



Microsoft's Continued Investments in the UEFI Ecosystem

UEFI 2020 Virtual Plugfest

July 15, 2020

Presented by Bret Barkelew, Matthew Carlson, & Jeremiah Cox

Presenter: Jeremiah Cox

Jeremiah is a Senior Software Engineer in Microsoft's Core UEFI team focused on enabling security features. His career has spanned from cross-platform driver development at National Instruments to over a dozen years of security development in myriad Windows security teams enabling UEFI Secure Boot, TPM 2.0, DRTM, & Secured Core PCs. His recent work includes both the Device and Manufacturing Firmware Configuration Interfaces which respectively enable secure remote configuration of UEFI by an IT administrator and enable secure re-configuration of security settings by an OEM.



Presenter: Bret Barkelew

- Viceroy of UEFI Security
- ROM Farmer and Commit Miner for Microsoft Firmware, Most Valuable Champion in Trials by Codenames Combat



Presenter: Matthew Carlson

Software Engineer II at Microsoft in Core UEFI focusing on open-source efforts.



Agenda



- Introduction
- Open Source Effort
- Code First
- Questions?





Today's Boot Landscape

- Vertically-integrated iBoot, the boot for iOS
- Vertically-integrated uBoot, the boot for Android, Chromebooks, routers, kindle, etc
- PCs (UEFI)

Committed to UEFI



- Successfully enables boot of a wide variety of operating systems, hardware, & virtual platforms
- Componentization supports the existing business realities
- Built by a community familiar with the challenges of scalability and flexibility



