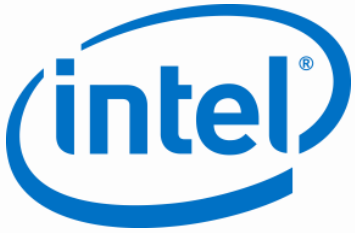
presented by



UEFI Network and Security Update

UEFI US Fall Plugfest – September 20 - 22, 2016

Presented by Vincent Zimmer –

vincent.zimmer@intel.com

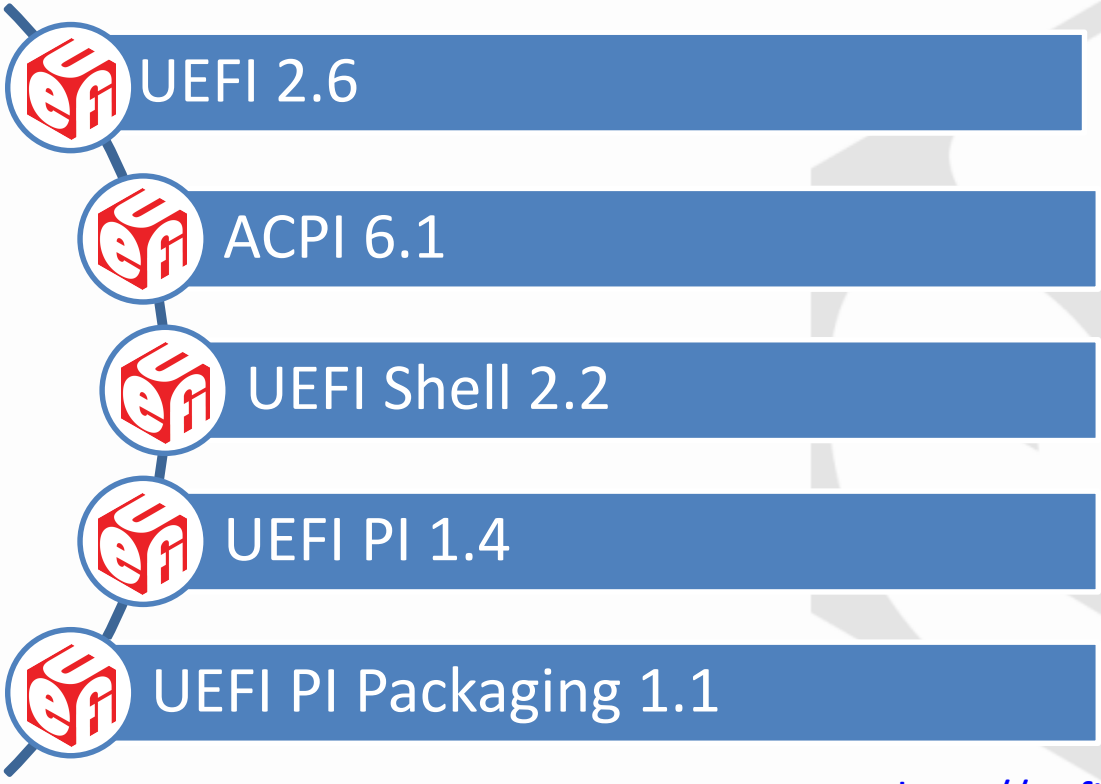
Agenda

- Where are we now
- Where are we going
- Challenges
- Questions

Section Heading

Where are we now?

