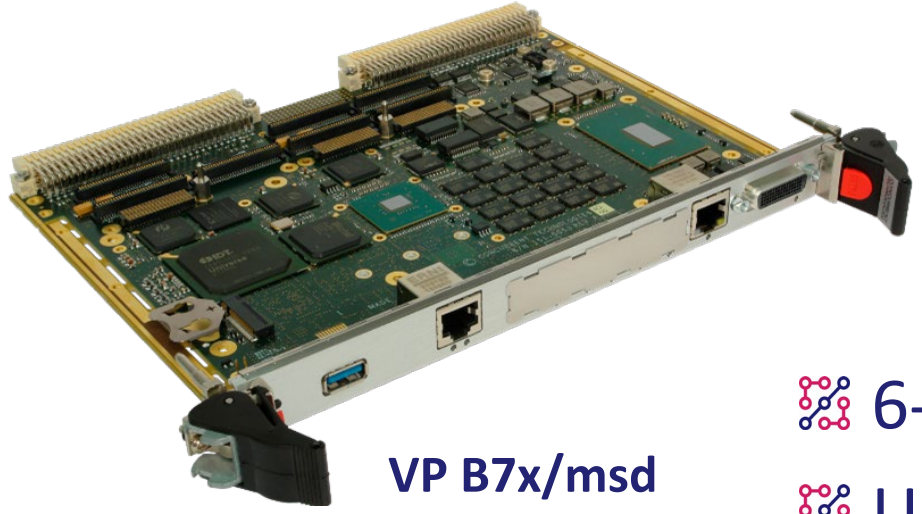
CONCURRENT TECHNOLOGIES

Back to the Future

Back to the Future?

Michael J. Fox as Marty McFly on a hoverboard in 2015 in Back To The Future Part II (1989), screen shot, January 1, 2015. (http://youtube.com). URL: https://donaldearlcollins.com/2015/01/01/back-to-my-future-forward-to-the-past/

An Example – one of our new VME boards



VP B7x/msd

Legacy

- VMEbus interface
- PMC support
- Backwards compatible rear I/O
- VME32 handles and 3-row P2 connector

Future

- 6-core Intel® Xeon® Processor
- UEFI BIOS only
- NVMe M.2 storage
- Enhanced Security

Typical Customer Needs:

- I have to run the same application code we developed x years ago
- I can’t change the Operating System (or version)
- I want to retain the option to change supplier
- I need the latest security features
- My program runs for another y years

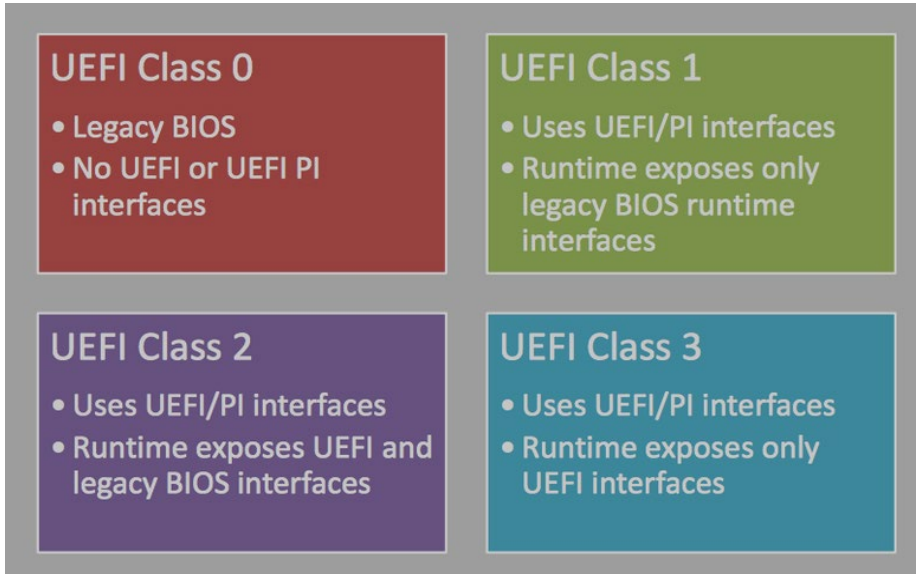
Not easy - but that’s what we’re good at

Intel Procesors

- Intel x86 processors are backwards compatible
- They can run any code developed for older processors
- For security and scalability there are now some restrictions

UEFI BIOS

- Many Intel processors now only support UEFI Class 3 BIOS (all by 2020)
  - Improves security
  - But there is no 16-bit legacy BIOS via a Compatibility Support Module (CSM)

