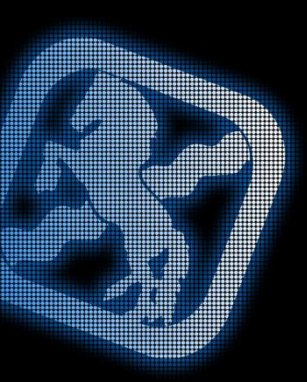
NRW Conf 2012:
FEEL THE ENERGY!

Warum UEFI das BIOS ablöst

Nicholas Dille
Microsoft MVP für Remote Desktop Services
@NicholasDille

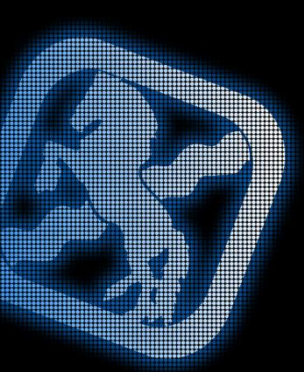


- Wer hat schon mit UEFI zu tun gehabt?
- Wer wollte schon von einer 3TB-Fastplatte booten?
- Wer hat von Secure Boot gehört?
- Wer kennt den Mann da vorne?

Wer ist Nicholas Dille?

- IT-Architekt bei sepago
  - Strategieberatung
  - Technische Konzeption
- Kernkompetenzen
  - Zentralisierung
  - Anwendungsbereitstellung
  - Kapazitätsmanagement

- Microsoft MVP für Remote Desktop Services (RDS)
- Blog: http://blogs.sepago.de/nicholas, Twitter: @NicholasDille

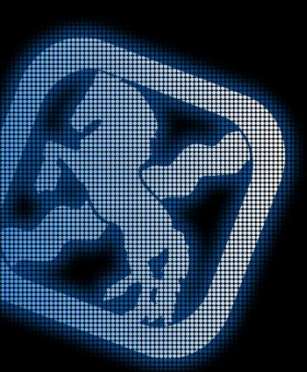


Etymologie von BIOS

- **BIOS,** Basic Input / Output System
- **BIOS,** Built-In Operating System (LINFO)

- **Bi·os,** *der; -, keine Mehrzahl;* das Leben, die belebte Welt als Teil des Kosmos (Langenscheidt Fremdwörterbuch)

- „Die Übereinstimmung mit dem altgriechischen Wort βίος […] ist eine Anspielung darauf, dass einem Computer mit dieser so genannten Software quasi ein Leben eingehaucht wird.“ (Wikipedia)



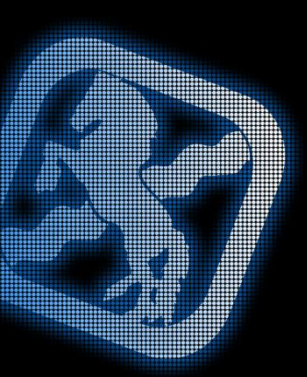
Etymologie von UEFI

- **UEFI,** Unified Extensible Firmware Interface
- Keine Anspielung
- Mein Kosename: ÜFI

BIOS-Bootvorgang

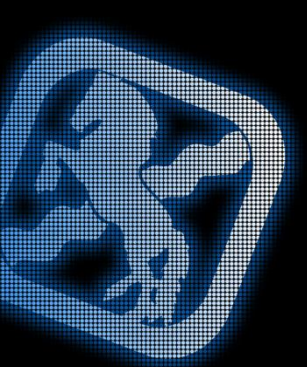
1. Power On Self Test (POST)
   a) Prozessor startet im Real Mode (16-Bit mit 20-Bit-Adressen)
   b) Hardwareinitialisierung und –diagnose (nur bei Kaltstart)
   c) Initialisierung von Firmware in Erweiterungskarten (Option PROMS)
2. Boot Loader
   a) Durchlaufen der Speichergeräte in der Bootreihenfolge
   b) Sprung in ersten Sektor zum Master Boot Record
3. Master Boot Record (MBR)

Ein Master Boot Record (MBR) ist genau einen Sektor lang – also 512 Bytes. Darin enthalten ist der Bootstrap Code (446 Bytes) und die Partitionstabelle (4x 16 Bytes) und die Bootsignatur 55AAh (2 Bytes).



