tianocore

UEFI & EDK II Base Training

UEFI Network in EDK II

tianocore.org

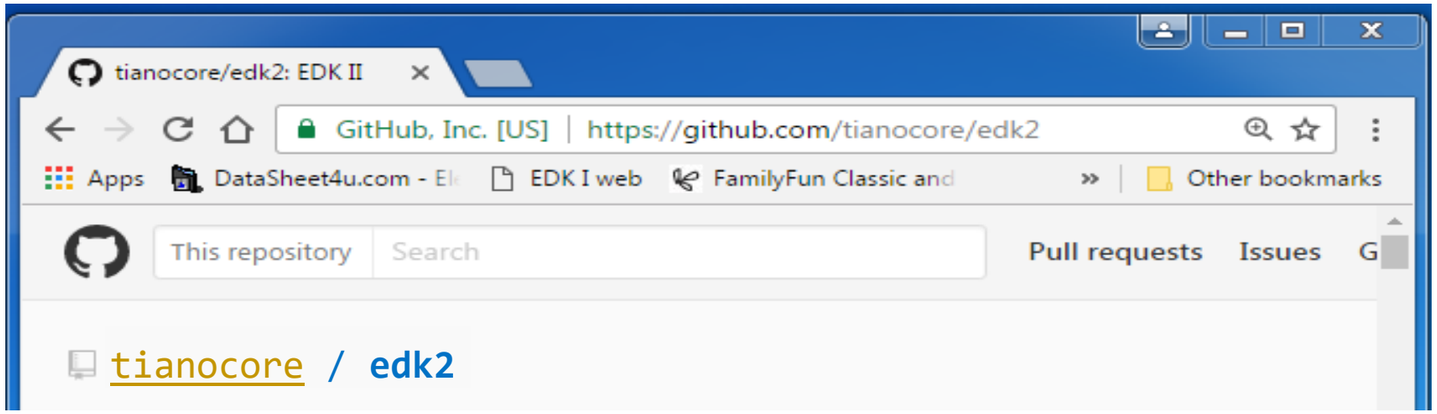
Lesson Objective

- UEFI Network Stack Layers
- EDK II Network Features Overview
- What UEFI Protocols Make Network Work in EDK II
- UEFI HTTP(s) Boot Overview

tianocore

EDK II Network Features Overview

Where is the EDK II Network Stack located?



github.com/tianocore/edk2

Network Related Libraries

MdeModulePkg

Library

NetworkPkg

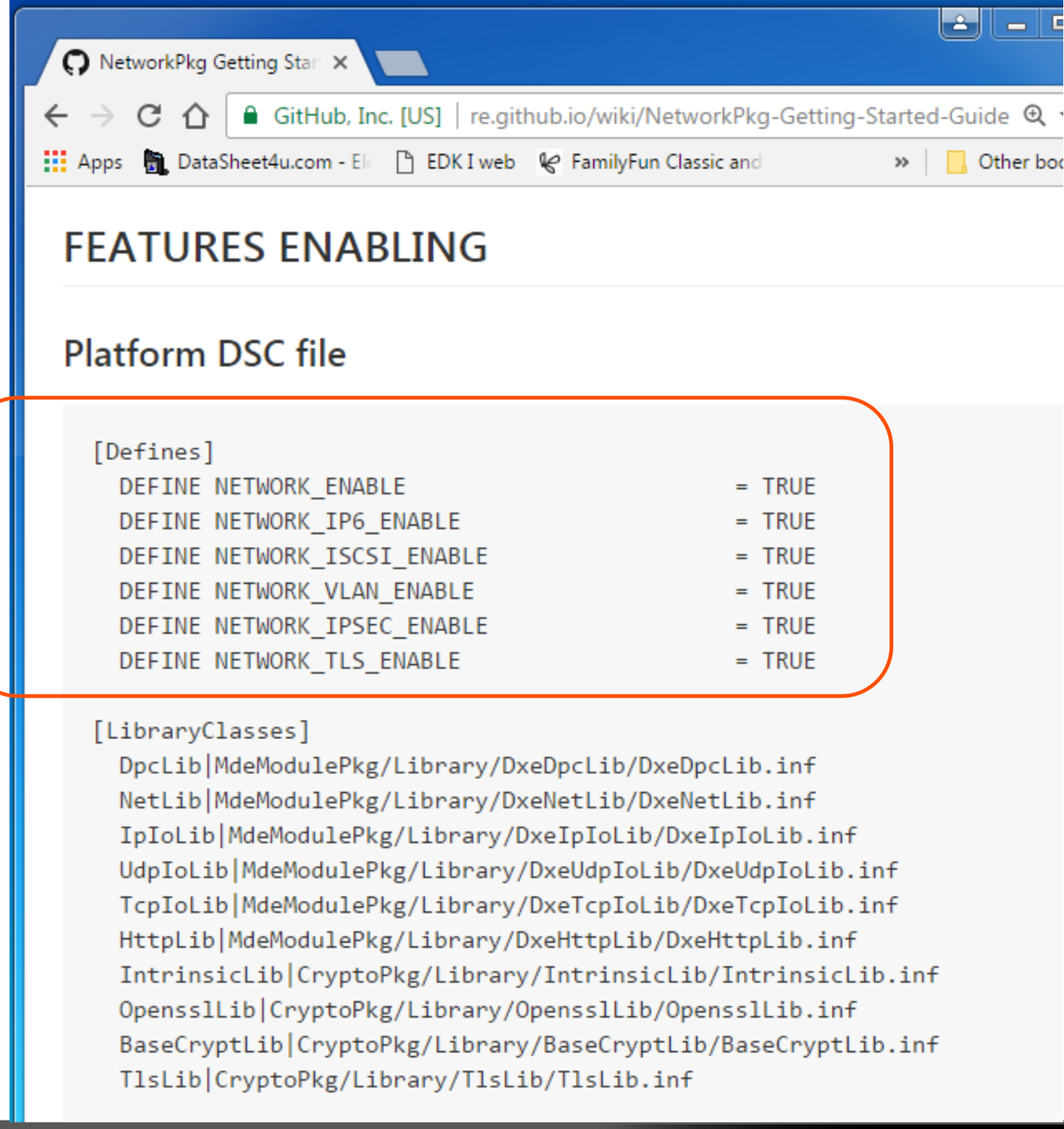
- Application
- ArpDxe
- Dhcp4Dxe
- Dhcp6Dxe
- DnsDxe
- DpcDxe
- HttpBootDxe
- HttpDxe
- HttpUtilitiesDxe
- Include
- Ip4Dxe
- Ip6Dxe
- IScsiDxe
- Library
- MnpDxe
- Mtftp4Dxe
- Mtftp6Dxe
- SnpDxe
- TcpDxe
- TlsAuthConfigDxe
- TlsDxe
- Udp4Dxe
- Udp6Dxe
- UefiPxeBcDxe
- VlanConfigDxe
- WifiConnectionManagerDxe

How to Enable the EDK II Network Stack

Update the Platform DSC and FDF files

- Link: https://github.com/tianocore/tianocore.github.io/wiki/NetworkPkg-Getting-Started-Guide#features-enabling

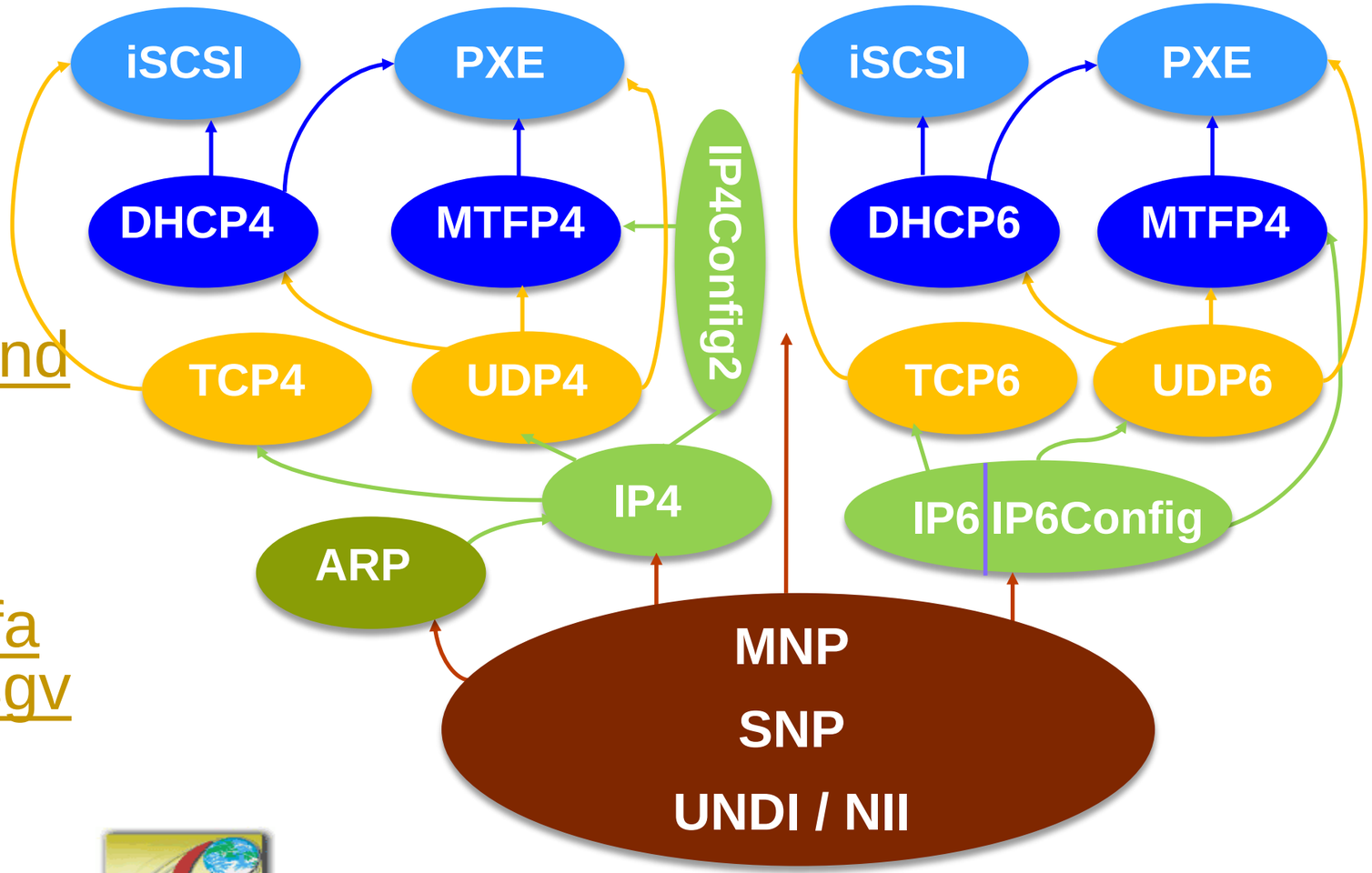
```
DEFINE NETWORK_ENABLE   = TRUE
. . .
```