Latest UEFI & ACPI Specifications

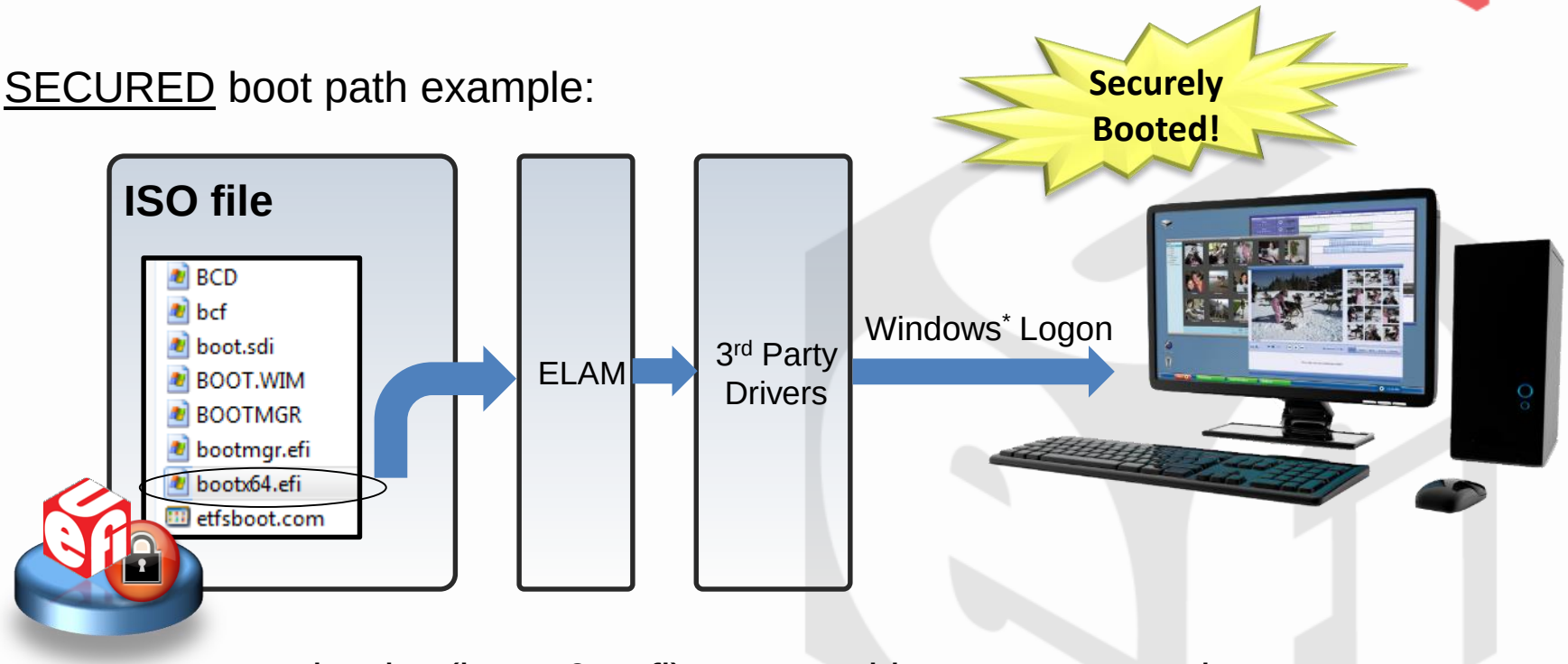


http://uefi.org/specifications

UEFI Secure Boot

https://github.com/tianocore/edk2/tree/master/SecurityPkg

SECURED boot path example:



- Boot loader (bootx64.efi) protected by UEFI secure boot
- Early Launch Anti-Malware (ELAM) protected by Boot loader
- Rootkit malware can no longer bypass anti-malware inspection
  Similar models w/ other OS's, including Linux Shim, Android/Brillo kernel flinger

Customized UEFI Secure boot

https://github.com/tianocore/edk2-staging/tree/Customized-Secure-Boot

Deployment

Initial	Advanced
Platform Specific PK_{pub} Clear	Standardized solution to customize the secure boot keys
Setup Mode User Mode	Setup Mode User Mode Audit Mode Deployed Mode