BIOS-Bootvorgang

1. Power On Self Test (POST)
   a) Prozessor startet im Real Mode (16-Bit mit 20-Bit-Adressen)
   b) Hardwareinitialisierung und –diagnose (nur bei Kaltstart)
   c) Initialisierung von Firmware in Erweiterungskarten (Option PROMS)
2. Boot Loader
   a) Durchlaufen der Speichergeräte in der Bootreihenfolge
   b) Sprung in ersten Sektor zum Master Boot Record
3. Master Boot Record (MBR)
   a) MBR sucht aktive Partition
   b) Sprung in deren ersten Sektor zum Volume Boot Record
4. Volume Boot Record (VBR)
   a) Läd Boot Loader des Betriebssystems



Was stimmt nicht mit dem BIOS?

- Nutzung des MBR
  - Verlassen auf feste Positionen
  - Beschränkung des Boot-Codes auf 446 Bytes
  - Maximale Festplattengröße ist 2 TiB

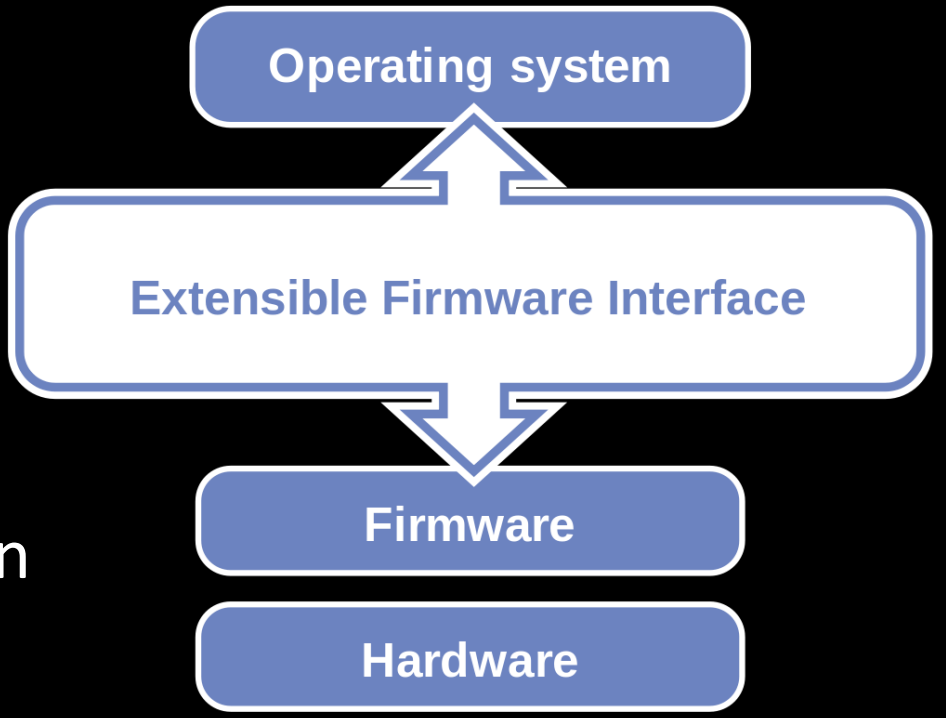
    4 Bytes (32 Bit) zur Adressierung von Sektoren ergeben
    2^32 * 512 Bytes = 2 TiB ≈ 2.2 TB

  - Begrenzung auf vier Partitionen (4x 16 Bytes im MBR)

- Ausführung im Real Mode des Prozessors
  - Zugriff auf ≈1MB Arbeitsspeicher (20 Bit-Adressen)
  - Keine Unterscheidung von Code-Privilegien
  - Kein Multi-Tasking