IP6 Networking - Vendors

IPv6 protocols compliance

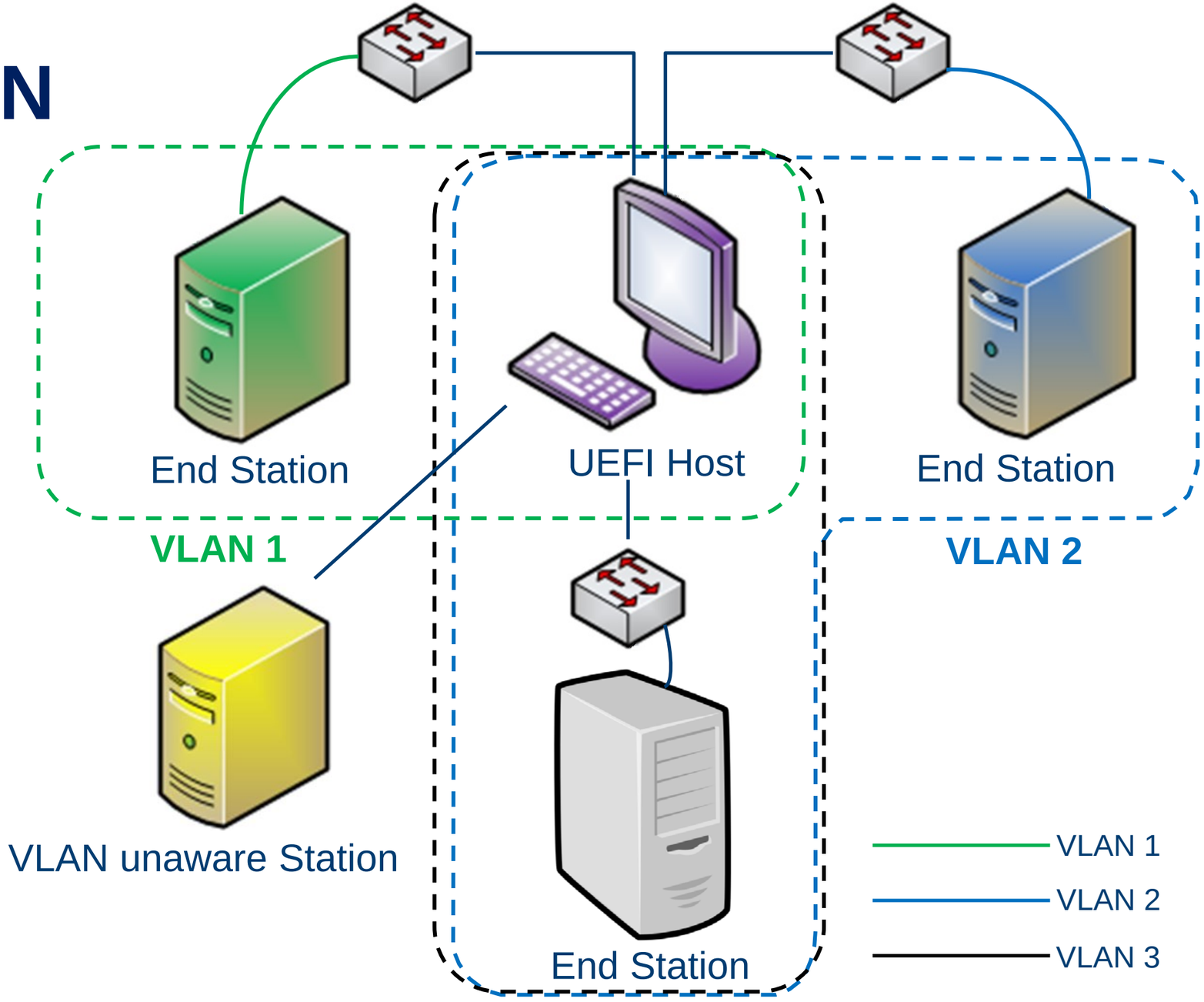
- IPv6 ready logo approved
  http://www.ipv6ready.org/db/index.php/public/
- Requirements for IPv6 transition
  https://www.nist.gov/sites/default/files/documents/itl/antd/usgv6-v1.pdf
- Vendor Testing: https://www-x.antd.nist.gov/usgv6/faq-c.html#vendors



IPv6 UEFI 2.4 B
2011

Virtual LAN - VLAN

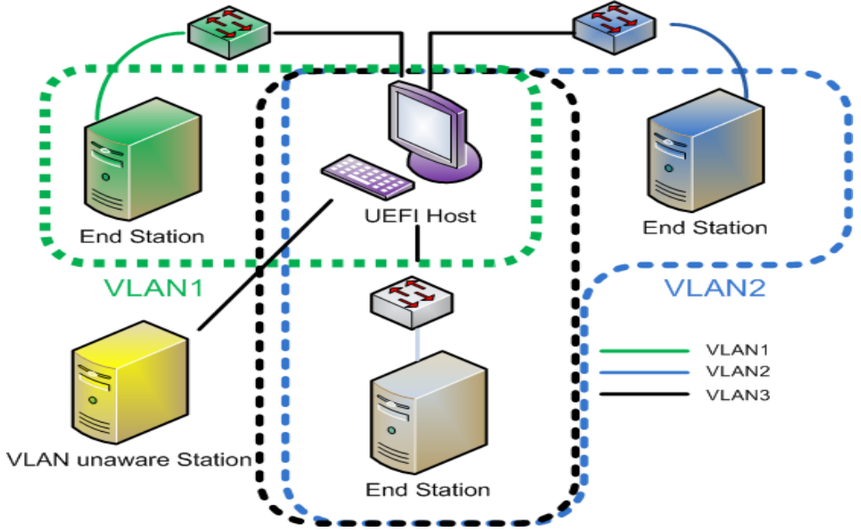
- Logical groups of Stations at the data link layer (Tagging)
  - VLAN's allow a network manager to logically segment a LAN into different broadcast domains Link
- Why VLAN?
  - Performance
  - Security
  - Formation of Virtual Workgroups
  - Simplified Administration
  - Cost
- VLAN Standard: IEEE 802.1Q

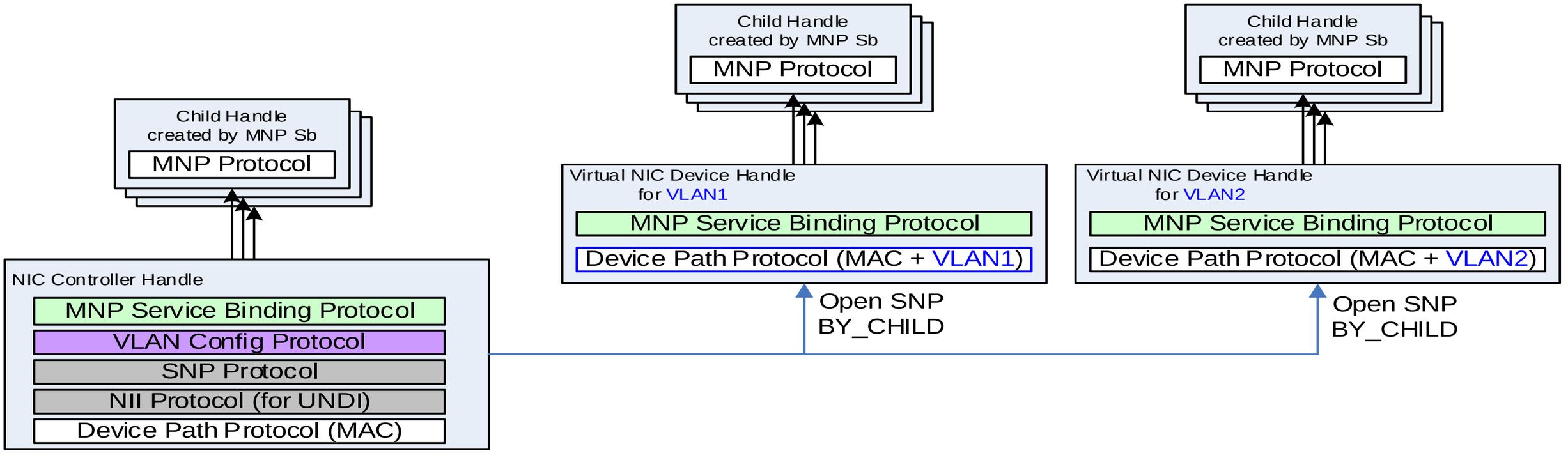


VLAN Support - EDK II

Technology includes

- Support Hybrid LAN topology
- Multiple VLAN for one station
- MNP and VLAN Configuration Protocol
- VLAN configuration by Shell Application `Vconfig`