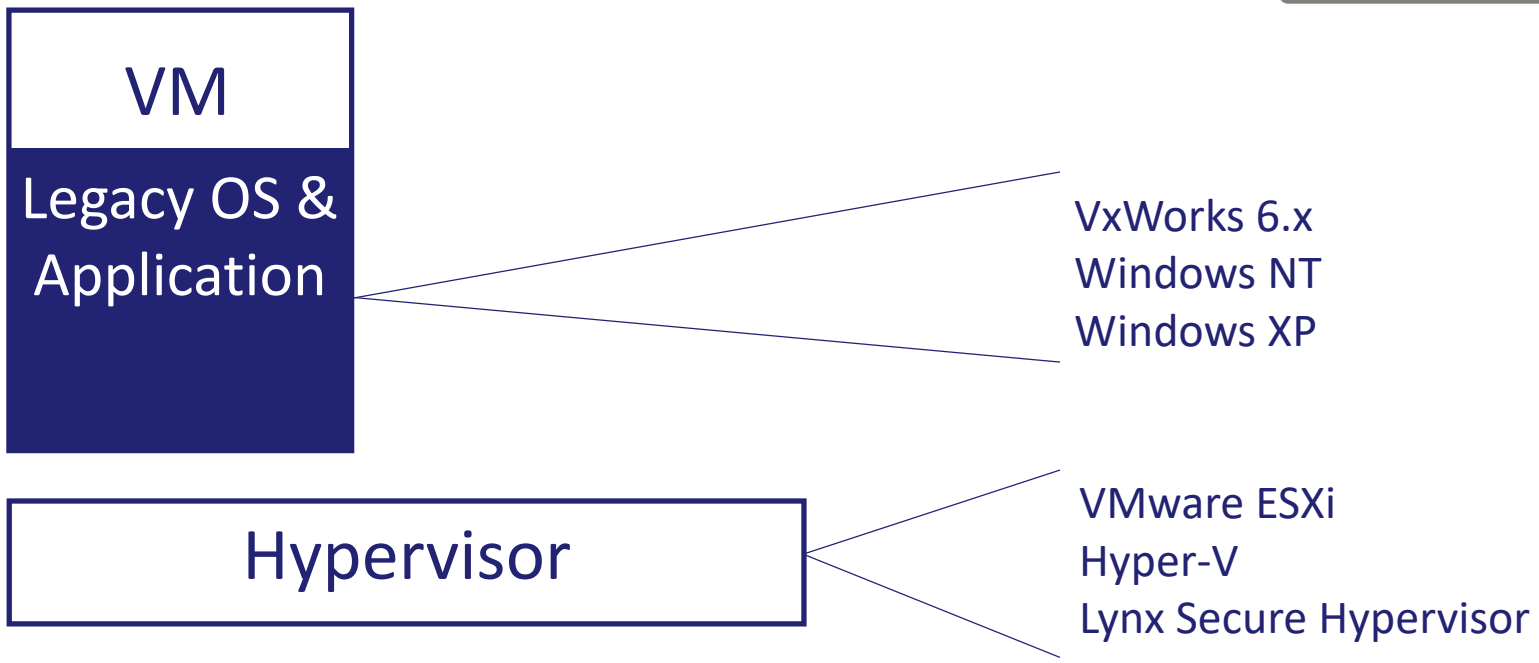


www.uefi.org

- It is not possible to natively boot a legacy Operating System including:
  - Windows 7
  - Any 32-bit OS like VxWorks 6.x, Linux 32-bit etc