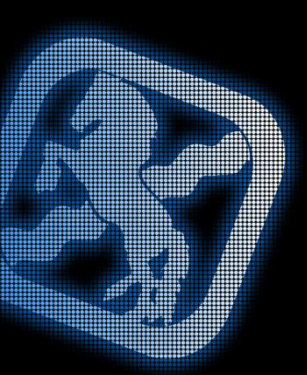
UEFI: Architektur

- Verband UEFI Forum
  - Mitglieder: AMD, American Megatrends, Apple, Dell, HP, IBM, Insyde, Intel, Lenovo, Microsoft, Pheonix



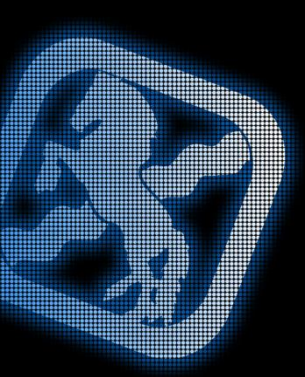
Quelle: Wikipedia UEFI

- Treiber und Anwendungen
- EFI Byte Code (EBC)
- Integrierter Boot Manager
- Compatibility Support Modul (CSM) für MBR
- Spezifikation umfasst kein POST und Setup



CSM für MBR

- Konfigurationsmöglichkeiten
  - Verfügbarkeit des CSM: nur UEFI, nur BIOS, beide
  - Reihenfolge: UEFI zuerst, BIOS zuerst
- EFI-Treiber (EfiCompatibility)
  - Verantwortlich für die Steuerung und die korrekte Hardwarekonfiguration beim Wechsel zwischen UEFI- und Legacy-Code
  - Vermutlich mithilfe des Virtual 8086 Mode
- Beschnittenes BIOS (Compatibility16)
  - Kein POST und kein Setup
  - Läuft parallel zu UEFI
  - Installiert Interrupt-Handlers und bootet wie echtes BIOS



UEFI-Bootvorgang

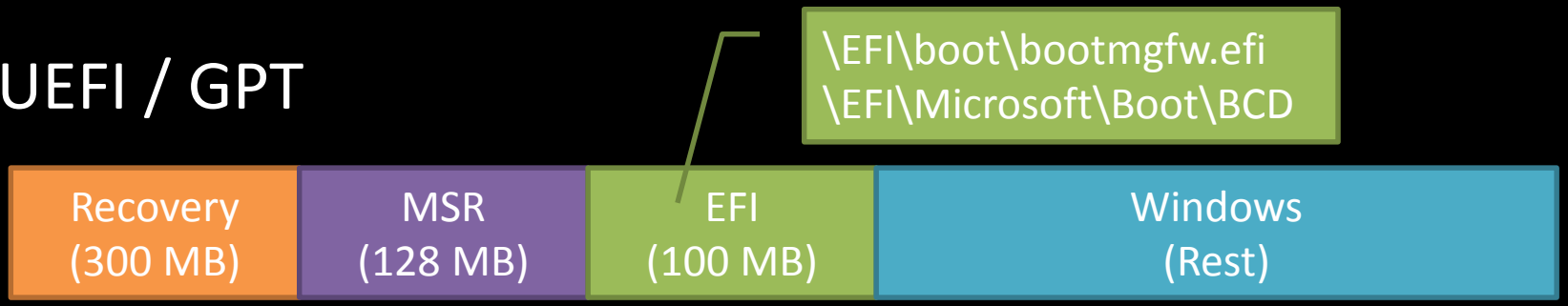
1. Power On Self Test (POST)
   a) Prozessor startet im Real Mode (16-Bit mit 20-Bit-Adressen)
   b) Hardwareinitialisierung
2. UEFI
   a) Wechseln in den Protected Mode
   b) Laden von Hardware-Treibern
3. Boot Manager
   a) OS im Non-Volatile RAM (NVRAM) registriert → zeigt ein Bootmenü
   b) Kein OS registriert → Durchlaufen der Speichergeräte für Bootmenü
   c) Suche nach Bootprogramm, z.B. \EFI\Microsoft\Boot\bootmgfw.efi