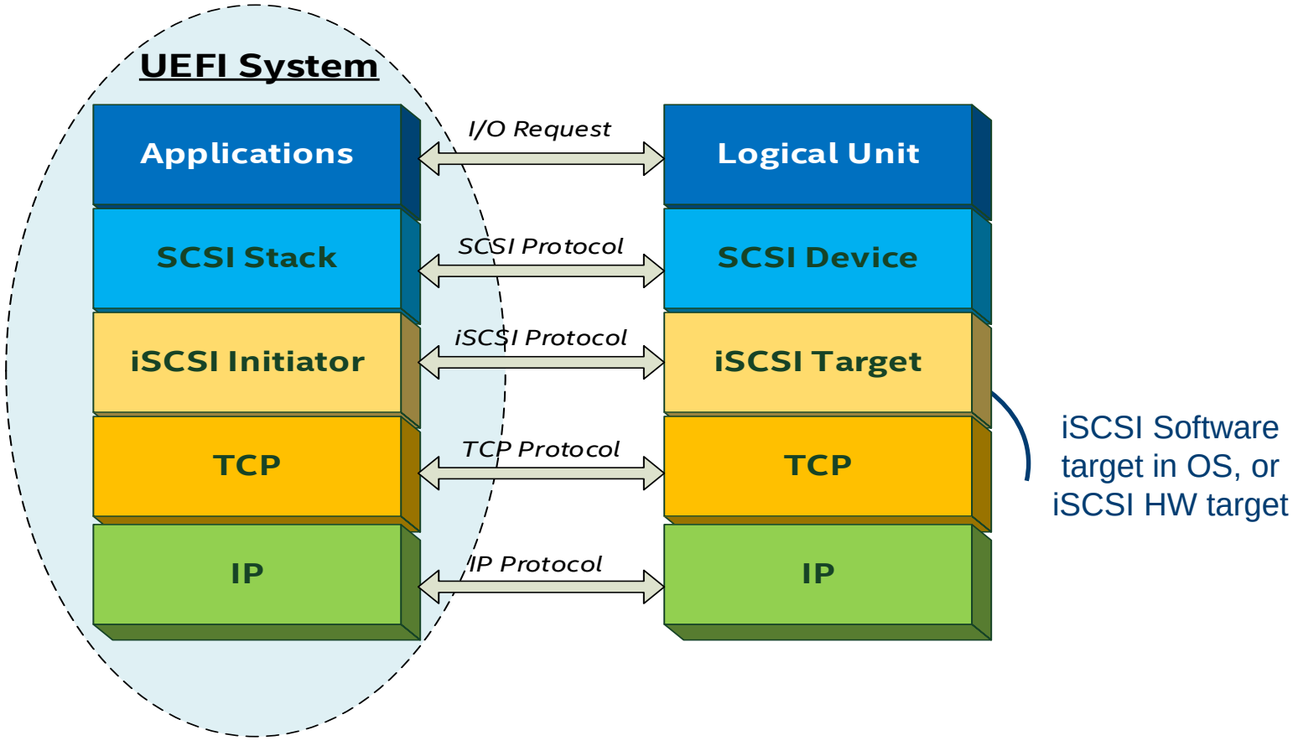


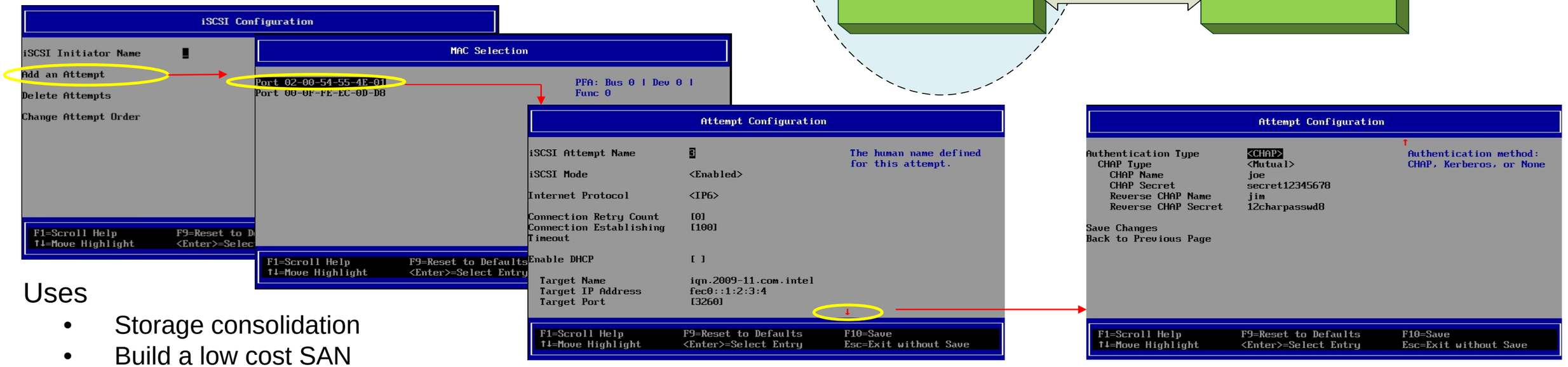


UEFI iSCSI Solutions

SAN/Data center boot over iSCSI

- Manual/DHCP based configuration allowed
- Cryptographic logon with CHAP
- Multi-path/fail-over capable
- User Interface using HII

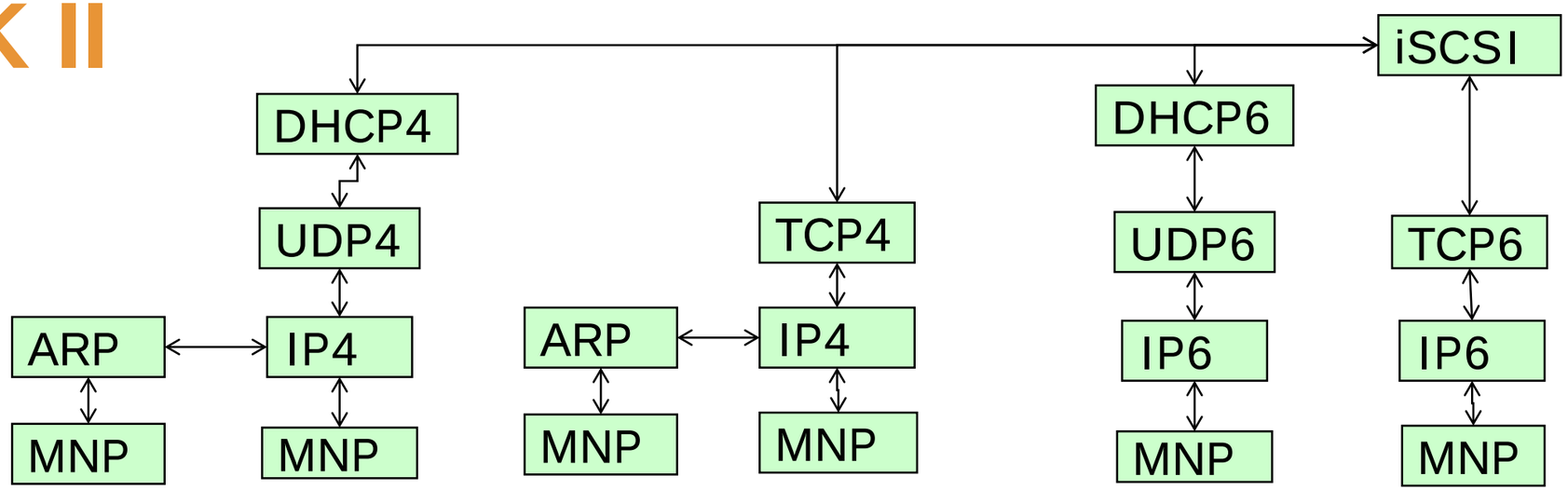


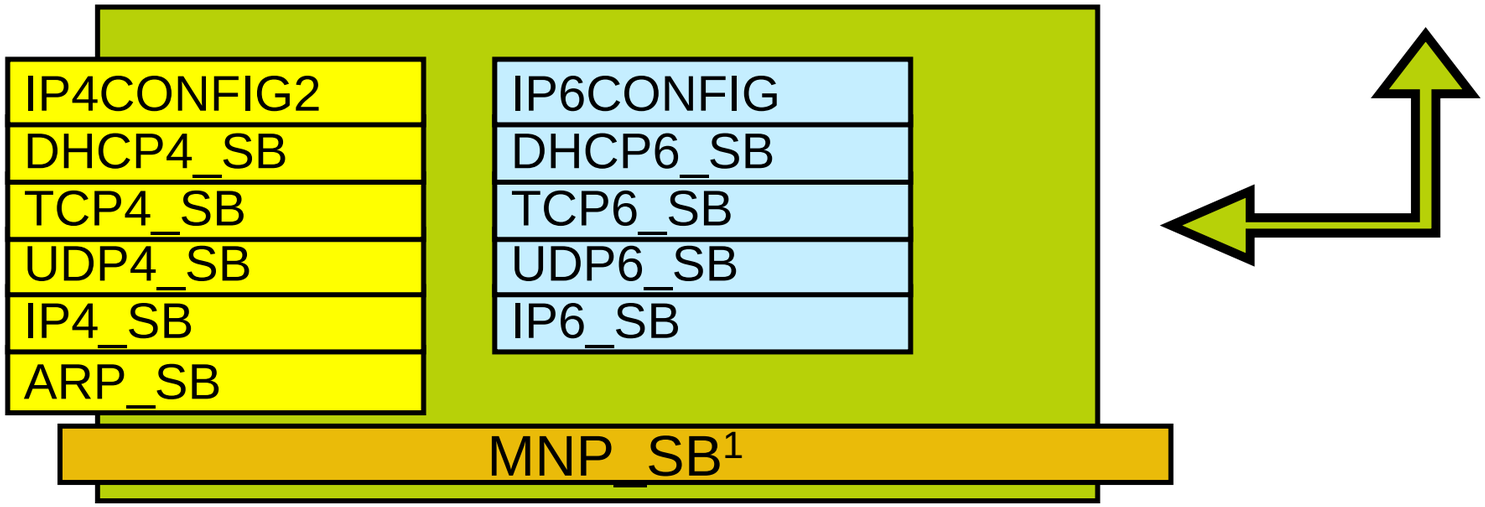


Uses

- Storage consolidation
- Build a low cost SAN
- Cluster Shared Volumes
- Diskless Booting

