



UEFI Driver Development Training Application Development

Kevin Li
UEFI Development
Intel

Agenda

- Introduction
- EFI Shell
- EFI Toolkit
- 3rd Party Libraries



What are EFI Applications?

- EFI Applications extend firmware
 - No hardware dependence
 - No OS dependence
- Portable across platforms
 - IA-32, Intel® 64, IA-64
- Enables rapid application development



What is an EFI Application?

- An EFI Loadable Image
 - Loaded by EFI loader just like drivers
 - Does not register protocols like drivers do
 - Consumes protocols
 - Typically user driven (exits when task completed)
 - Same set of interfaces available as drivers have
- Can be used for
 - Platform diagnostics
 - Factory diagnostics
 - Utilities
 - Driver prototyping
 - 'Platform' applications



EFI Shell

- An EFI Application
- Interactive Console Interface
- Application Launch
- Load EFI Drivers
- Scripting Capability
- Automatic execution of startup script file
- Console redirection to files

Open Source located on <http://efi-shell.tianocore.org> project



EFI Toolkit Components

- Utilities
- C Library
- Network Stack
- Platform Management
- Compression
- Database

Source Included

**Useful tools for EFI
application development**

Open Source located on <http://efi-toolkit.tianocore.org> project

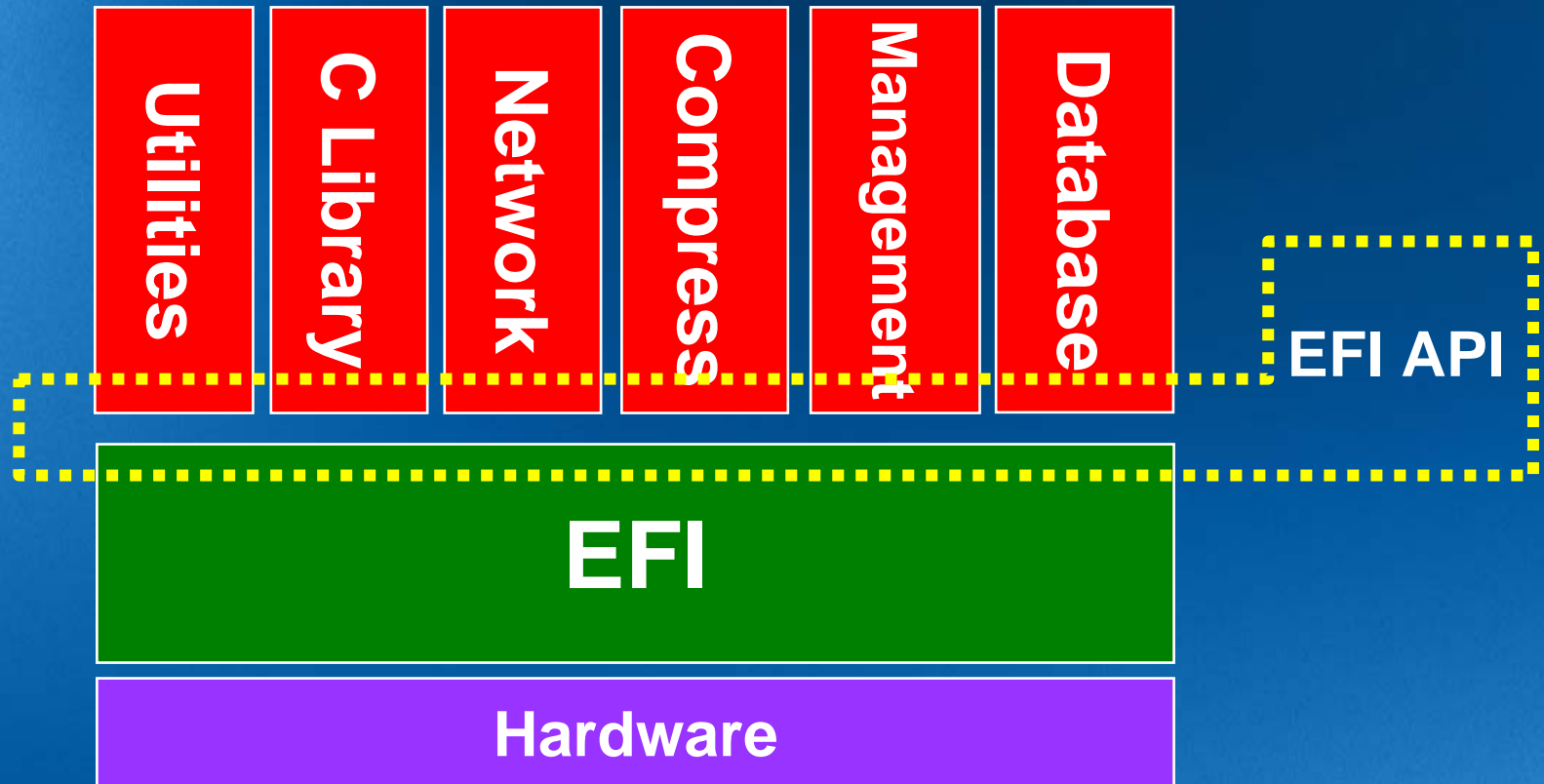


Programming Models

- Native EFI Model
 - Uses only EFI constructs
 - Access to all EFI constructs
 - Smaller code size
- Portability Model
 - Familiar programming interfaces
 - Easier to port ANSI/POSIX based programs
 - Larger binary image
- A single program can use both