- UEFI kennt Partitionen nach dem GUID Partitioning Scheme (GPT)
- UEFI kennt das Dateisystem FAT32

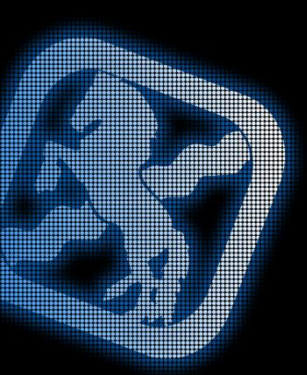
Partitionierung für Windows

- Minimale Anforderungen von Microsoft
  - Windows x64 ab Windows Vista bzw. Windows Server 2008
- BIOS / MBR

- UEFI / GPT



  - Microsoft Reserverd (MSR) als Ersatz für ungenutzte Speicherbereiche von MBR-partitionierten Festplatten



Bootbare Speichergeräte

- Optische Medien
  - BIOS: Emulation von Diskette oder Festplatte mit El Torito
  - UEFI: FAT-formatiert mit Boot Loader in `\EFI`
- USB-Sticks
  - BIOS: beliebig formatiert
  - UEFI: FAT-formatiert mit Boot Loader in `\EFI`
- Windows To Go (Windows 8)
  - BIOS braucht nur eine NTFS-Partition und funktioniert daher auch auf Wechseldatenträgern
  - UEFI benötigt zwei Partitionen
    - MBR mit FAT für EFI und BCD sowie NTFS für Windows To Go
    - Funktioniert nicht auf Wechseldatenträgern

