es

- As OS Technologies Advance, New Device Drivers Must Be Developed
- With SciTech SNAP Adding Device Support For A New OS Is Easy
- Instantly Leverage Existing SciTech SNAP Device Support Libraries.

Who Benefits

- Every One Benefits
  - End Users
  - IS Managers
  - Software Developers
  - Hardware Vendors
  - Processor Vendors
  - Operating System Vendors

SciTech SNAP SDK/DDK release

- Source code to SciTech SNAP SDK and DDK will soon be released
- Licensed under dual GPL/proprietary license
- Full source code to OS neutral components
- Source code for a few sample chipset drivers

Questions and Answers

- Any Questions?