EFI Toolkit Integration



C Library

- FreeBSD Port
- ANSI/POSIX compliant
- System I/O - `open()`, `read()`, `write()`, `close()`, `stat()`
- Standard I/O - `fopen()`, `printf()`, `gets()`, ...
- String/Char - `strcmp()`, `isascii()`, `atoi()`, ...
- Memory - `malloc()`, `free()`, `realloc()`, ...
- Time/Date - `time()`, `asctime()`, `ctime()`, ...
- Math - `sqrt()`, `pow()`, `sin()`, `log()`, ...



EFI Library

- “Lite Weight” C Library like functions
 - String Functions
 - Memory Support Functions
 - CRC Support Functions
 - Text I/O Functions
 - Math Functions
 - Spin Lock Functions
- Specific EFI functions
 - Handle and Protocol Support Functions
 - Device Path Support Functions



Network Components

- Port of FreeBSD TCP/IP stack
- Supports standard protocols
 - IPv4, ICMP, ARP, UDP, TCP
- Socket library interface
- Implemented as an EFI protocol



Miscellaneous

- SMBIOS Library
 - Library routines for parsing SMBIOS tables
- Database
 - btree
 - Hashing
- Compression
 - General purpose compression/decompression
 - Gzip functionality



Utilities

- Network utilities
 - FTP client and server, ping
- Text editor
- Scripting interpreter (Python)
- Sample applications



EFI Hello.c

```
#include "efi.h"

EFI_STATUS
InitializeHelloApplication (
    IN EFI_HANDLE      ImageHandle,
    IN EFI_SYSTEM_TABLE *SystemTable
)
{
    UINTN Index;

    SystemTable->ConOut->OutputString(SystemTable->ConOut,
                                      L"Hello application started\n");
    SystemTable->ConOut->OutputString(SystemTable->ConOut,
                                      L"\n\r\n\r\n\rHit any key to exit this
image\n\r");
    SystemTable->BootServices->WaitForEvent(
        1, &(SystemTable->ConIn->WaitForKey), &Index);
    SystemTable->ConOut->OutputString(SystemTable->ConOut,
                                      L"\n\r\n\r");
    return EFI_SUCCESS;
}
```



EFI Library Hello.c

```
#include "efi.h"
#include "efilib.h"

EFI_STATUS
InitializeHelloLibApplication (
    IN EFI_HANDLE          ImageHandle,
    IN EFI_SYSTEM_TABLE    *SystemTable
)
{
    InitializeLib (ImageHandle, SystemTable);
    Print(L"\n\n\nHelloLib application started\n\n\n");
    Print(L"\nHit any key to exit this image\n");
    WaitForSingleEvent(ST->ConIn->WaitForKey, 0);
    ST->ConOut->OutputString (ST->ConOut, L"\n\r\n\r");
    return EFI_SUCCESS;
}
```



C Library Hello.c

```
#include <atk_libc.h>
#include <stdio.h>

EFI_STATUS
InitializeHelloLibCApplication (
    IN EFI_HANDLE          ImageHandle,
    IN EFI_SYSTEM_TABLE    *SystemTable
)
{
    InitializeLib(ImageHandle, SystemTable);
    printf("Hello LibC application started\n\n\n");
    printf("Hit C/R to exit this image\n");
    return( getchar() );
}
```



C Library Hello.c

```
#include <atk_libc.h>
#include <stdio.h>

int main (int argc, char **argv )
{
    printf("Hello LibC application started\n\n\n");
    printf("Hit C/R to exit this image\n");
    return( getchar() );
}
```



C++ Support

- No direct support
 - No Global constructors and destructors
- New and Delete can be mapped to malloc/free



Portable Embedded Graphics

- Portable Embedded Graphics
 - Portable graphics library for EFI
 - Similar windowing components (widgets)
 - Dialog boxes
 - Progress bars, scroll bars
 - Text boxes
 - Window Management
 - Fonts
 - Bitmaps, JPEG, ...
- Contact Swell Software
 - <http://www.swellsoftware.com>