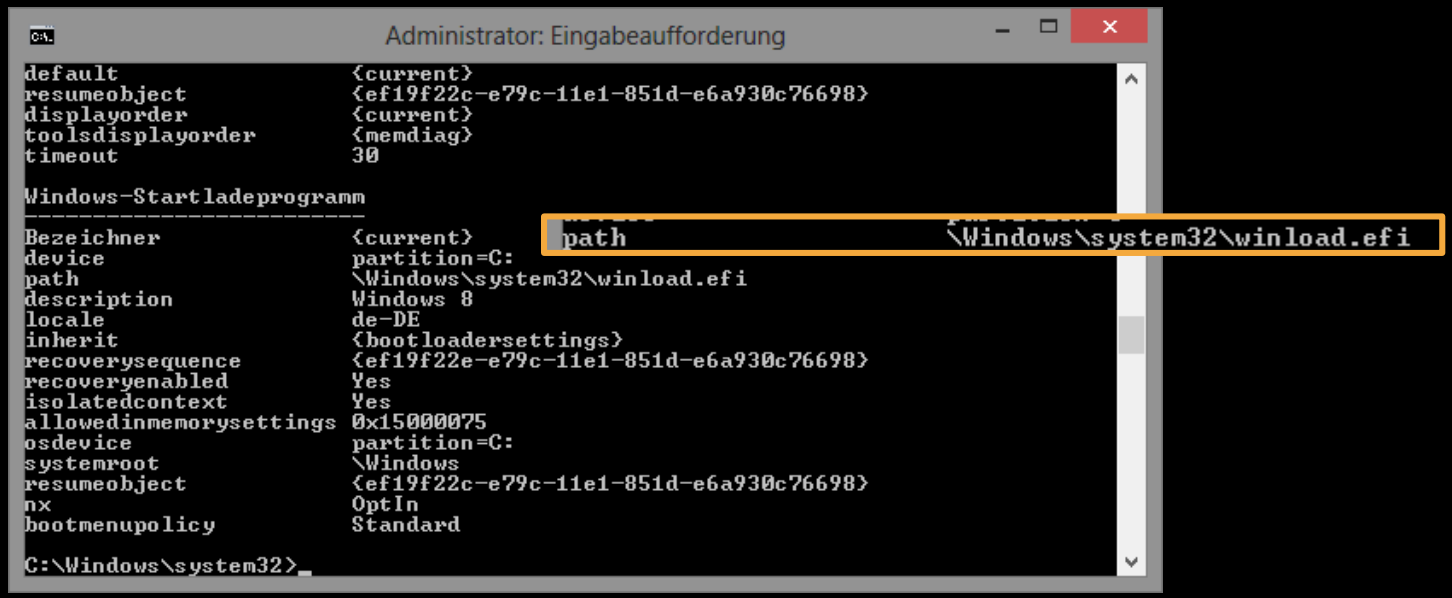
Windows-Bootvorgang

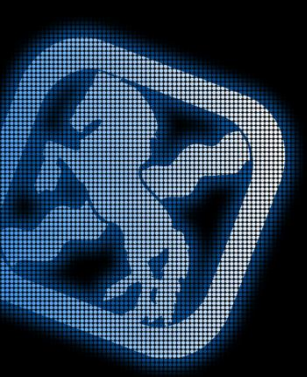
1. Bootmgr bzw. bootmgfw.efi ist ein Boot Manager
   a) Bootmgr ist PE-EXE mit 16-Bit Stub und aktiviert den Protected Mode
   b) Liest Boot Configuration Data (BCD) aus \EFI\Microsoft\Boot\BCD
2. BCD kann folgende Einträge enthalten
   a) Windows Boot Manager – Konfigurationsdaten für den Boot Manager
   b) Windows Boot Loader – Eintrag für ein OS via Winload.exe
   c) Windows Resume Loader – Fortsetzen aus einem Stromsparmodus
   d) Boot Applications – Diagnose- und Wartungswerkzeuge

   All das kann ohne Microsofts Boot Manager in EFI Byte Code implementiert werden. Es vereinfacht aber die Portierung!
3. Winload.exe bzw. Winload.efi
   a) Initialisierung von Windows und Übergabe an Kernel

Wie erkenne ich ein UEFI-System?

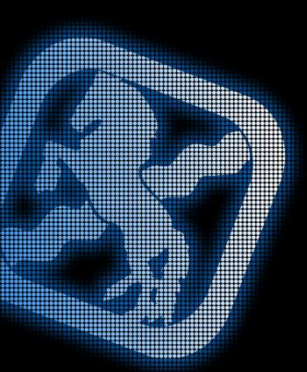
- Am Bootvorgang? Nein!
- Am grafischen Setup? Nein!
- Am laufenden Betriebssystem!
  - Bcdedit in Administrator Command Prompt
  - Eintrag {current} enthält \Windows\system32\winload.efi





Gegenüberstellung BIOS/UEFI

	BIOS	UEFI
Einführung	1981	EFI in 1998, UEFI in 2007
Prozessormodus	Real Mode (16-Bit)	Protected Mode
Verfügbarer Arbeitsspeicher	640 KB – 1 MB	Vollständig
Festplattenformat	MBR	GPT mit Protective MBR
Unterstützte Festplattengröße	2 TiB	8 ZiB
Anzahl unterstützter Partitionen	4	Unbegrenzt (128 unter Windows)
Boot-PROMS	Begrenzt durch Speicher	Keine Beschränkung
Boot-Manager	Keiner	Integriert
Grafisches Setup	Proprietäre Lösungen	Integriert
Unterstützung durch Windows	Alle	Seit Windows Vista 64-Bit bzw. Server 2008 64-Bit