Dual-Stack Heritage – iSCSI usage model EDK II





[1] Service Binding Protocol

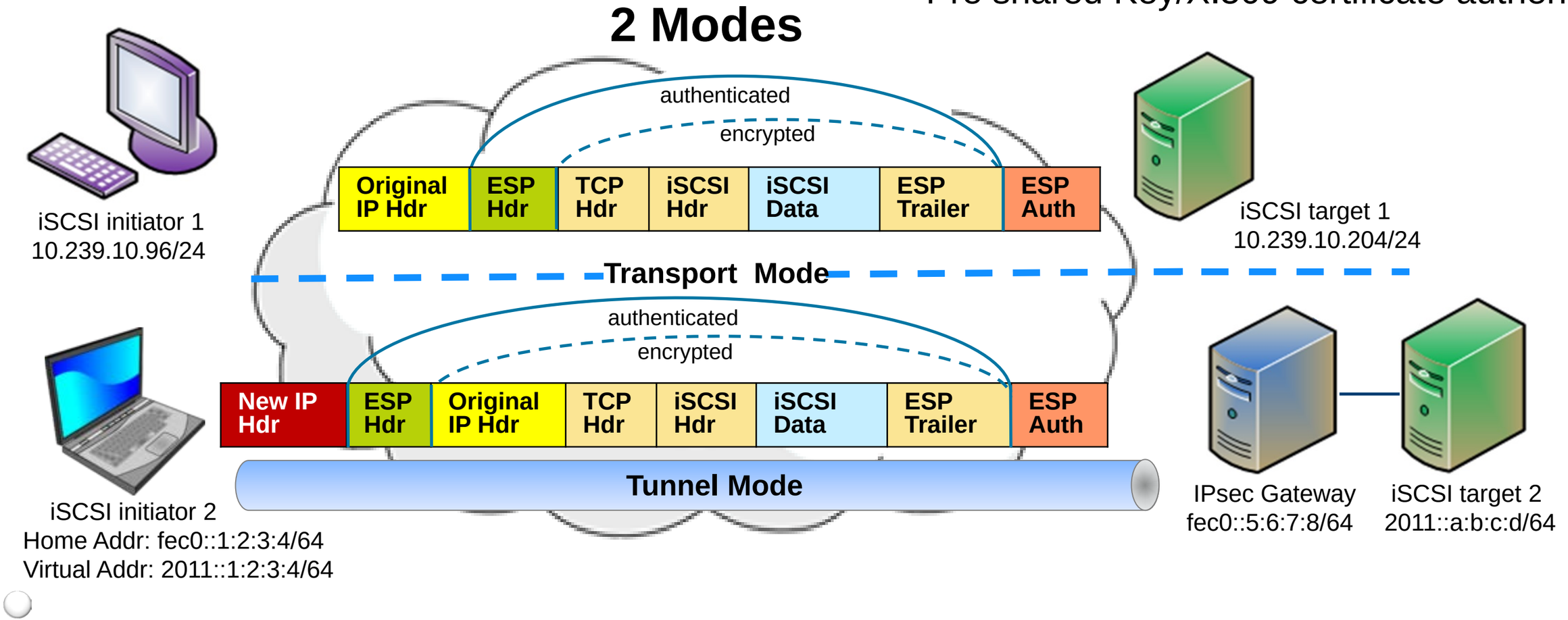
IPsec – Network Security

Secure Internet Protocol Communication

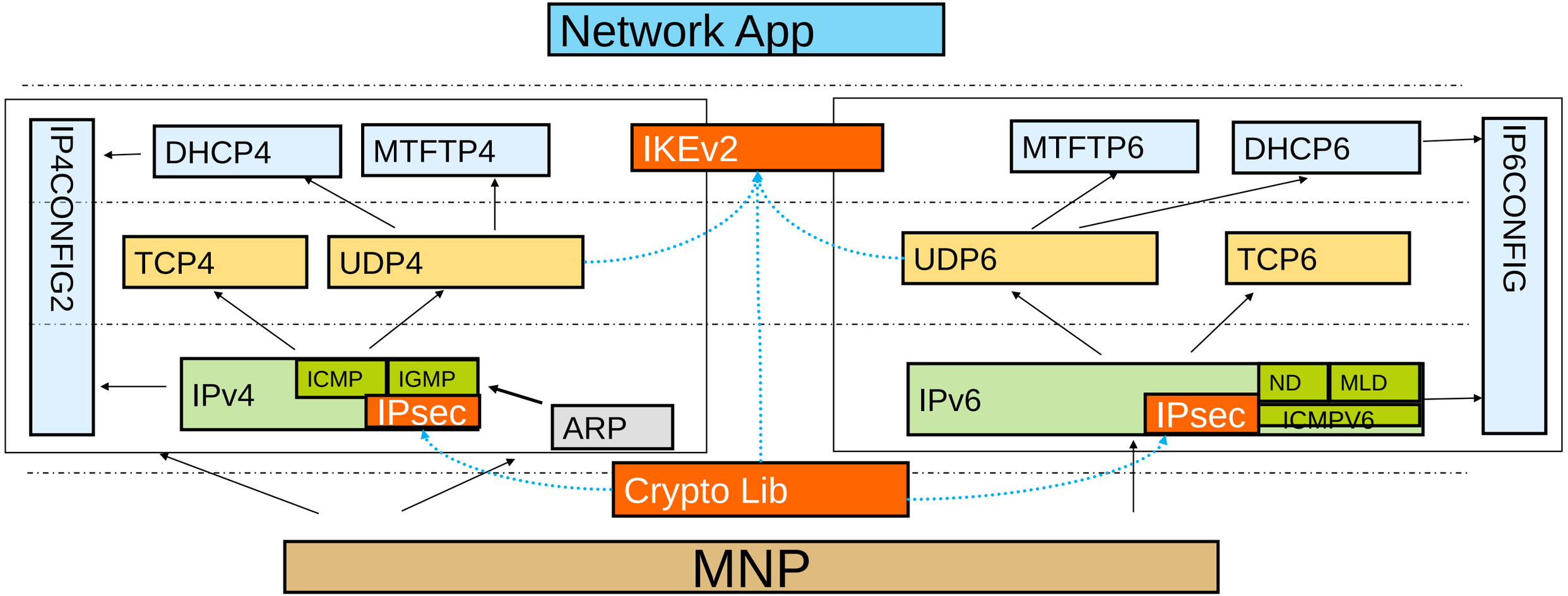
- Protects any application traffic across an IP network
- Mandatory for IPv6

Features include

- AH, ESP, IKE version 2
- HMAC-SHA1, TripleDES-CBC, AES-CBC
- Modes of operation : Transport vs. Tunnel
- Pre shared Key/X.509 certificate authentication

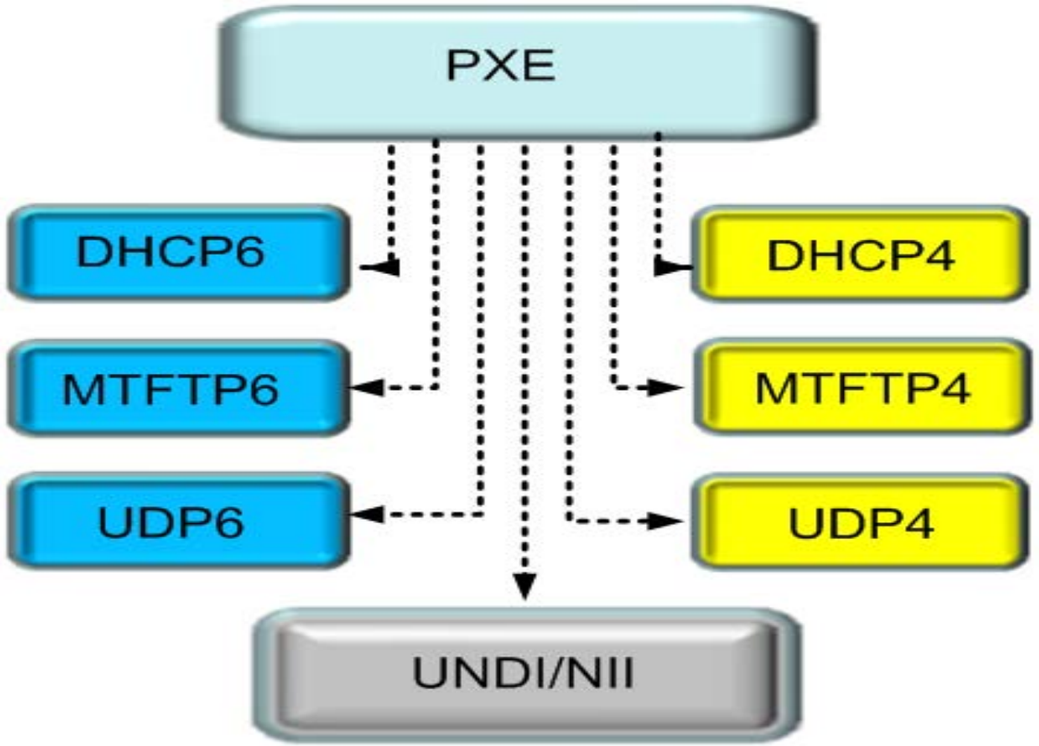


IPsec support: shared



UEFI PXE Solutions

- Preboot eXecution Environment
  - General network booting
    - Independent of data storage device
  - IPv4 based PXE is defined in PXE 2.1
  - IPv6 based PXE is defined in UEFI 2.3
- Technology includes
  - Dual network stack support
  - Evolution of network boot to IPv6 defined
    - IETF RFC 5970
  - DMI-UUID support
    - Use SMBIOS system GUID as UUID