PEG Components

Button Dialog Window

Normal Disabled **Big Font**

☒ Moveable ☐ Fixed

☒ Enable Group

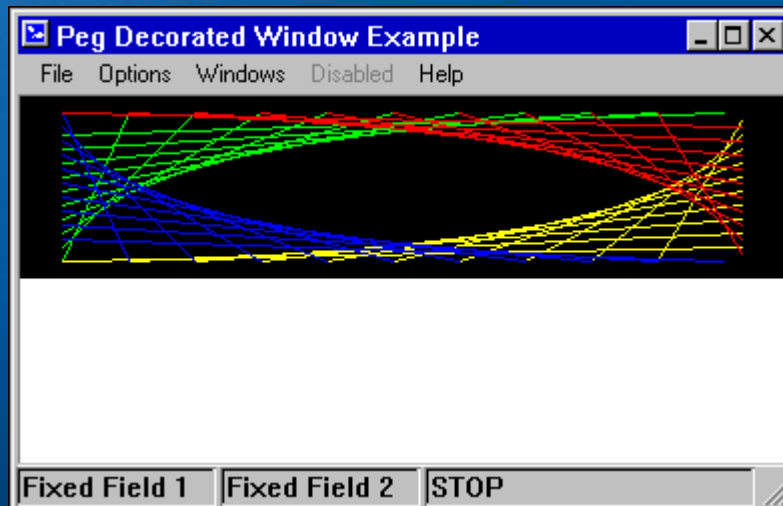
PegGroup

☒ Radio 1 ☐ Radio 2 ☐ Radio 3 ☐ CheckBox

Left Center Right

Multiple Line

OK



Slider/ProgressBar Example

25%

75

20%

60

66

20%

60

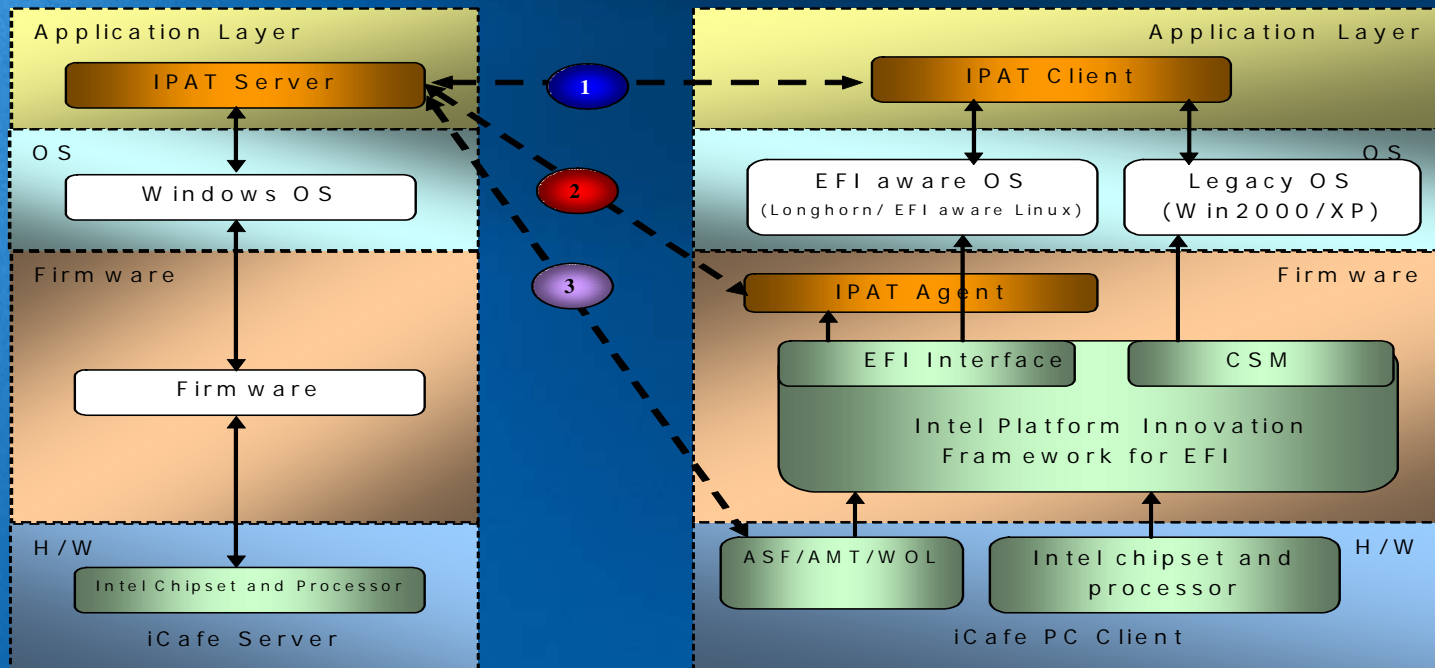
IPAT Solution

- A EFI based remote manageability solution
 - Remote OS image provision (multicast)
 - Remote control
 - Hard disk content protection and recovery
 - Asset management