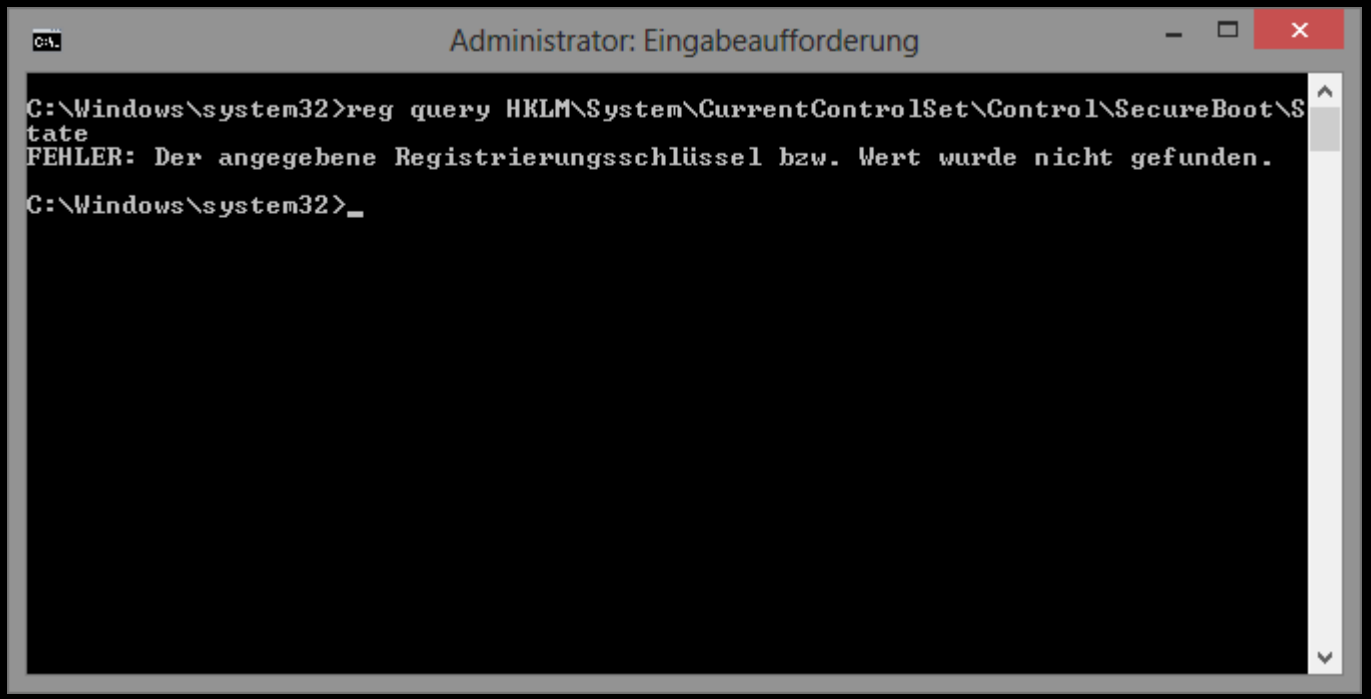


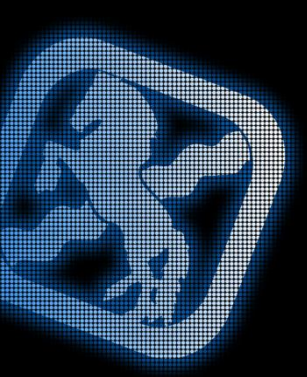
UEFI Secure Boot

- Sicherer Systemstart durch vertrauenswürdigen Code
  - Voraussetzung ist UEFI 2.3.1
  - Kein TPM notwendig
- Boot Loader müssen digital signiert sein
  - Vertrauenswürdige CAs sind in Firmware enthalten
  - Die Auswahl obliegt den Herstellern
- Windows RT erfordert Secure Boot
  - Microsoft bietet Signierungsservice für Boot Loader
  - Das notwendige Zertifikat kostet $99 für unbegrenzte Signaturen
  - Umsetzung in Fedora: Minimaler Boot Loader vor Boot Manager

Wie erkenne ich Secure Boot?