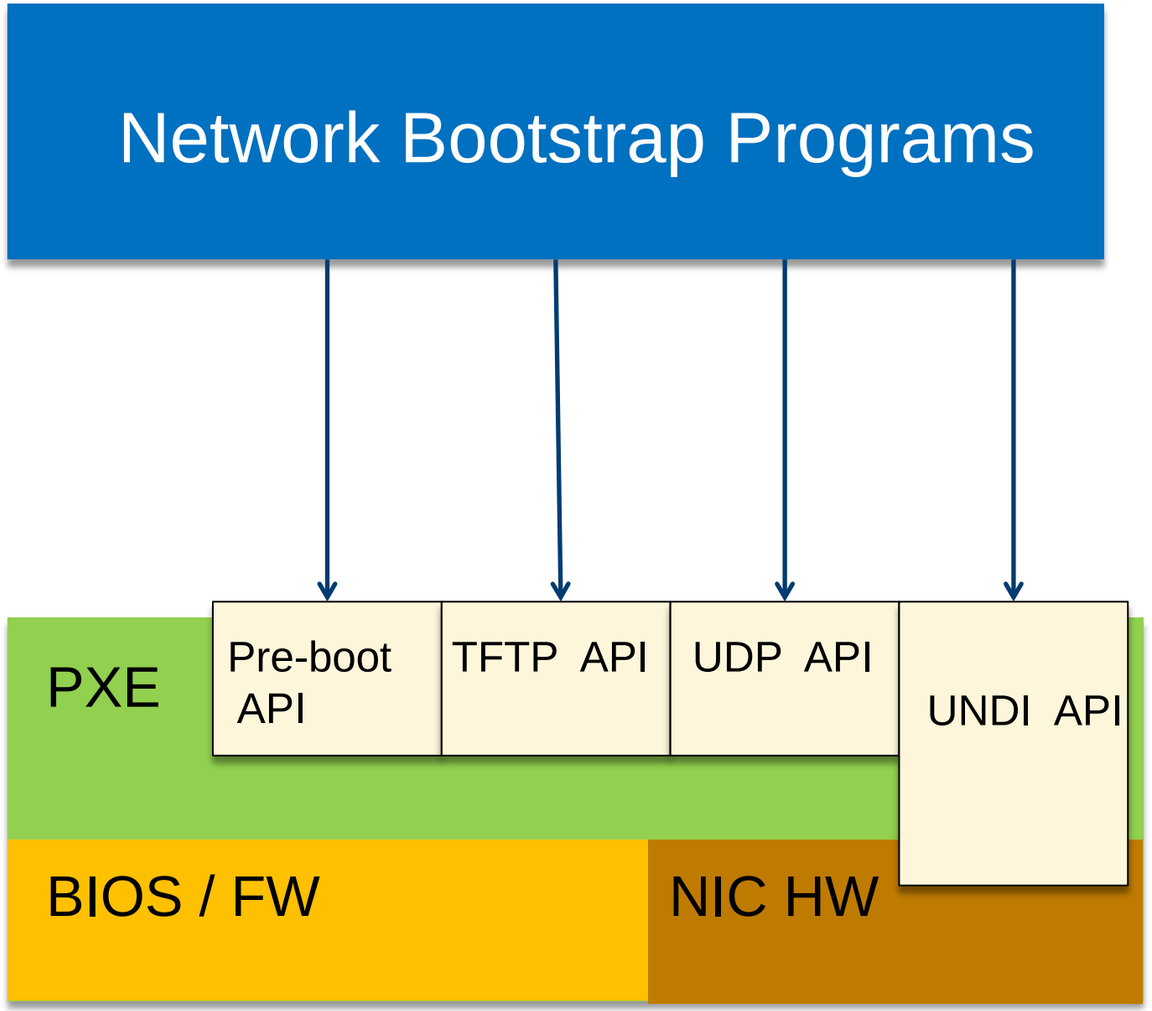


BUT

BUT PXE is not keeping up with modern data center needs

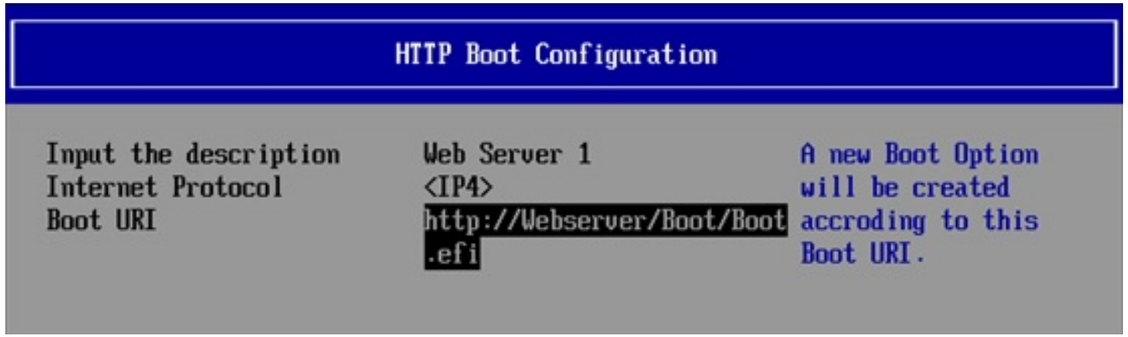
PXE Boot Challenges

- Security Issues
  - Only physical. No encryption or authentication.
  - Rouge DHCP servers, man-in-the-middle attacks
- Scaling issues
  - Circa 1998
  - TFTP timeouts / UDP packet loss
  - Download time = deployment time = $$$
  - Aggravated in density-optimized data centers
- OEMs and users workarounds "*duct-tape*"
  - Chain-load 3rd party boot loaders (iPXE, mini-OS)
  - Alternative Net Booting (SAN, iSCSI, etc.)
- Open source PXE iPXE issues (http://ipxe.org) – pre UEFI 2.5

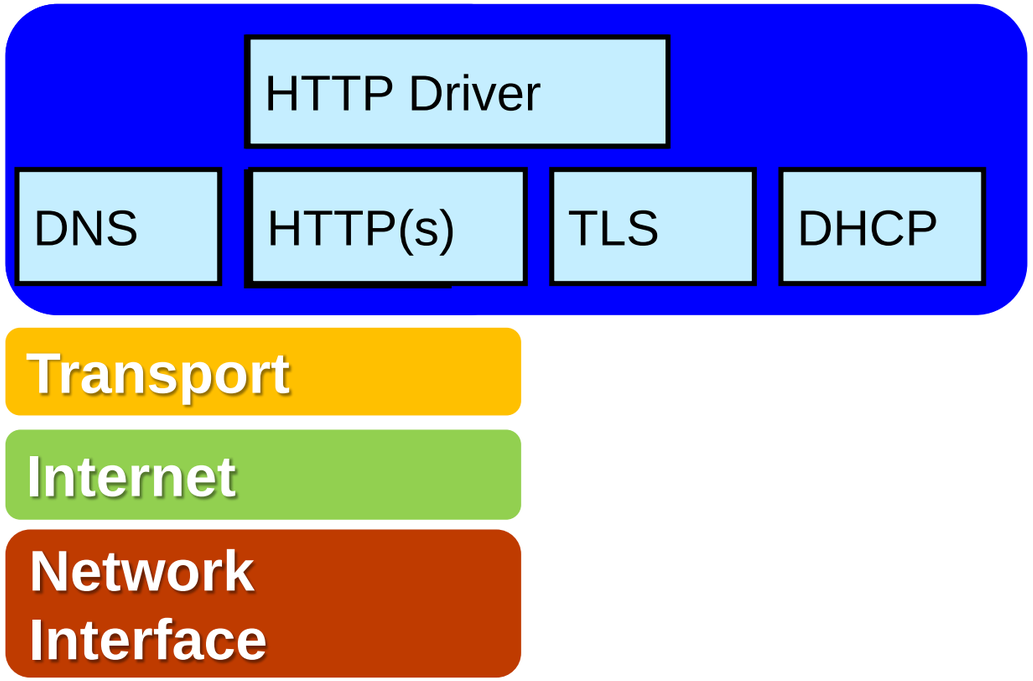


Why not solve PXE Boot challenges natively with standards using UEFI

HTTP(s) Boot Solutions



Add HTTP(s) to Network Stack



UEFI 2.5 defined RAM Disk device path nodes

- Standard access to a RAM Disk in UEFI and Virtual CD (ISO image)

ACPI 6.0 NVDIMM Firmware Interface Table (NFIT)

- Describe the RAM Disks to the OS
- Runtime access of the ISO boot image in memory

HTTP UEFI 2.5 2015

UEFI HTTP(S) BOOT OVERVIEW

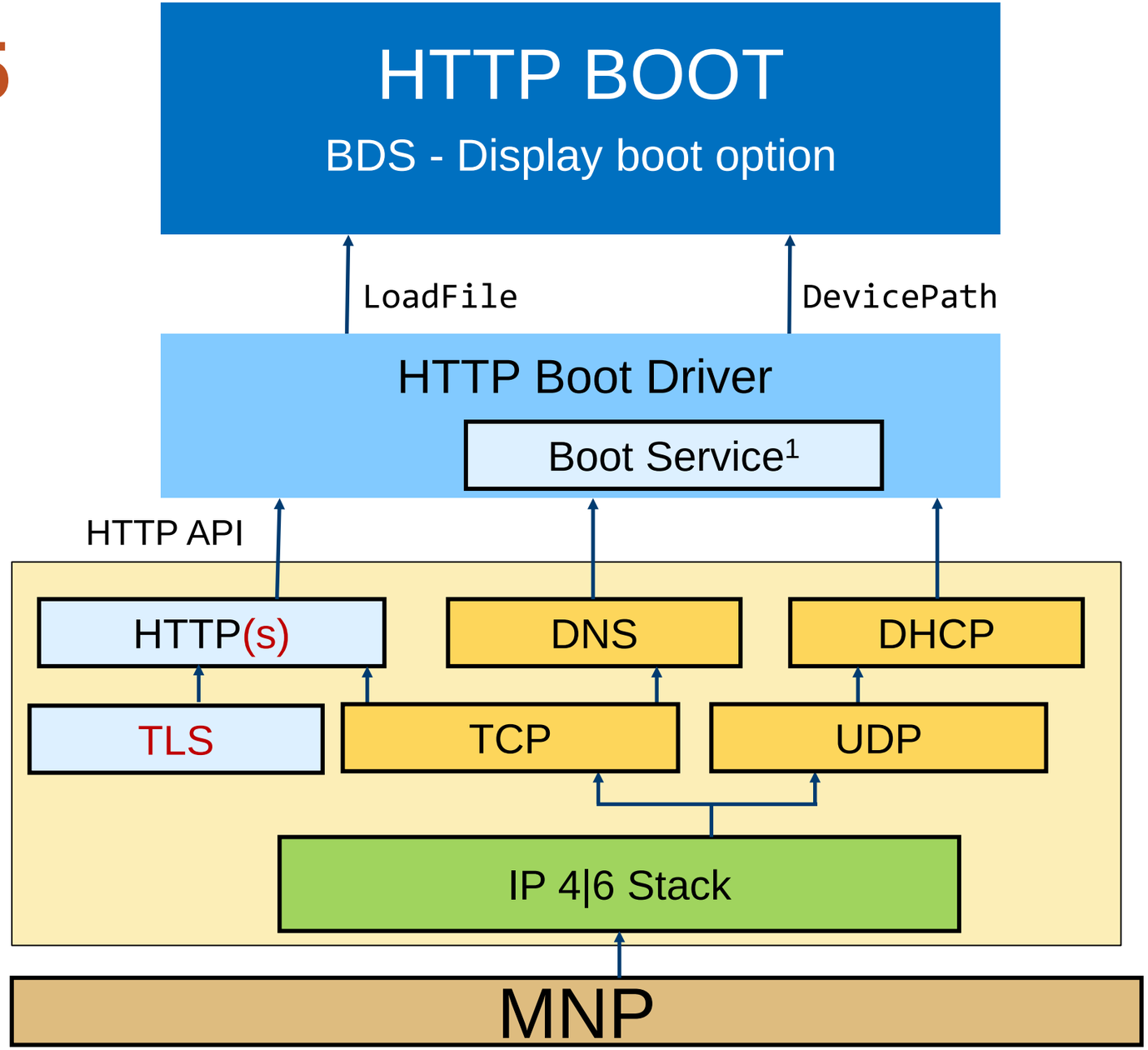
UEFI HTTP Boot Overview
-HTTP protocol for network booting

- HTTP can handle much larger files than TFTP, and scale to much larger distances. You can easily download multi-megabyte files, such as a Linux kernel and a root file system, and you can download from servers that are not on your local area network
- Booting from HTTP is as simple as replacing the DHCP filename field with an http:// URL
  - Specify URI[1]-based pointers to the “Network Boot Program (NBP)”, the binary image to download and run, which can be used using HTTP instead of TFTP
- DHCP Servers will need to support HTTP Boot

[1]Uniform Resource Identifiers (URI)
[1]Uniform Resource Locator (URL)