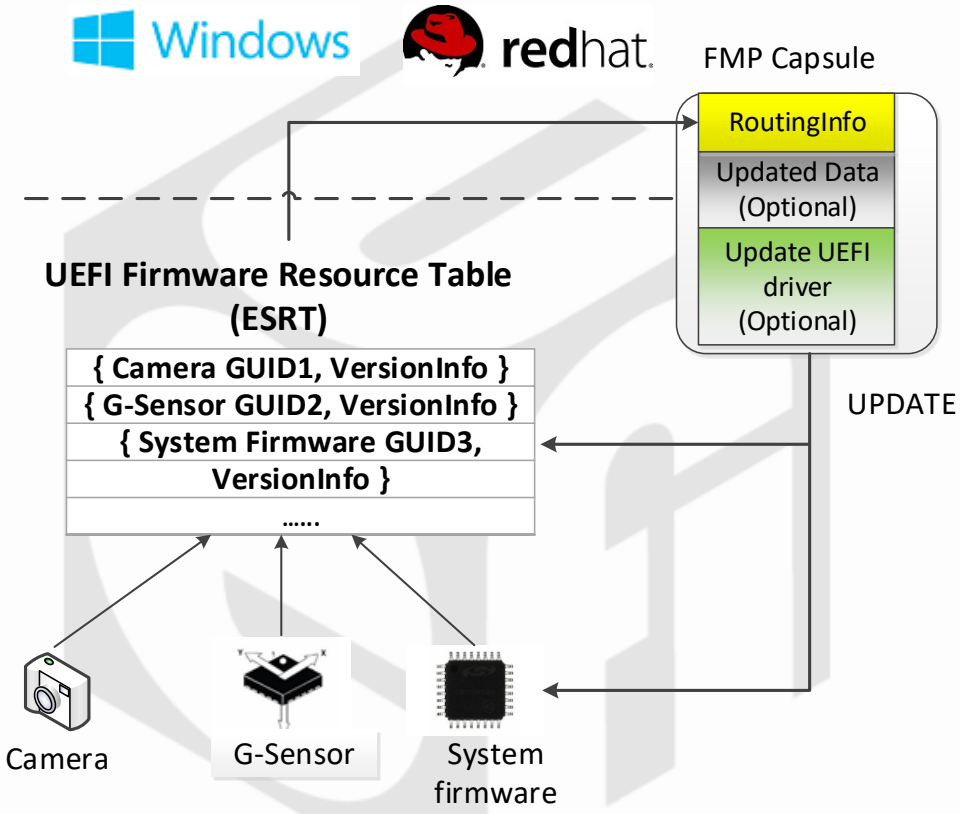
Benefits

- No specific solution → Security
- Higher utilization → Flexibility
- Verification status → Extensibility

Customized UEFI Secure Boot reduces the security risk introduced by platform specific solutions. Working w/ OS vendors on interoperability and readiness.

Secure firmware update

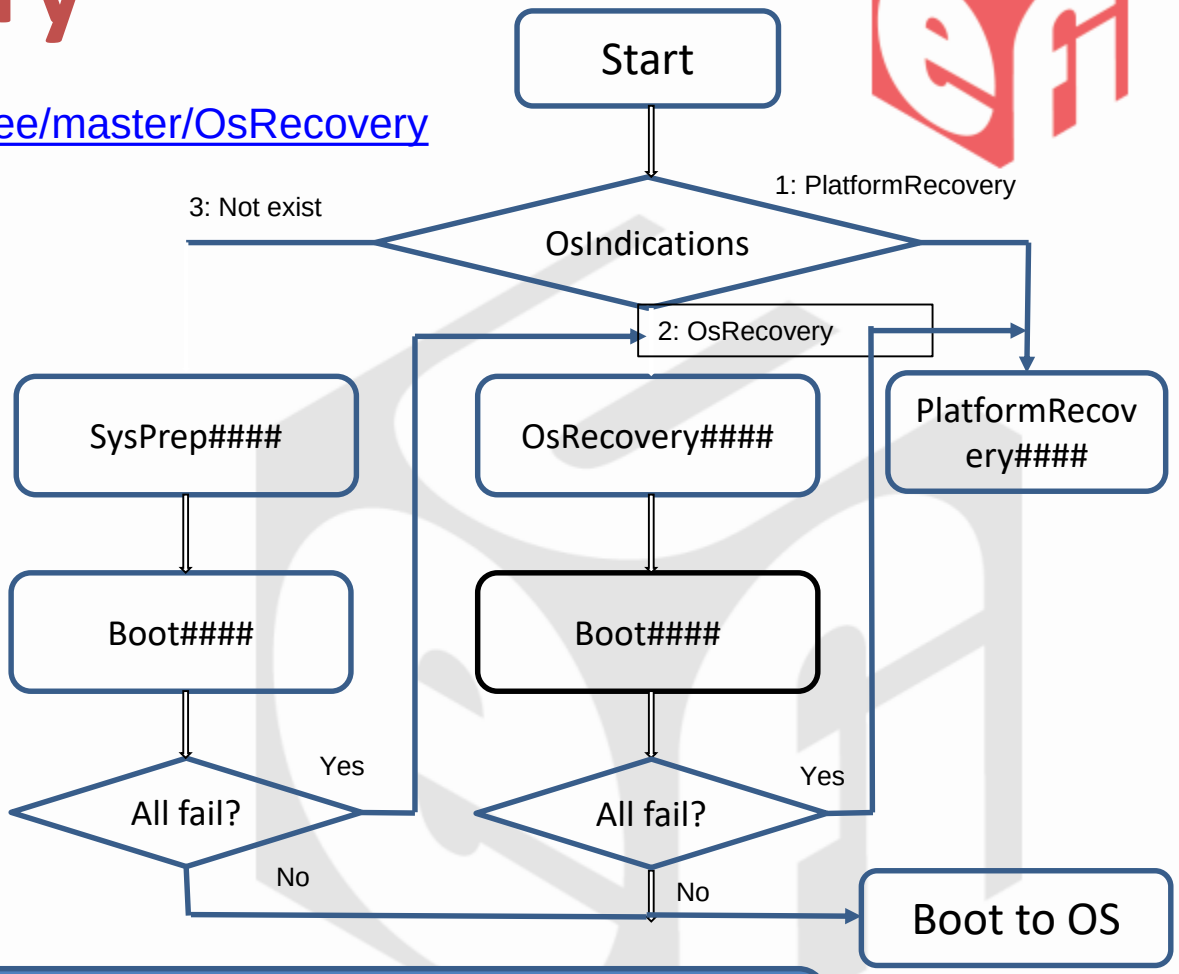
- Firmware update protected by:
  - OS verify the update driver when creating capsule
  - UEFI secure boot verify capsule payload before performing update
- What's new:
  - ESRT
  - FMPv3
  - FMP capsule