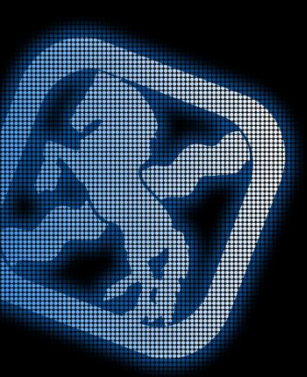
- Im Setup: Option zum Ein- und Ausschalten
- Am laufenden Betriebssystem:
  - In HKLM\System\CurrentControlSet\Control\SecureBoot\State
  - Enthält Wert UEFISecureBootEnabled (0x1 steht für eingeschaltet)





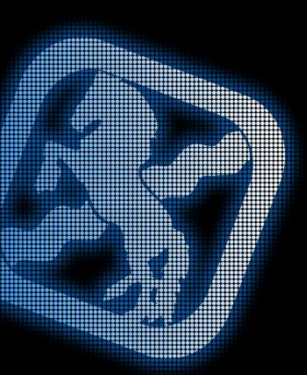
Unterstützung in aktueller Hardware

- Verquickung von Windows 8 und UEFI-Firmware
- Initiativen der Consumer-Hersteller:
  - MSI, Gigabyte, ASUS, ASRock (Listen zertifizierter Motherboards)
  - AMI und Phoenix arbeiten schon lange an UEFI-Firmware
- Business
  - Lenovo: UEFI verfügbar in allen Modellen seit 2011
  - HP: UEFI verfügbar seit 2008
  - Apple: UEFI seit Mac OS X (2006) – Kritik an der Implementierung
  - Dell und Fujitsu: unbekannt
- Alles mithilfe Intels offener Referenzimplementierung



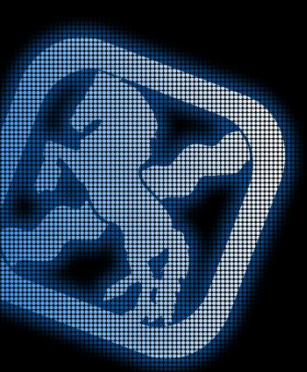
UEFI in virtuellen Maschinen testen

- Im Allgemeinen keine Unterstützung, Ausnahmen:
  - Oracle VirtualBox (VM-Einstellung)
  - VMware Workstation („firmware = efi“ in .vmx)
  - VMware ESXi
- Lösungsansatz
  - Developer‘s UEFI Environment (DUET)
  - Bootbar auf BIOS-Systemen
- Ziel
  - Generische Lösung vorzugsweise mit Diskette
  - Disketten werden immer unterstützt aber kaum genutzt
  - USB-Boot wird von Virtualisierungslösungen nicht unterstützt



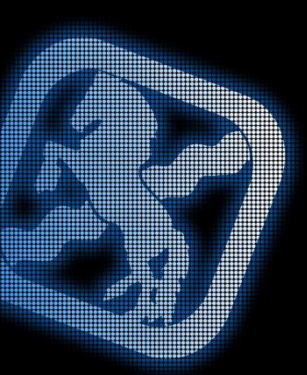
Starten von DUET in einer VM

- Von Diskette
  - Einbinden und starten
  - Funktioniert mit allen Virtualisierungslösungen
- Von USB-Stick
  - Keine Virtualisierungslösung unterstützt USB-Boot
  - Starten des PLoP Boot Manager als ISO
  - Chainloading vom USB-Stick



Intel UEFI Development Kit

- UEFI Development Kit (UDK)
  - Zweite Generation des EFI Development Kit (EDK)
  - UDK = EDK2
  - Codename Tiano
- Quelloffenes Projekt auf Sourceforge: tianocore
  - Darin enthalten ist DUET
- Binärpakete auf gitorious
  - `tianocore_uefi_duet_installer` vorkompiliert
  - `tianocore_uefi_duet_memdisk_*` nur für SYSLINUX