HTTP(s) Boot UEFI 2.5
- Network Stack

- DNS IPv4 / IPv6
- HTTP IPv4 / IPv6
- TLS for HTTPs
- HTTP Boot Driver

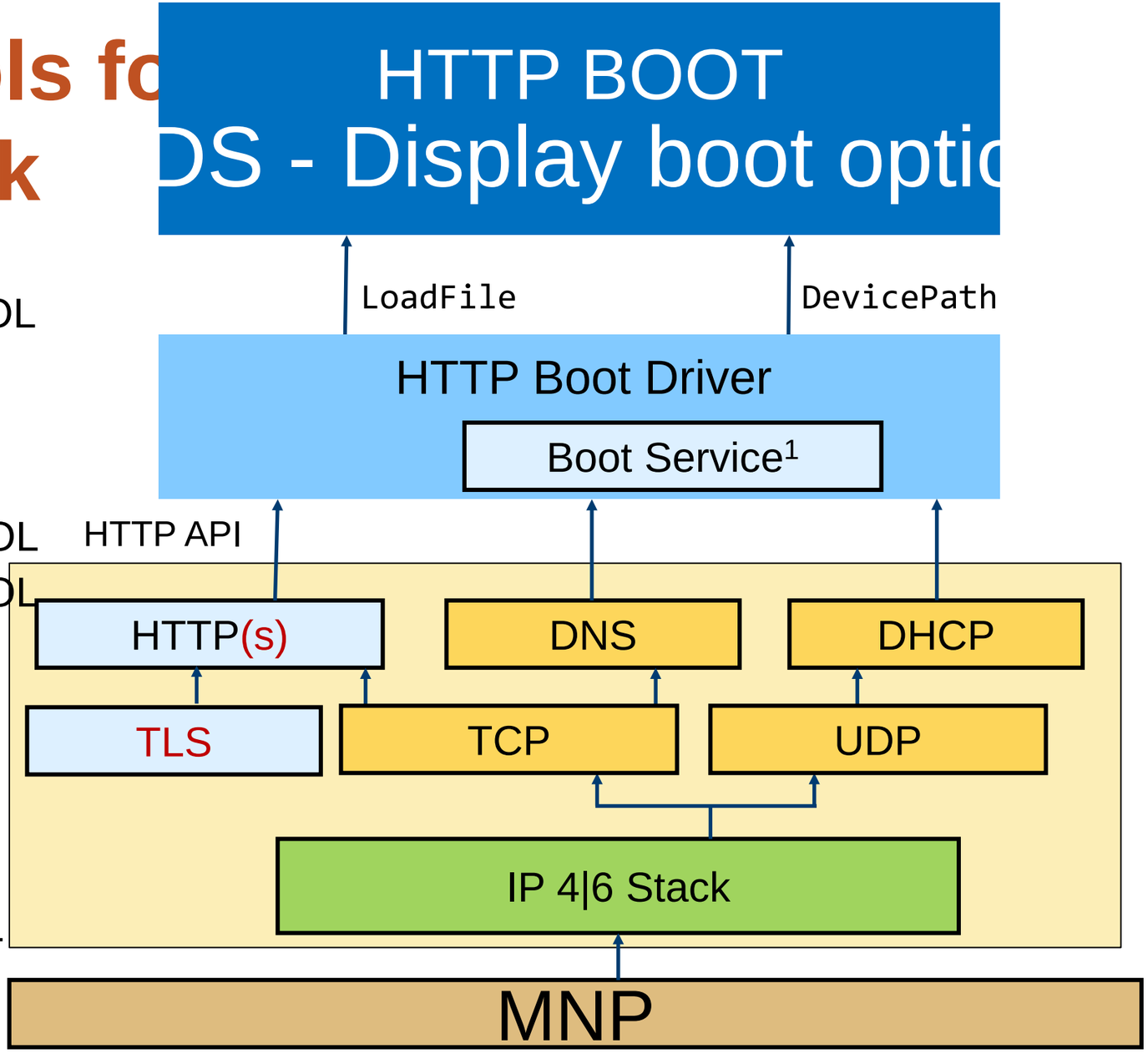


UEFI 2.5
2015

[1]Boot Service Discovery / Configuration

UEFI & EDK II Protocols fo
HTTP(s) Network Stack

- HTTP support
  - EFI_HTTP_SERVICE_BINDING_PROTOCOL
  - EFI_HTTP_PROTOCOL
  - EFI_HTTP_UTILITIES_PROTOCOL
- DNS Support
  - EFI_DNS4_SERVICE_BINDING_PROTOCOL
  - EFI_DNS6_SERVICE_BINDING_PROTOCOL
  - EFI_DNS4_PROTOCOL
  - EFI_DNS6_PROTOCOL
  - EFI_IP4_CONFIG2_PROTOCOL
  - EFI_IP6_CONFIG_PROTOCOL
- TLS support
  - EFI_TLS_SERVICE_BINDING_PROTOCOL
  - EFI_TLS_PROTOCOL
  - EFI_TLS_CONFIGURATION_PROTOCOL



[1]Boot Service Discovery / Configuration

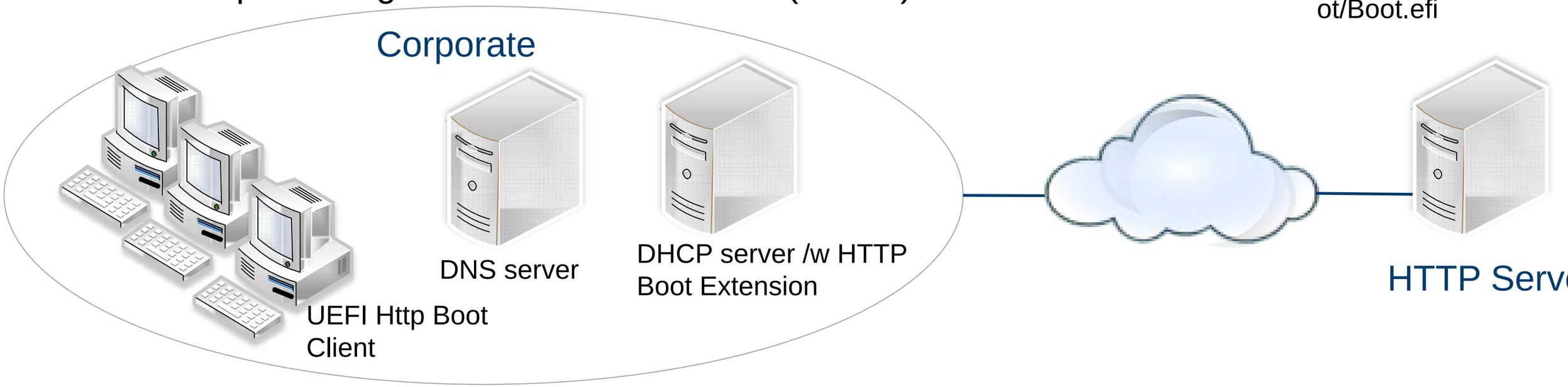
UEFI Native HTTP Boot – Corporate Environment

HTTP Protocols

- Boot from a configured URL
- Target can be:
  - UEFI Network Boot Program (NBP)
  - Shrink-wrapped ISO image
- URL pre-configured or auto-discovered (DHCP)