Boot recovery

https://github.com/UEFI/uefiproto/tree/master/OsRecovery

- What's new
  - OS defined recovery
  - Platform defined recovery
  - Recovery policy protected by authentication
    - OsRecoveryOrder
    - dbrDefault, dbr
  - Default platform recovery supported



Security enhancements help in accelerating the system startup stage

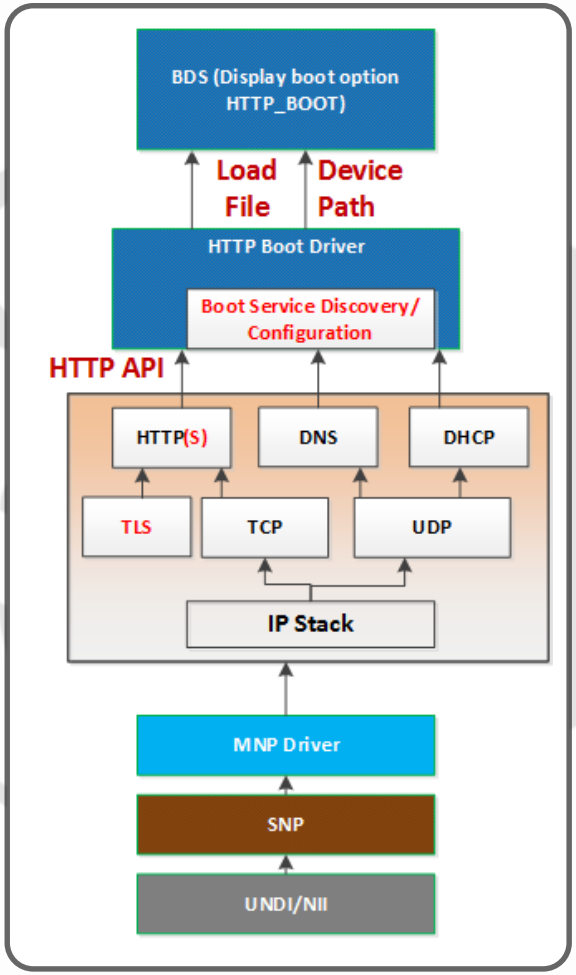
HTTP Stack

https://github.com/tianocore/edk2-staging/tree/HTTPS-TLS
https://github.com/tianocore/edk2/tree/master/NetworkPkg