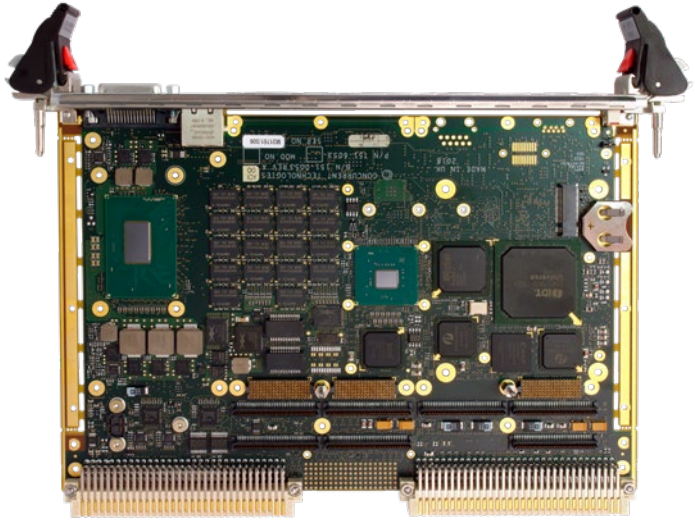
One solution

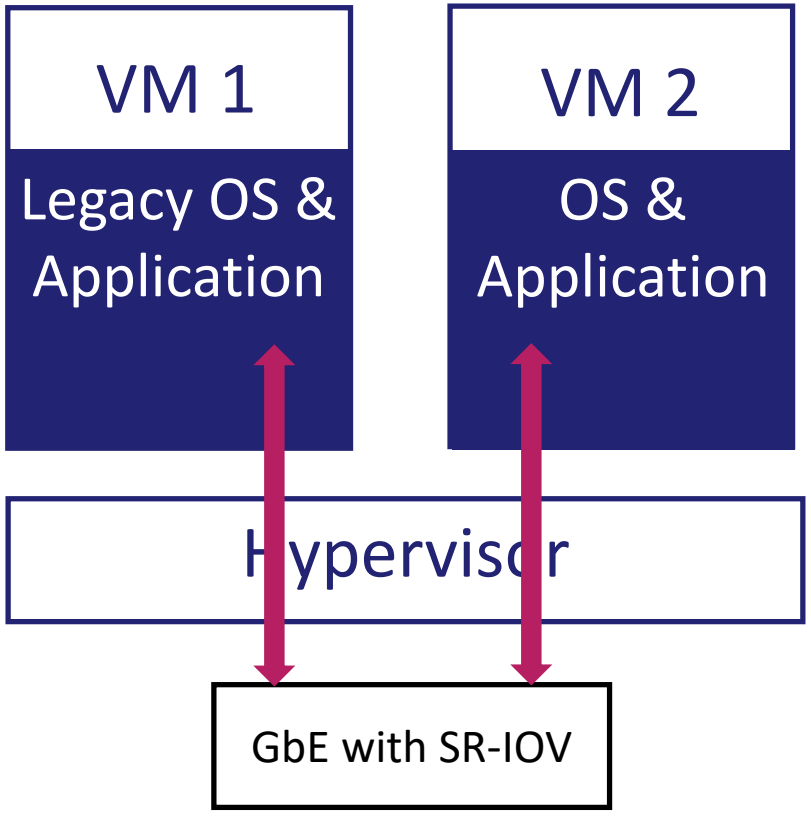
- Boot using a bare metal or hosted hypervisor
- Run the legacy OS and application in a Virtual Machine
- Has an small impact on real time performance



VP B7x/msd Example

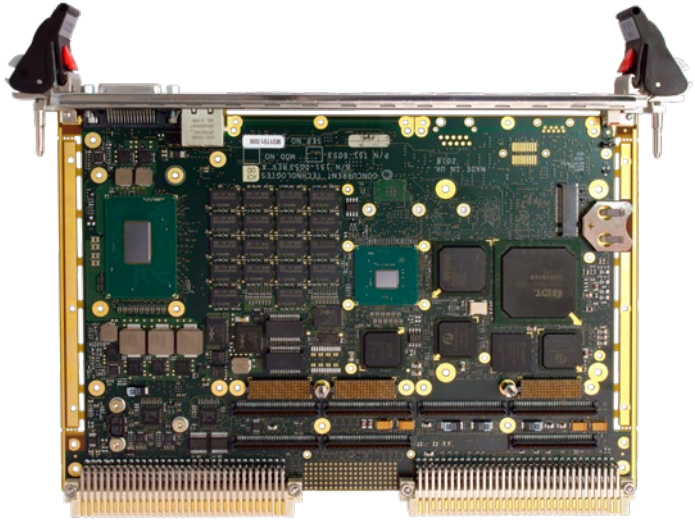
- Many interfaces support Single Root I/O Virtualization (SR-IOV)
  - One physical device appears as multiple separate physical devices
  - Each VM has the ability to access the interface