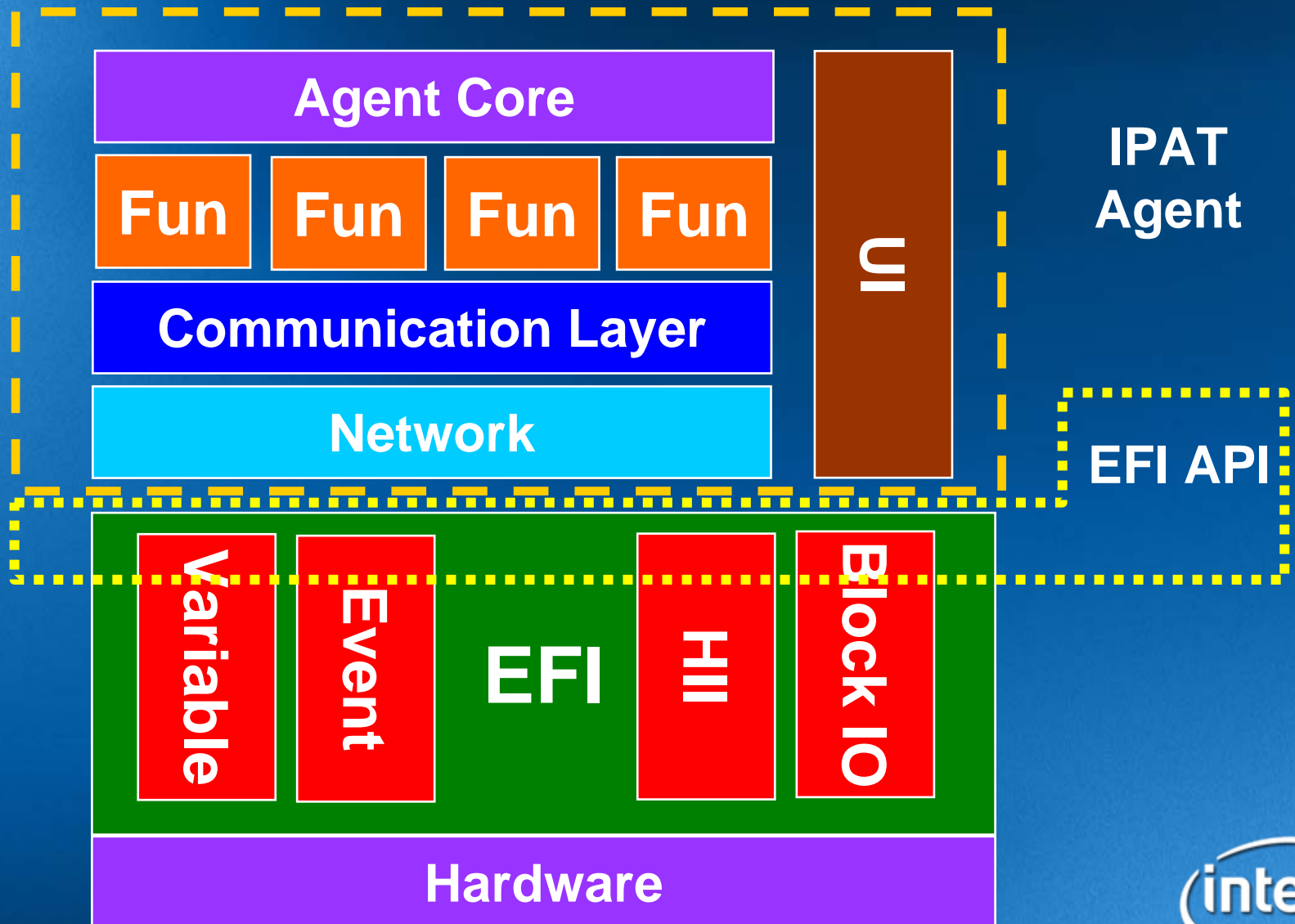
<http://platformadmintech.intel.com/>



IPAT Architecture



IPAT Agent



Summary

- EFI Applications extend firmware
 - Provides system independence in the pre-boot space
 - Hardware
 - Operating System
 - Platform
 - IA-32, Intel® 64, and IA-64
- Large library support
- EFI Shell provides convenient launch point



Further Information

- <https://efi-toolkit.tianocore.org/>
 - EFI Web site for EFI Toolkit download
- <https://www.TianoCore.org>
 - Website for EFI open source resources
 - [EFI Developer Kit](#) (EDK)
 - Nt32 emulation environment
 - [EFI Shell Source](#)
- <http://www.swellsoftware.com>
 - Portable Embedded Graphics toolkit
- <http://uefi.org>
 - UEFI Specification