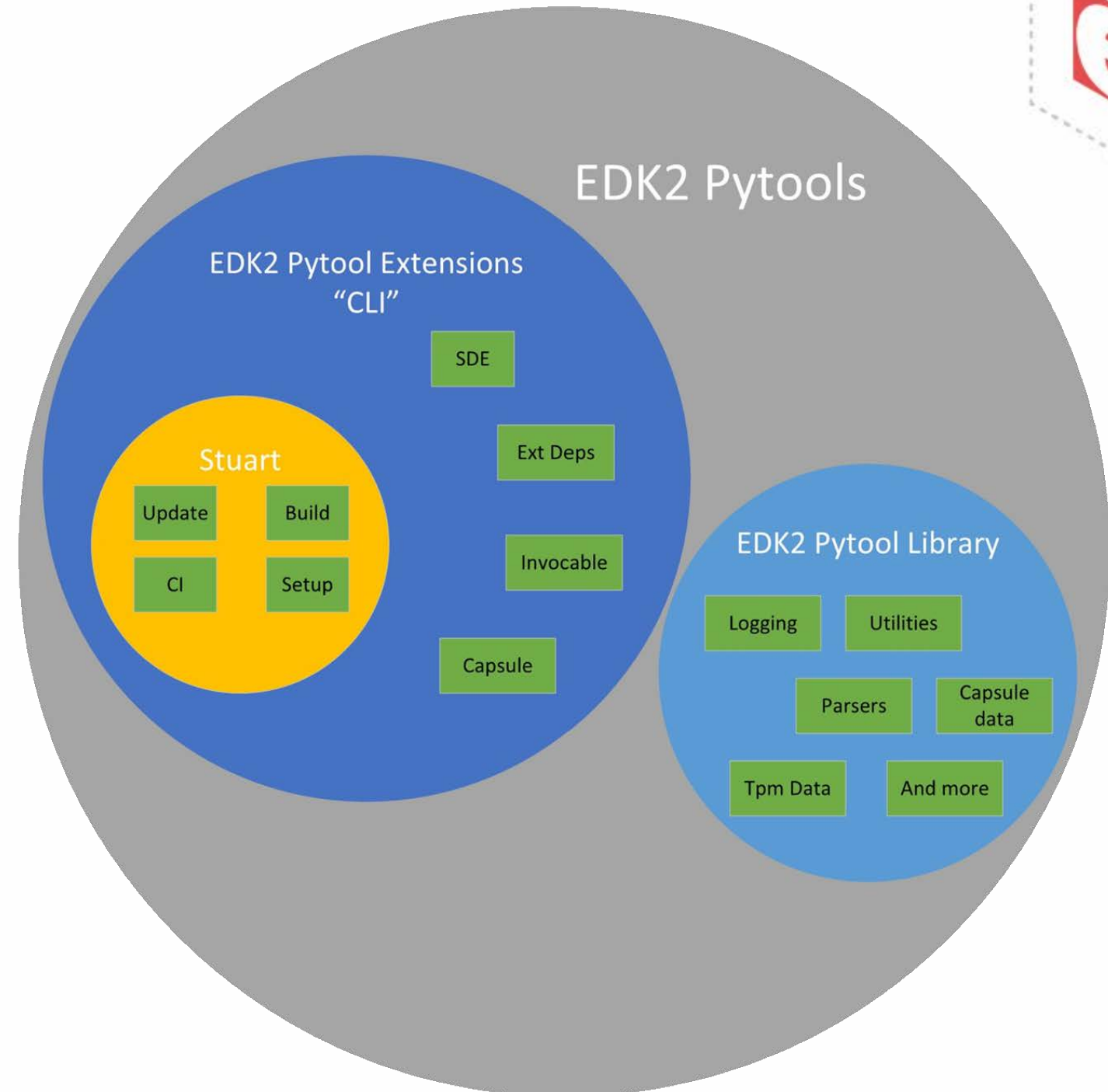


Microsoft's

Open Source Contributions

Pytools

- Started in Mu
- Library
- Extension
- Stuart

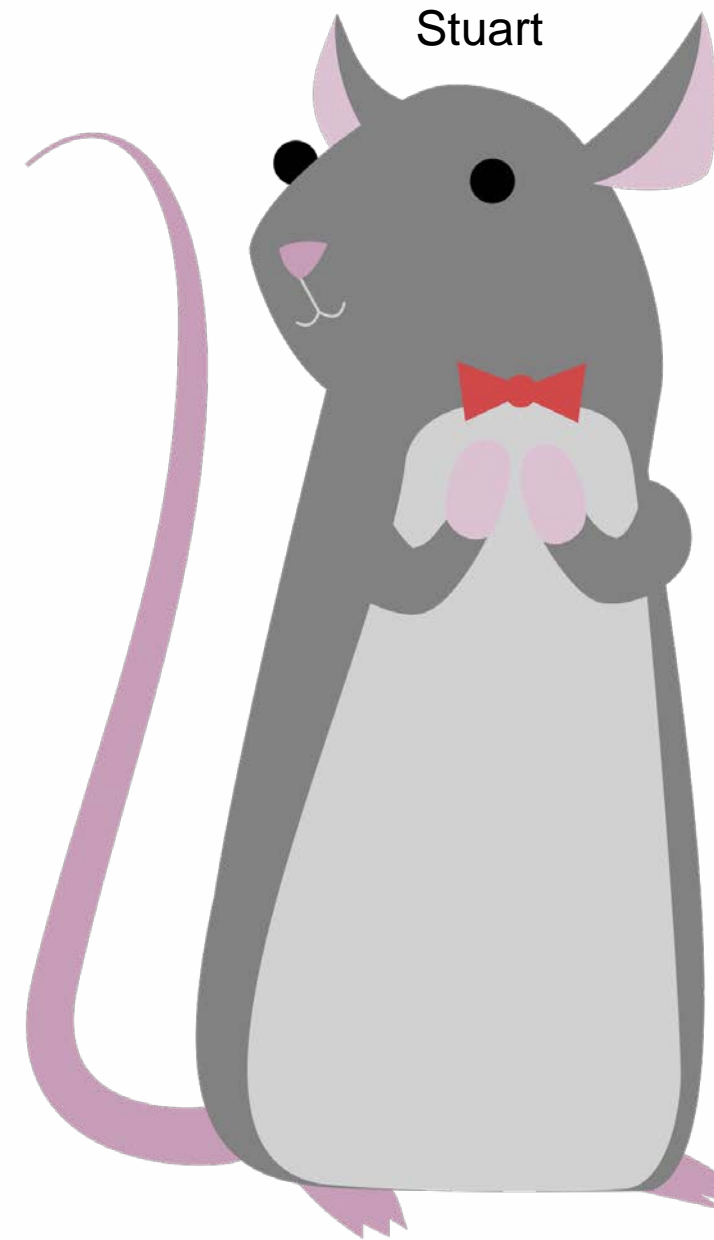


<https://github.com/tianocore/edk2-pytool-library>
<https://github.com/tianocore/edk2-pytool-extensions>