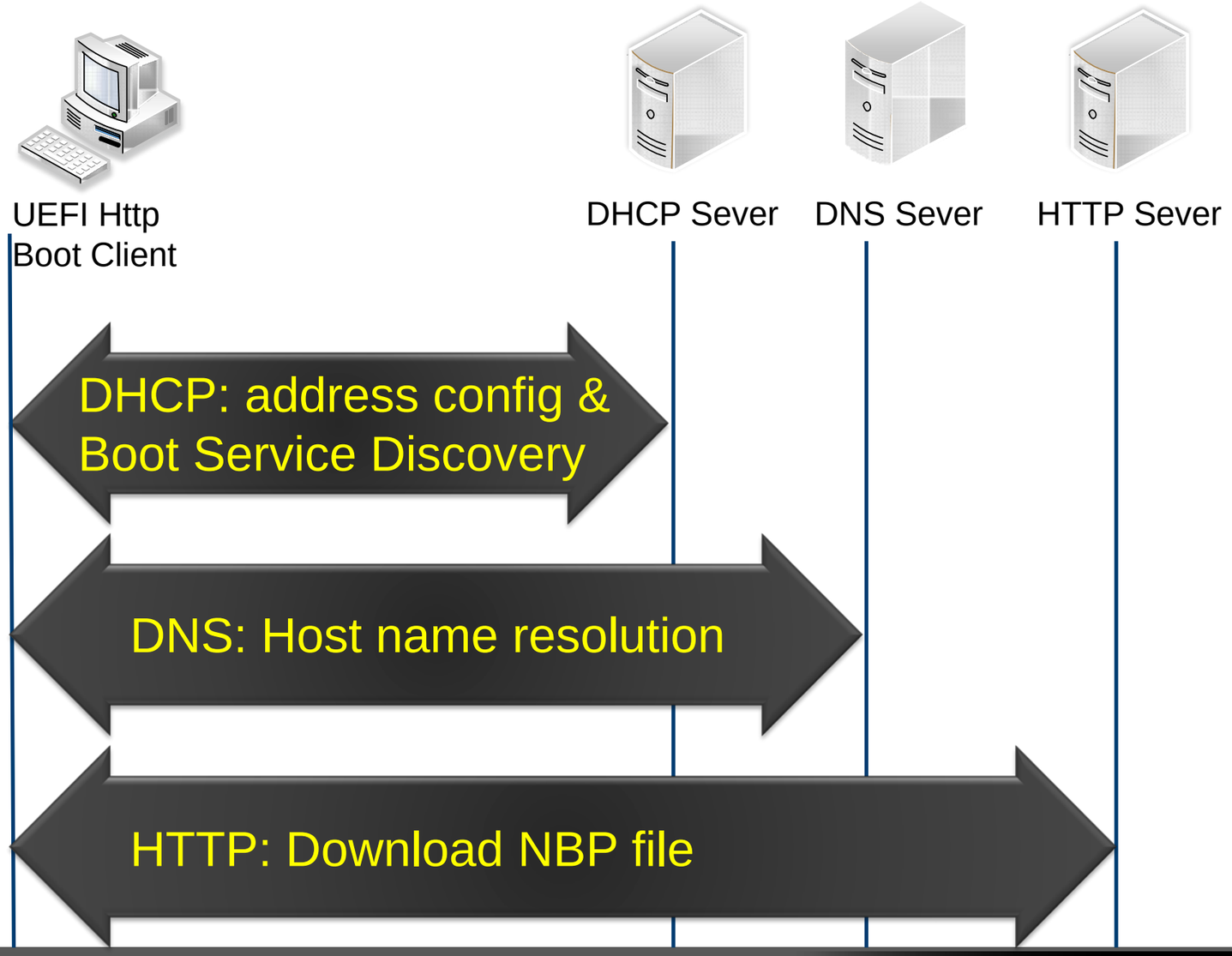
Addresses PXE issues

- HTTP(s) addresses security
- TCP reliability
- HTTP load balancing



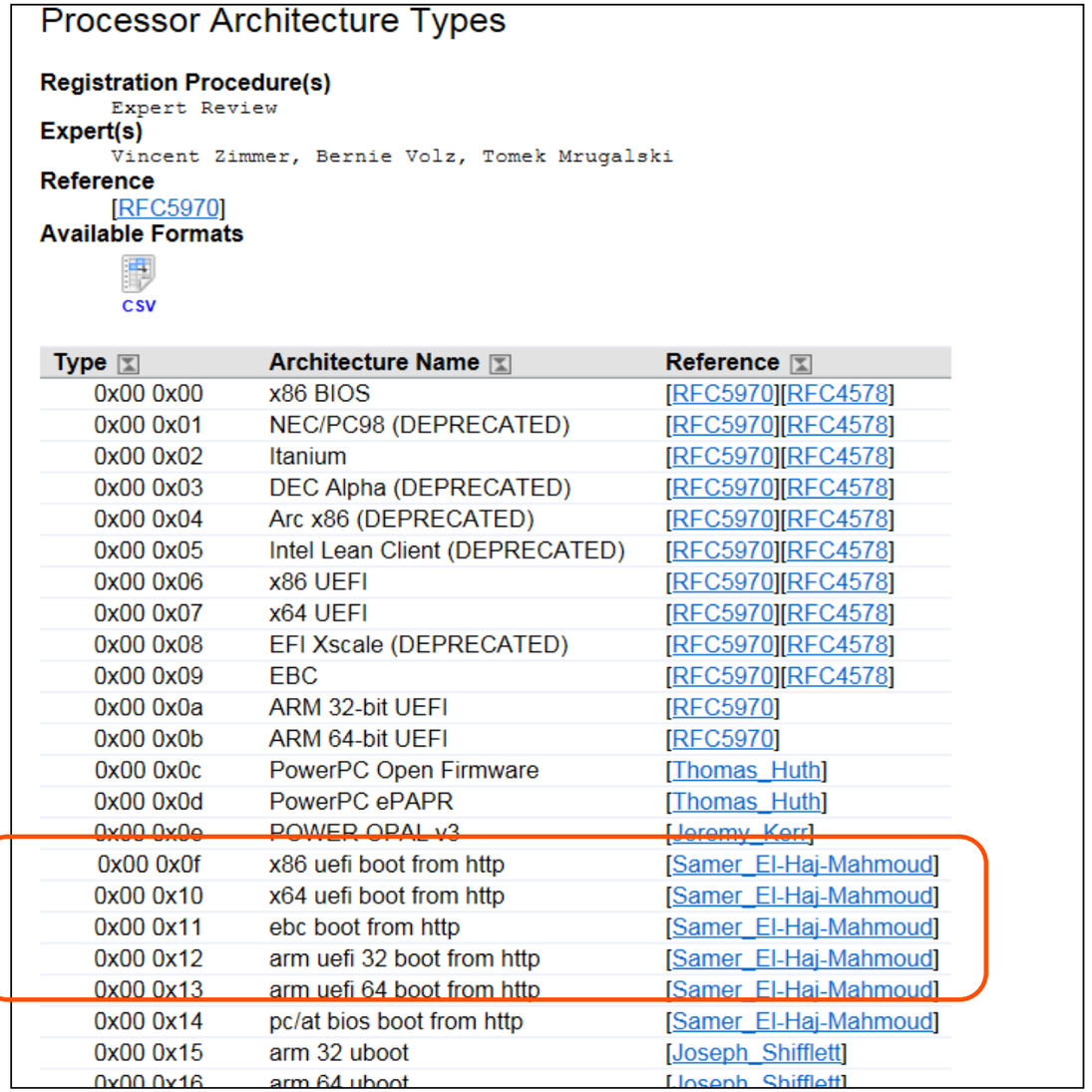
HTTP Boot DHCP Discovery

- New HTTP Boot “Architectural Types” to distinguish from PXE
- Client sends DHCP Discover **request**
- DHCP Server responds with **offer** that includes the boot file URI
- Clients resolves URI server name from **DNS**
- Client downloads boot image from HTTP server using HTTP(s)



HTTP(s) Boot Discovery - Architectural Types

- DHCP - http://www.iana.org/assignments/dhcpv6-parameters/dhcpv6-parameters.xml
- IPv4/IPv6 DHCP Discover request
  - DHCP Option 93: Client system Architecture
  - DHCPv6 Option 61: Client system Architecture
    - 0x10 = x64 UEFI boot from HTTP
    - 0x0F = x86 UEFI boot from HTTP
- Server responds with DHCPOFFER that includes the boot file HTTP URI for the requested processor architecture



Processor Architecture Types

Registration Procedure(s)
Expert Review
Expert(s)
Vincent Zimmer, Bernie Volz, Tomek Mrugalski
Reference
[RFC5970]
Available Formats
CSV

Type	Architecture Name	Reference
0x00 0x00	x86 BIOS	[RFC5970][RFC4578]
0x00 0x01	NEC/PC98 (DEPRECATED)	[RFC5970][RFC4578]
0x00 0x02	Itanium	[RFC5970][RFC4578]
0x00 0x03	DEC Alpha (DEPRECATED)	[RFC5970][RFC4578]
0x00 0x04	Arc x86 (DEPRECATED)	[RFC5970][RFC4578]
0x00 0x05	Intel Lean Client (DEPRECATED)	[RFC5970][RFC4578]
0x00 0x06	x86 UEFI	[RFC5970][RFC4578]
0x00 0x07	x64 UEFI	[RFC5970][RFC4578]
0x00 0x08	EFI Xscale (DEPRECATED)	[RFC5970][RFC4578]
0x00 0x09	EBC	[RFC5970][RFC4578]
0x00 0x0a	ARM 32-bit UEFI	[RFC5970]
0x00 0x0b	ARM 64-bit UEFI	[RFC5970]
0x00 0x0c	PowerPC Open Firmware	[Thomas_Huth]
0x00 0x0d	PowerPC ePAPR	[Thomas_Huth]
0x00 0x0e	POWER OPAL v3	[Jeremy_Kerr]
0x00 0x0f	x86 uefi boot from http	[Samer_El-Haj-Mahmoud]
0x00 0x10	x64 uefi boot from http	[Samer_El-Haj-Mahmoud]
0x00 0x11	ebc boot from http	[Samer_El-Haj-Mahmoud]
0x00 0x12	arm uefi 32 boot from http	[Samer_El-Haj-Mahmoud]
0x00 0x13	arm uefi 64 boot from http	[Samer_El-Haj-Mahmoud]
0x00 0x14	pc/at bios boot from http	[Samer_El-Haj-Mahmoud]
0x00 0x15	arm 32 uboot	[Joseph_Shifflett]
0x00 0x16	arm 64 uboot	[Joseph_Shifflett]

iPXE – UEFI HTTP Chainloading

UEFI HTTP Boot client to chainload iPXE from an HTTP server (HTTP boot to iPXE then run iPXE to HTTP download)

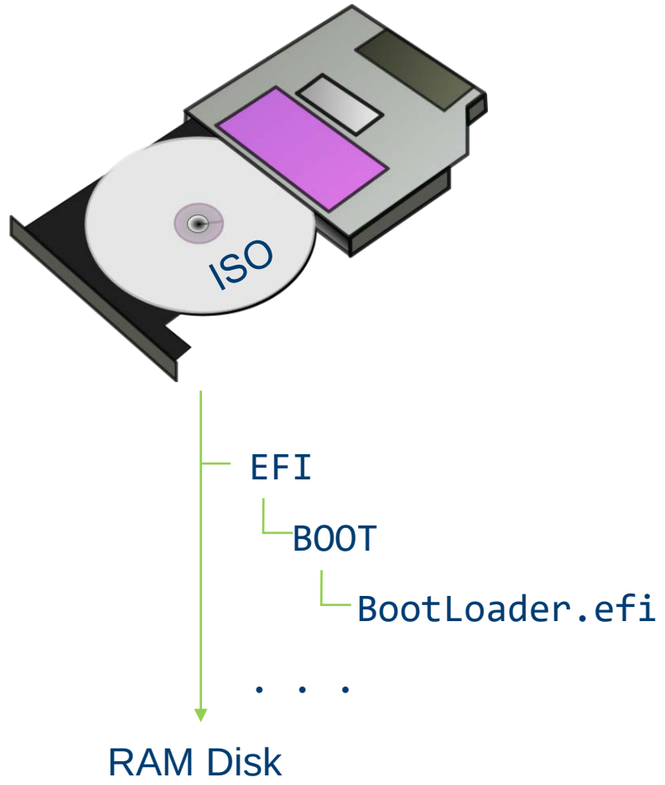
- Eliminates need for separate TFTP server
- UEFI HTTP Boot client will download and boot iPXE
- iPxe offers advanced features to download and boot OS

- Application note: http://ipxe.org/appnote/uefihttp

2 Options to address the PXE challenges: Native UEFI HTTP Boot and iPXE using UEFI HTTP

RAM Disk Boot from HTTP

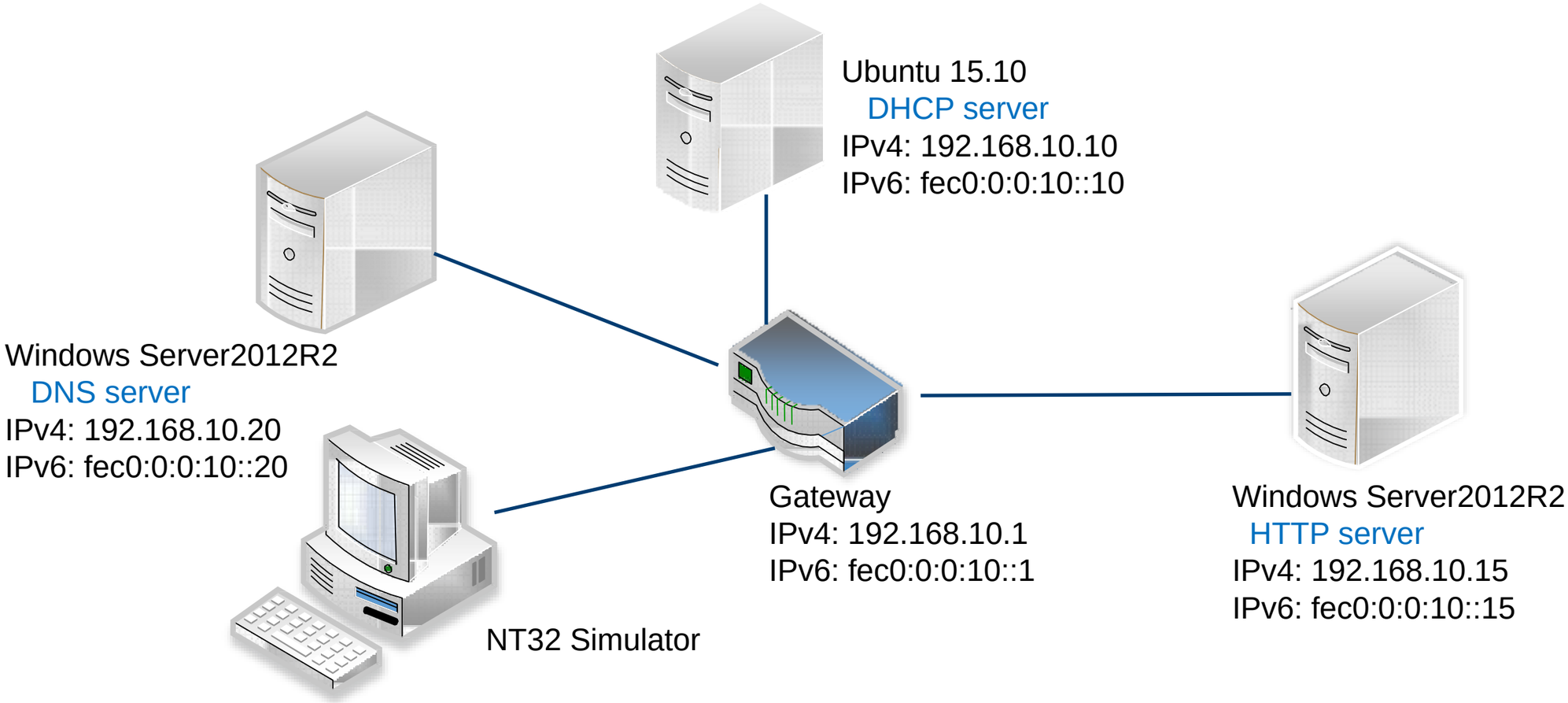
- UEFI 2.5 defined RAM Disk device path nodes
  - Standard access to a RAM Disk in UEFI
  - Supports Virtual Disk and Virtual CD (ISO image) in persistent or volatile memory
  - Device Path: Type:4 Subtype: 9
- ACPI 6.0 NVDIMM Firmware Interface Table (NFIT)
  - Describe the RAM Disks to the OS
  - Runtime access of the ISO boot image in memory
- Supported Image Types
  - *.ISO Virtual CD Image
  - *.img Virtual Disk Image
  - *.efi UEFI Executable Image