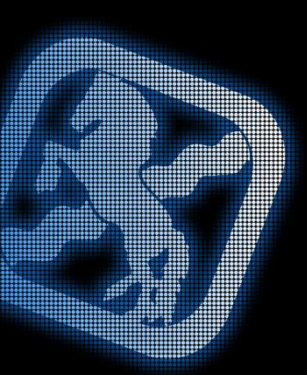


Verwenden des Binärpakets

- Minimale Voraussetzungen
- Funktioniert nur für USB-Sticks

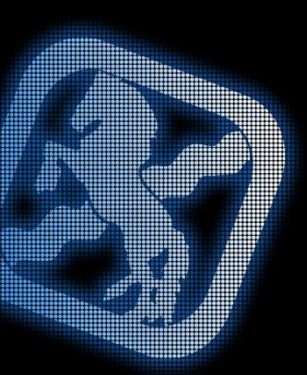
- Vorgehen
  - Formatieren mit einem Tool von HP
  - Auswerfen und wieder einstecken
  - Boot-Sektor schreiben: `CreateUSB.cmd d:`
  - Auswerfen und wieder einstecken
  - Dateien kopieren: `CreateUSB.cmd d: UDK_X64`

- Lässt sich nur umständlich in VMs booten



Build Environment für UDK

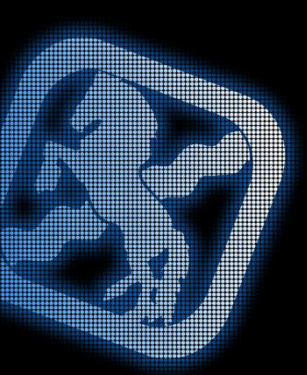
- Variante 1: Kompilieren mit Visual Studio



Ein Build Environment für UDK

- Variante 1: Kompilieren mit Visual Studio
- Variante 2: Windows Driver Development Kit

- Windows Server 2003 SP1 DDK (Download)
  - Installation nach `C:\WINDDK\3790.1830`, ansonsten (`mklink`)
- Quellen aus dem EDK2-SVN-Repository
- DDK-Konsole (Win2k3 Free x64 Build Environment)
  - Ausführen von edksetup.bat
- Anpassen von Conf\target.txt
  - TOOL_CHAIN_TAG = DDK3790xASL
  - TARGET_ARCH = X64

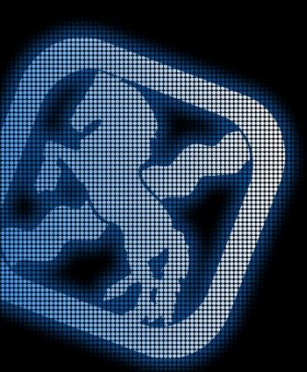


Erstellen von DUET

- Kompilieren
  - build -p DuetPkg\DuetPkgX64.dsc
  - cd DuetPkg; PostBuild.bat
- Erstellen einer Diskette
  - CreateBootDisk.bat floppy A: FAT16 X64
  - … hat aber wenig Platz
- Erstellen eines US-Sticks
  - CreateBootDisk.bat usb d: FAT32 X64
  - Lässt sich nur umständlich in VMs booten

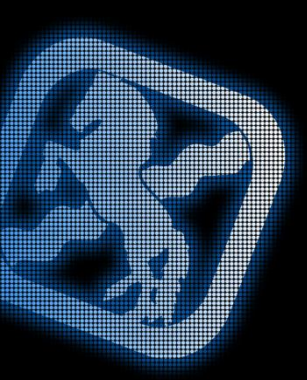
Erfolgloser DUET-Start

- … sowohl von Diskette als auch USB-Stick
- Vermutlich aufgrund einer Hardware-Inkompatibilität



Fazit

- Das BIOS ist ohne Frage veraltet
  - MBR beschränkt Festplatten auf 2TiB
  - Der CPU Real Mode beschränkt auf 1MB Arbeitsspeicher
- UEFI behebt diese Beschränkungen
- Einführung von UEFI
  - Im Business-Umfeld ist es bereits unter der Haube vorhanden
  - Consumer bekommen UEFI mit neuen Windows 8-PCs
- Tianocore
  - Quelloffene Referenzimplementierung von Intel
  - Basis für alle UEFI-Firmwares



Vielen Dank!

Fragen?