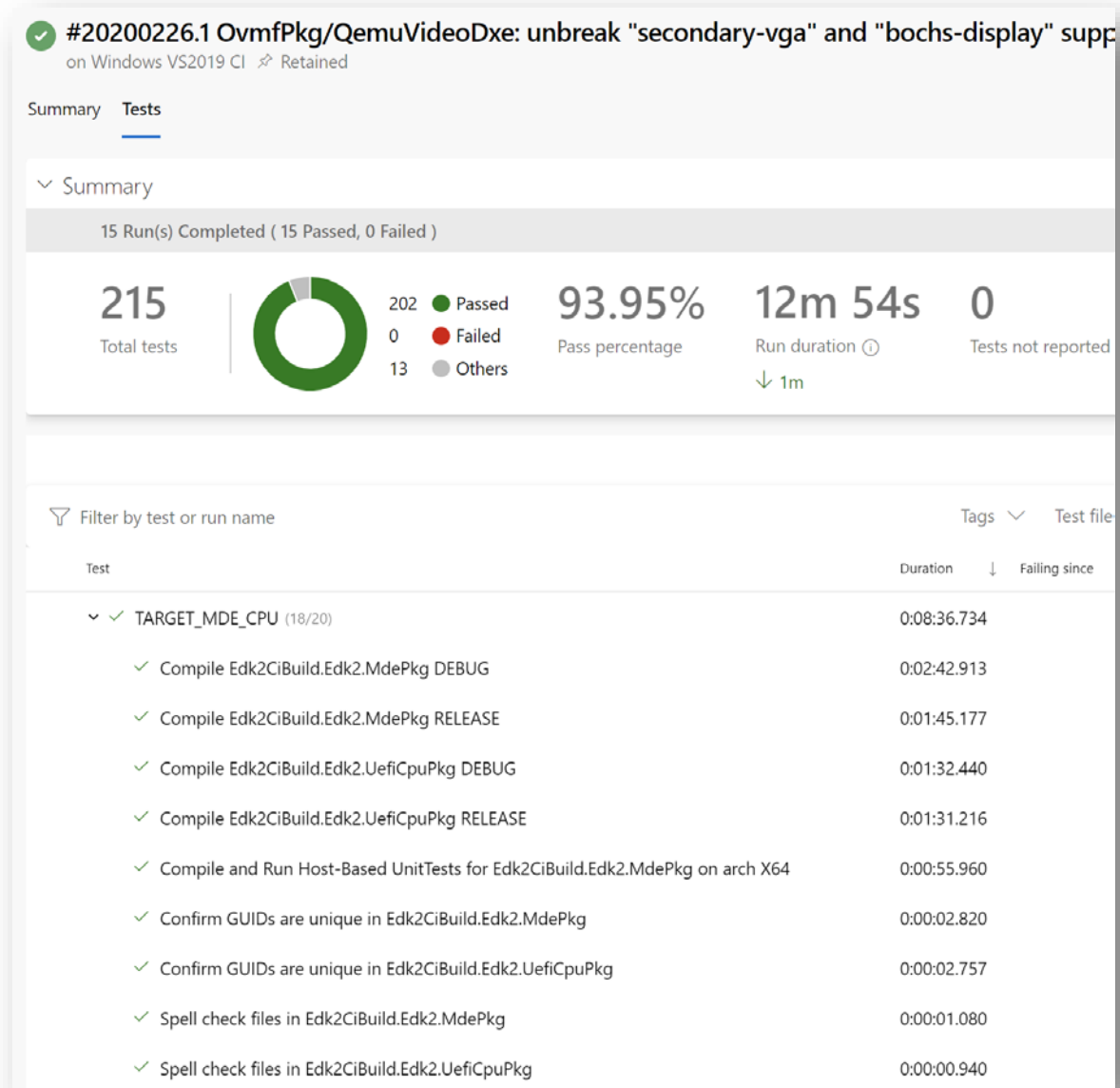


Pytools

- Full Github Process (PR, Issues, Milestones, etc)
- Invocable framework
- Extensible/Flexible
- Meant to make life easier
- Helps others contribute to community



EDK2 CI & Unit Tests



- Leverages Pytools-Library and Pytools-Extensions to have a cohesive, turnkey experience
- Plugin model enables easy test contribution or in-house development
- Several plugins already enabled, including: Compile, Host-Based Tests, and DSC Completeness checks