Feature Enabling:

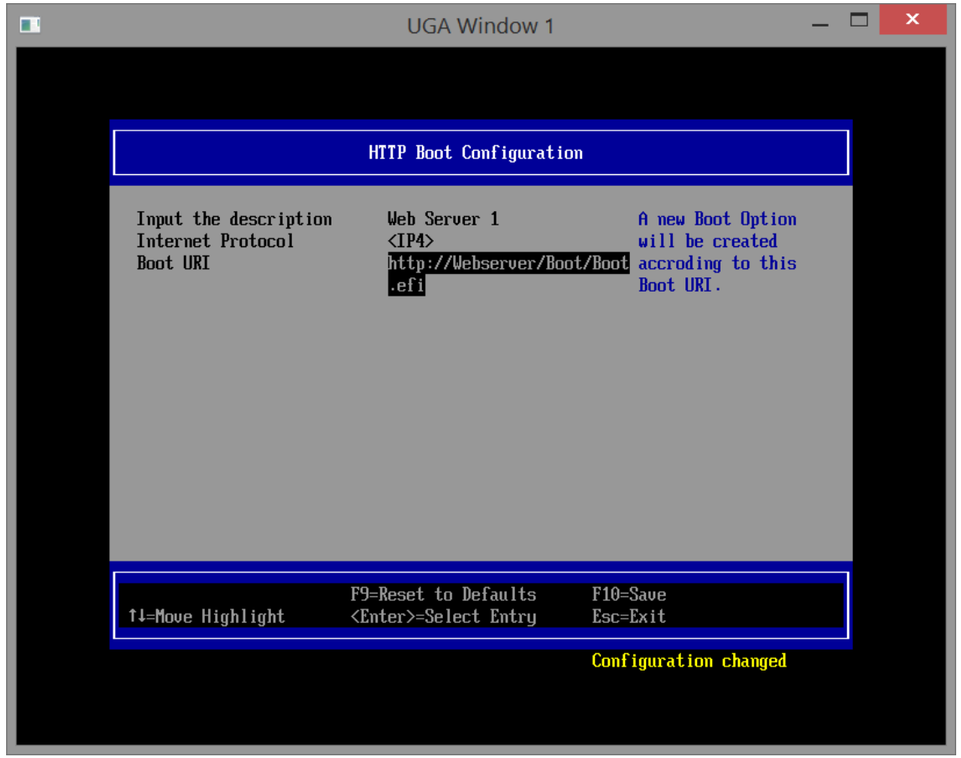
Add Edk2 `RamDiskDxe.inf` to Platform .DSC

UEFI Http Boot Example

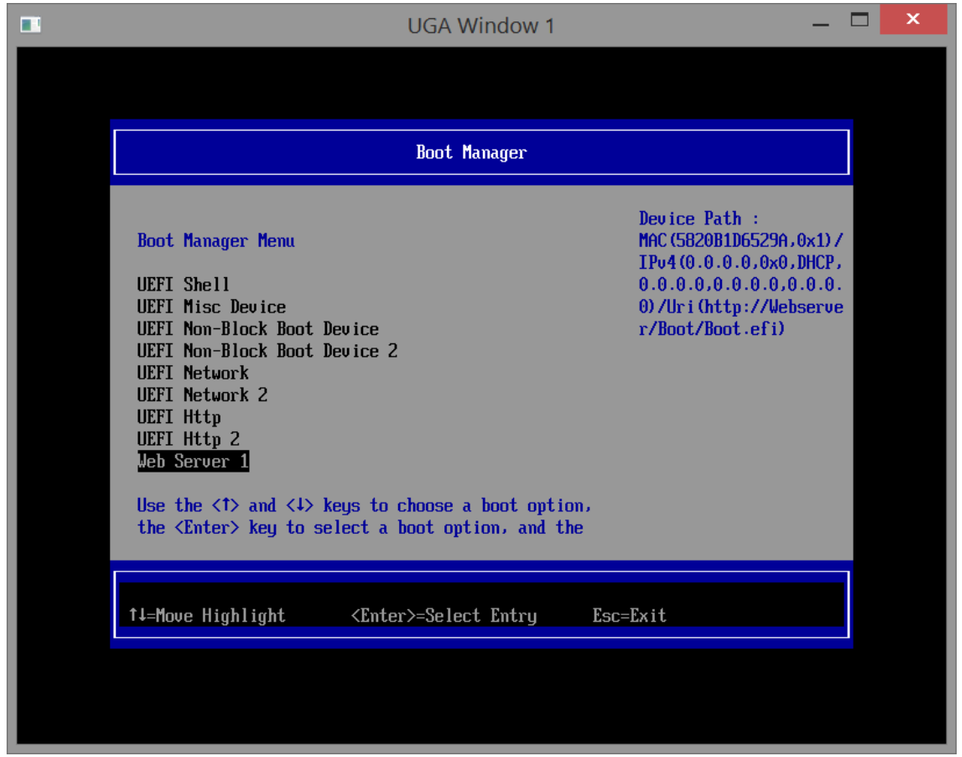


Public paper EDK II HTTP Boot Getting Started Guide for a step-by-step guide of the HTTP Boot enabling and server configuration in **corporate environment**.

EDK II HTTP Boot Configuration



In the main page of Boot Manager Menu, enter [Device Manager] -> [Network Device List] -> Select a NIC device -> [HTTP Boot Configuration], set the HTTP boot parameters such as the boot option title, IP start version and the URI address

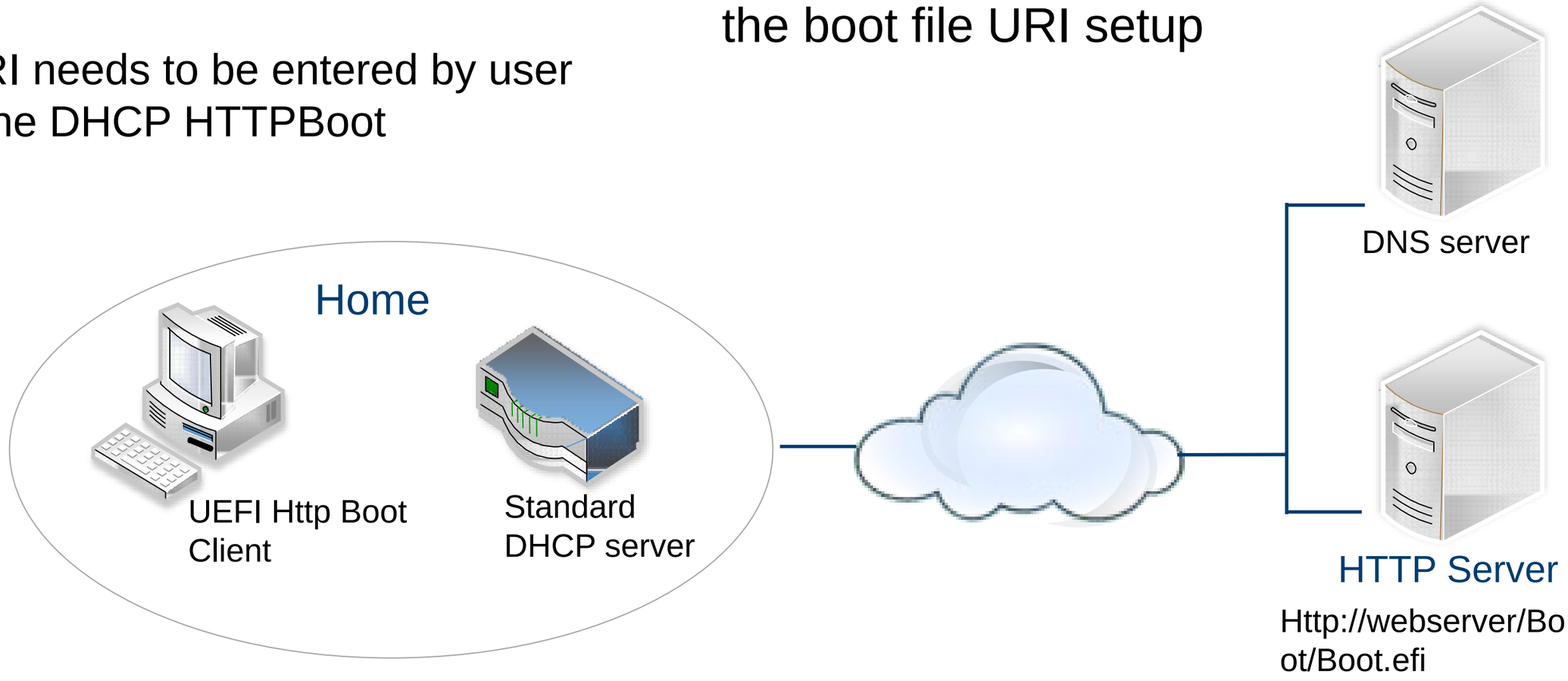


Save the configuration and back to the main page, enter [Boot Manager] menu and select the new created boot option to start the HTTP Boot

UEFI Native HTTP Boot – Home Environment

- Only a Standard DHCP server is available for host IP configuration assignment
- Boot file URI needs to be entered by user instead of the DHCP HTTPBoot extensions.

- The EDK II HTTP Boot Driver provides a configuration pages for the boot file URI setup



Getting Started Guides

HTTP:

Wiki Page https://github.com/tianocore/tianocore.github.io/wiki/HTTP-Boot

PDF HttpBootGettingStartedGuide_0_9.pdf

HTTPS:

Wiki Page https://github.com/tianocore/tianocore.github.io/wiki/HTTPs-Boot

PDF Getting Started with UEFI HTTPS Boot on EDK II .pdf

tianocore

Questions?

Return to Main Training Page

Return to Training Table of contents for next presentation link

tianocore

ACKNOWLEDGEMENTS