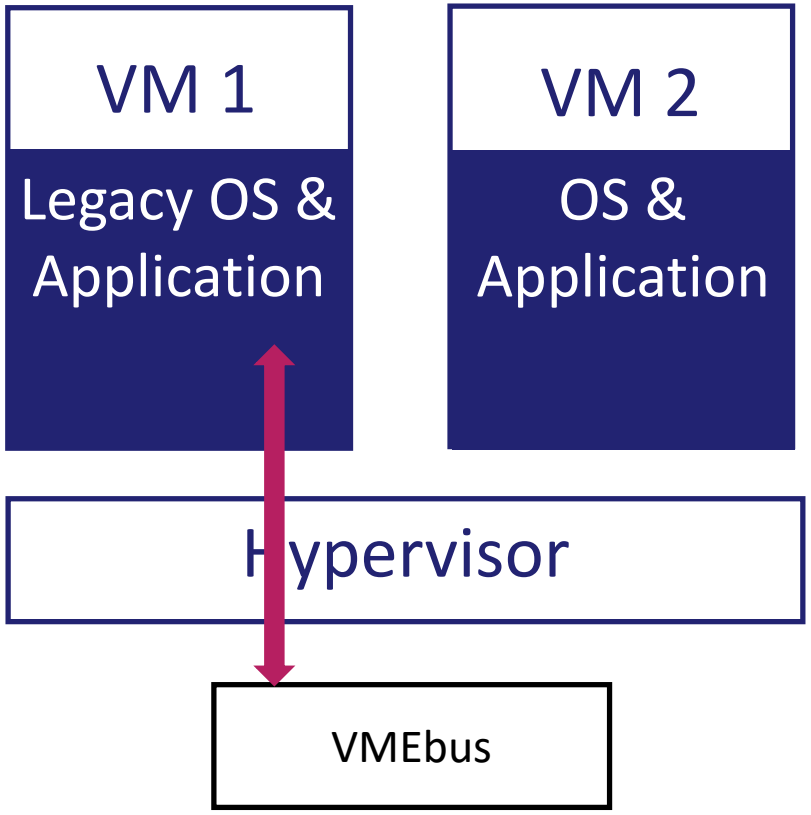




VP B7x/msd Example

- Some devices like the VMEbus interface chip are not SR-IOV capable
- It works in Direct Path I/O mode
  - Limits one Virtual Machine to access the VMEbus interface directly



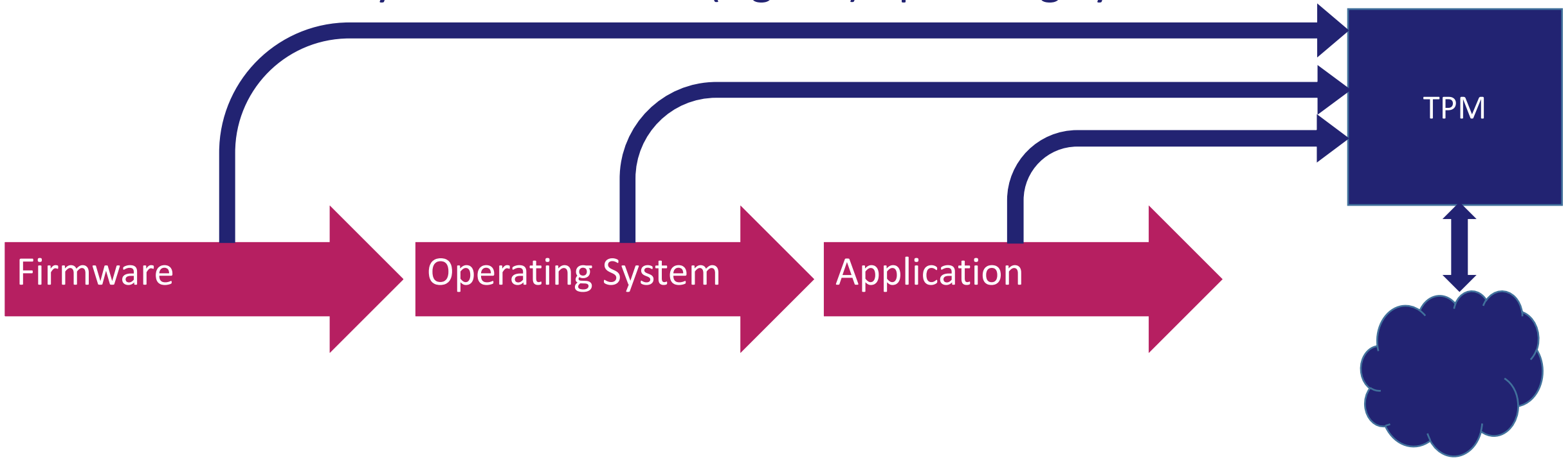


Security Timeline

1993	2002	2009	2012	2015
Microsoft Windows NT	Microsoft Windows xp	Windows 7	Windows 8	Windows 10
User access permissions	IPv6 using IPSec	TPM 1.2 Bitlocker, User Account Control, Defender	**UEFI** Secure Boot	TPM 2.0 Device Guard, Credential Guard, **Boot Guard**

Trust Levels and Attestation

- Each level can be measured
- The hashes are recorded in a TPM for remote attestation
- Secure Boot only loads a trusted (signed) operating system bootloader



Only measures what's happening

End to end Security is needed

Firmware

Secure Application

Loaded at our factory

ur customer's facility

Any break in this chain is a potential risk

Intel Boot Guard

- We now sign our firmware using a private key
- During the manufacturing process, the board is ‘fused’ to the public key
- Any attempt to boot using non-authorized firmware will fail:
  - Verified and Measured profiles implemented with Immediate Shutdown
- Maintenance updates can be done:
  - We provide a new firmware image to the customer signed with the private key
- Ensures the firmware has not been tampered with:
  - Between leaving our factory and arriving at a customer’s site
  - During the life-cycle of the product

Conclusion

- Continue the balancing act
- Part of our moral and ethical duty as COTS suppliers in this space

CONCURRENT TECHNOLOGIES

Thanks for listening