Microsoft's

Implementation-First Initiatives

DFCI

Device

Firmware

Configuration

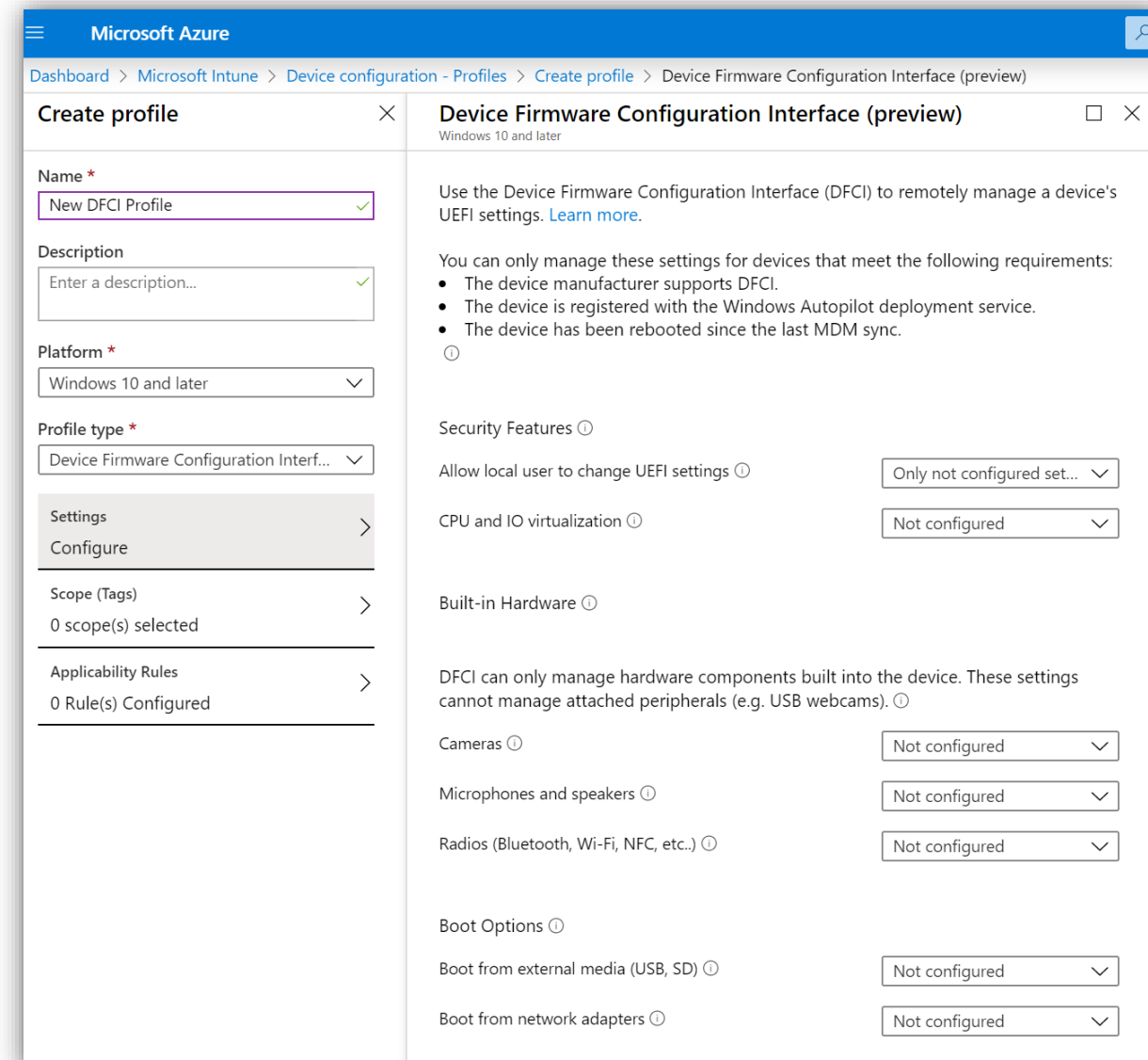
Interface



Device Firmware Configuration Interface



- Secure configuration of UEFI from Microsoft Intune in Azure
- [UEFI documentation](#) & [code in Project Mu](#)
- Available now



Manufacturer Firmware Configuration Interface (MFCI)



- Securely enable non-retail device behavior
 - E.g. remanufacturing mode
 - Strongly-authenticated, rollback protected
 - Per-device targeting (make, model, SN)



Manufacturer Firmware Configuration Interface (MFCI)



- Microsoft provides
 - Project Mu: example UEFI code & docs
 - Signing service for device manufacturers
- https://github.com/microsoft/mu_plus/tree/release/202002/MfciPkg



Variable Policy

- It's like VariableLock, but WAY more complicated
- Code-first approach that is currently in review.
 - Functionality approved and will land within the next stable iteration of EDK2
- Following the proposed process of: RFC 
Code  Spec/Standard

Protected Runtime Mechanism (PRM)



- Joint firmware and OS feature targeted at moving a class of modules from SMM to an OS runtime environment.
- Initial POC developed by Intel and discussed at past UEFI Plugfests
 - https://uefi.org/sites/default/files/resources/8_Sarathy_Intel_case%20study%20smm%20alternatives.pdf
- Current iteration developed by Microsoft and Intel.
- Open sourced and available to any collaborators
- Published to EDK2-Staging:
 - Documentation, samples, and technical details available
 - <https://github.com/tianocore/edk2-staging/tree/PlatformRuntimeMechanism>