New Modules

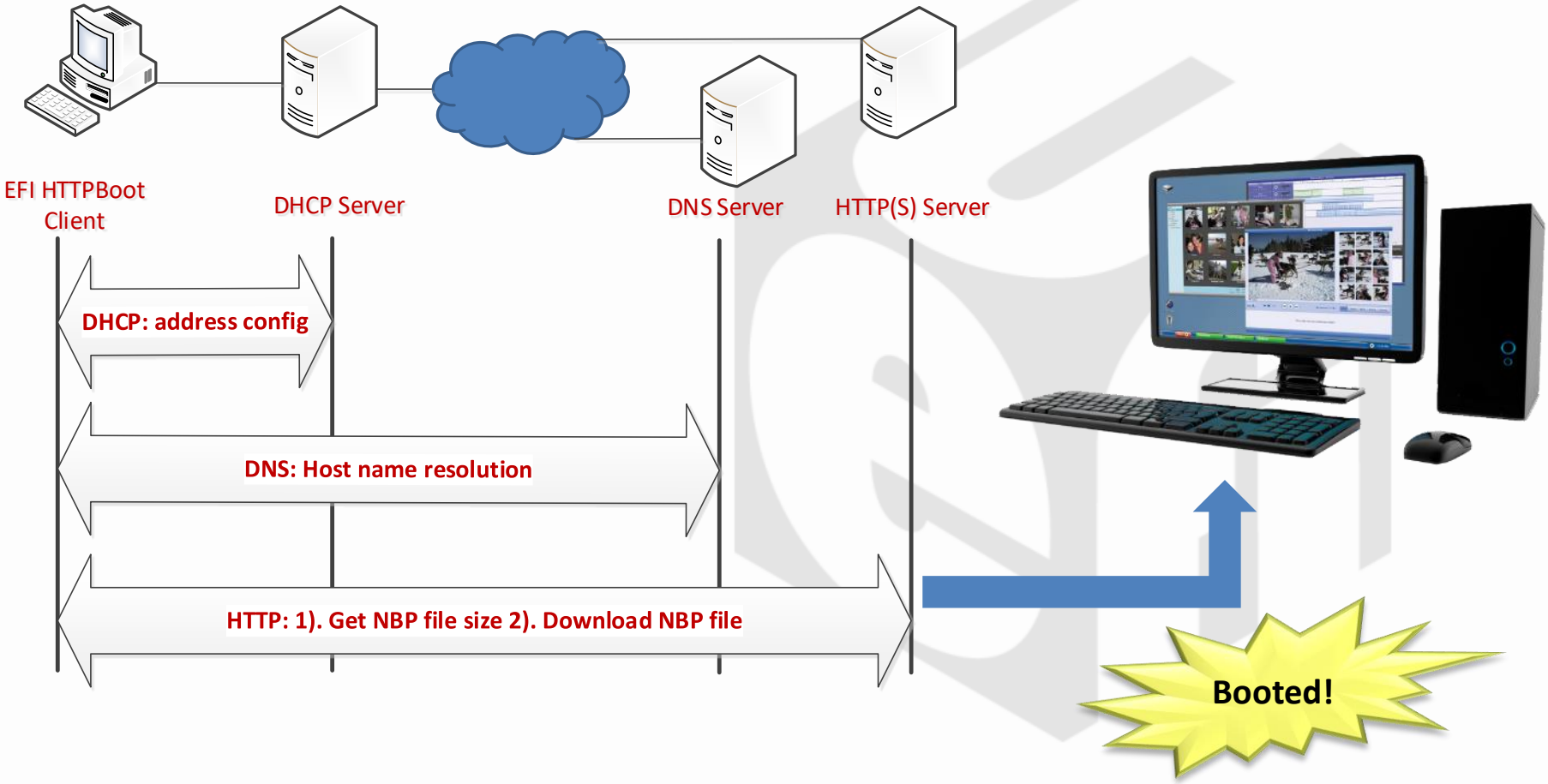
Driver	Library
HTTP Boot Driver HTTP Driver HTTP Utilities Driver TLS Driver	HTTP Library TlsLib Library OpenslTlsLib Library