Deep Dive

Binary Model

Binary Model

- Packaging a driver into a binary form that can be easily included in a platform with verifiable source
- Improves developer productivity
- Improves serviceability





BaseCryptoOnProtocol

- The first attempt at a binary model
- Many, many iterations internally
- Made it into edk2 Feb 7, 2020
- Harder than you might think
- ~100-200kb DXE savings (compressed)
- 3-5 minutes of build time saved



Where To Store

- NuGet
- Azure DevOps Artifacts
- Github Releases
- Github Packages
- File Mirror
- Email Archive



Lessons Learned

- Every platform is different, but they can all be supported if planned
- Provable source/versioning is crucial
- Independent serviceability is hard without crypto

Looking Forward

- Improving transparency of binaries
- Improving ease of integration into platforms
- Looking to binaryitize more components





Takeaways

Takeaways

- Committed to development experience improvements
- Investing in open source implementations
- Broadening the open source feature set





Questions?

DFCI Links



- UEFI enablement code & documentation
 - Project Mu:
https://microsoft.github.io/mu/dyn/mu_plus/DfciPkg/Docs/Scenarios/DfciScenarios/
- DFCI in Microsoft Intune
 - <https://docs.microsoft.com/en-us/intune/configuration/device-firmware-configuration-interface-windows>
- DFCI in Windows Autopilot Deployment
 - <https://docs.microsoft.com/en-us/windows/deployment/windows-autopilot/dfci-management>



Thanks for attending the UEFI 2020 Virtual Plugfest

For more information on UEFI Forum and UEFI Specifications, visit <http://www.uefi.org>

presented by