- Flexible Network Deployment
- Home Environment Support
- Corporate Environment Support



HTTP-S boot

https://github.com/tianocore/edk2-staging/tree/HTTPS-TLS
https://github.com/tianocore/edk2/tree/master/NetworkPkg

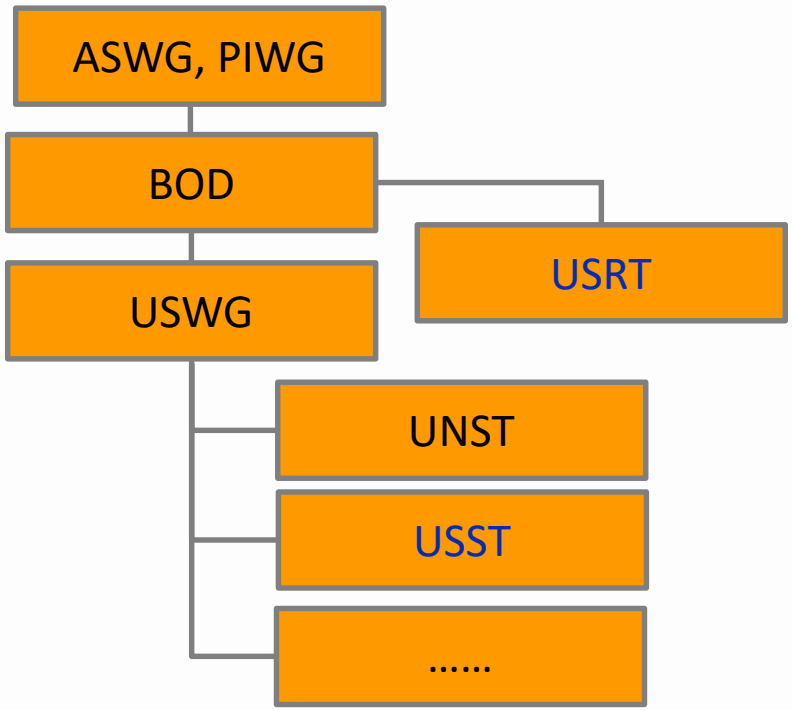


Section Heading

Where are we going?

Working Groups in the Forum

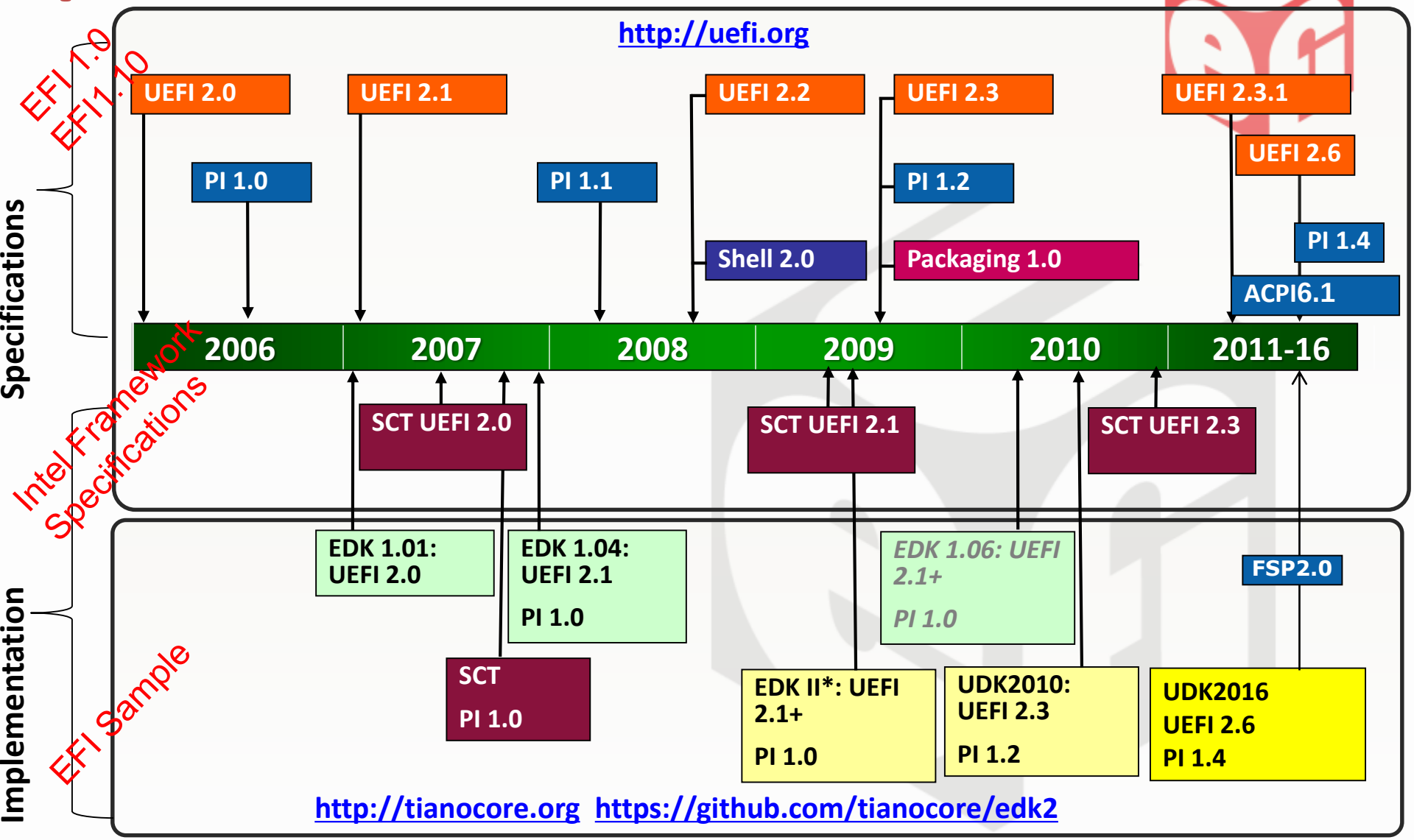
www.uefi.org



- **USWG**
  - **U**EFI **S**pecification **W**orking **G**roup
- **PIWG**
  - **P**latform **I**nitialization **W**orking **G**roup
- **ASWG**
  - **A**CPI **S**pecification **W**orking **G**roup
- **BOD**
  - **B**oard **O**f **D**irectors
- **USST**
  - **U**SWG **S**ecurity **S**ub-**t**eam
  - Chaired by Vincent Zimmer (Intel)
  - Responsible for all security related material and the team that has added security infrastructure in the UEFI spec
- **USRT**
  - UEFI Security Response Team
  - Chaired by Dick Wilkins (Phoenix)
  - Provide response to security issues.
- **UNST**
  - **U**EFI **N**etwork **S**ub-**t**eam (VZ chairs, too)
  - Evolve network boot & network security infrastructure for UEFI Specification

Note: Engaged in firmware/boot
Related WG's of Trusted Computing Group (TCG), IETF, DMTF

Specifications and code



All products, dates, and programs are based on current expectations and subject to change without notice.

How things happen today

1. Proposal (new content or errata) - closed
2. Specification creation – closed
3. Specification publication - open
4. Implementation creation – closed
5. Implementation upstream - open
6. Test creation closed
7. Test publication open
8. Bugs - Security (closed), functional (open)
    1. Goto #1

Can we do things differently?

1. Design proposal in the open
    1. Document and/or code, say as www.github.com/Random_dev_name - start w/ “EDKII_” code
2. ECR proposal
    1. Pre-specification closure write code https://github.com/UEFI/uefiproto
    2. Write rationale in ECR
3. After specification publication
    1. Publish rationale information in commit log, wiki, and/or white paper
    2. Engage w/ OS vendors and others via code written at

www.github.com/Random_dev_name and/or
https://github.com/tianocore/edk2-staging - “EDKII_” to “EFI_”
Upstream when all parties comfortable, some conformance tests

Putting it all together

- Having platforms with the features
  - Including
    - OVMF
    - Minnow
    - Galileo
    - Others…
  - UEFI Specification cannot prescribe ‘how’ to build (i.e., ‘where is my NIST 800-147 reference) but platforms can demonstrate
    - Windows Logo, Android CDD, NIST XYZ, ….
- Security Bugs
  - in EDKII code -> https://github.com/tianocore/tianocore.github.io/wiki/Reporting-Security-Issues
  - In other code and/or specification -> http://uefi.org/security

Bringing in other scenarios

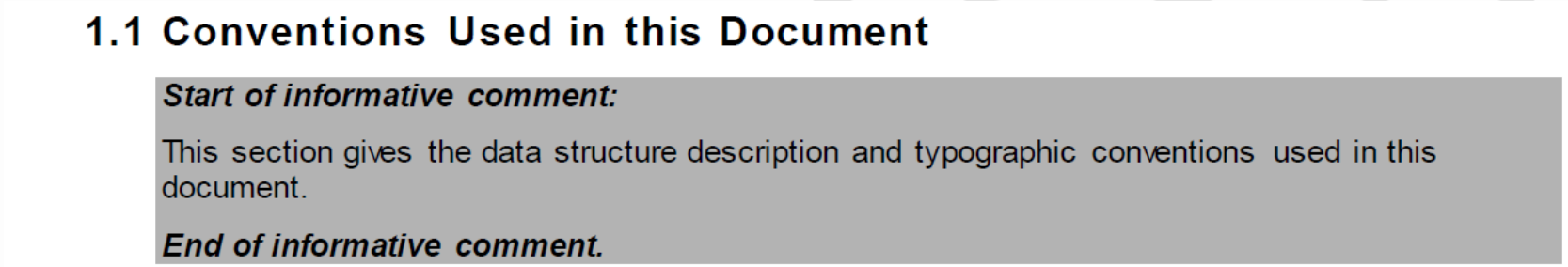
- Network based recovery
  - HTTP, Wireless, Recovery -> have OS's and platforms doing it
- Updates
  - Capsule, network, REST – harmonize payload between in-band and out of band http://www.uefi.org/sites/default/files/resources/OCPSummit2016_Towards%20a%20Firmware%20Update%20Standard.pdf and http://www.dmtf.org/sites/default/files/standards/documents/DSP0267_1.0.0a.pdf
- IPXE scenarios – evolve UEFI Shell to provide parity to IPXE scripting?

Section Heading

Challenges

Writing more down?

- Curate more documents on ‘why’ versus prescriptive ‘what’ of present specifications
- UEFI Forum != things like TCG Sample Spec



1.1 Conventions Used in this Document

Start of informative comment:

This section gives the data structure description and typographic conventions used in this document.

End of informative comment.

More to do

- Document the certificate handling & provisioning for network use cases
- Publish the informative documents
  - Enterprise deployment (in draft review)
  - Wireless design (in draft review)
  - Trust boundary document (more work to do)
  - Other…?
- Open up more defense in depth codes, touch specification where necessary
- Negative testing & assurance – https://github.com/mirrorer/afl https://github.com/chipsec/chipsec

More information

- UEFI Networking and Pre-OS Security
  http://www.intel.com/content/dam/www/public/us/en/documents/research/2011-vol15-iss-1-intel-technology-journal.pdf
- EDKII White papers
  https://github.com/tianocore/tianocore.github.io/wiki/EDK-II-white-papers
  - TPM, Variables, S3, memory profiling
- More white papers https://firmware.intel.com/share - memory map, APEI, ..
- Open source security
  https://firmware.intel.com/sites/default/files/STTS003%20-%20SF15_STTS003_100f.pdf
- UEFI Forum www.uefi.org
- EDKII https://github.com/tianocore/edk2

Getting connected

- Sign up on EDKII development mailing list
- Join the forum
- If interested in networking, joint up into UNST (send mail to me or unst-chair@uefi.org or ask admin@uefi.org to join). Same for USST (usst-chair@uefi.org)
- Reach out to me directly – Vincent.zimmer@intel.com, Vincent.zimmer@gmail.com, https://twitter.com/vincentzimmer
  - pls use earlier listed venues for bug reporting, not twitter https://twitter.com/aionescu and https://twitter.com/nikolajschlej

Thanks for attending the
UEFI US Fall Plugfest 2016

For more information on the Unified EFI Forum and UEFI Specifications, visit
http://www.uefi.org

presented by