eBPF on Windows

Dave Thaler <dthaler@microsoft.com>

(e)BPF Runtime Platforms

Platform →	Linux		MacOSX		Android		FreeBSD		Windows		TockOS (embedded)	
↓ Project	**Kernel**	**User**	**Kernel**	**User**	**Kernel**	**User**	**Kernel**	**User**	**Kernel**	**User**	**Kernel**	**User**
Linux	2014				2017							
uBPF		2015										
rbpf		2017		2017						2018		
Generic eBPF	2017	2017		2017			2017	2017				
eBPF for Windows									2021			
Tock											2021	

eBPF runs in many contexts

Multiple operating systems

Kernel mode or user mode

Main processor, co-processor, SmartNIC, etc.

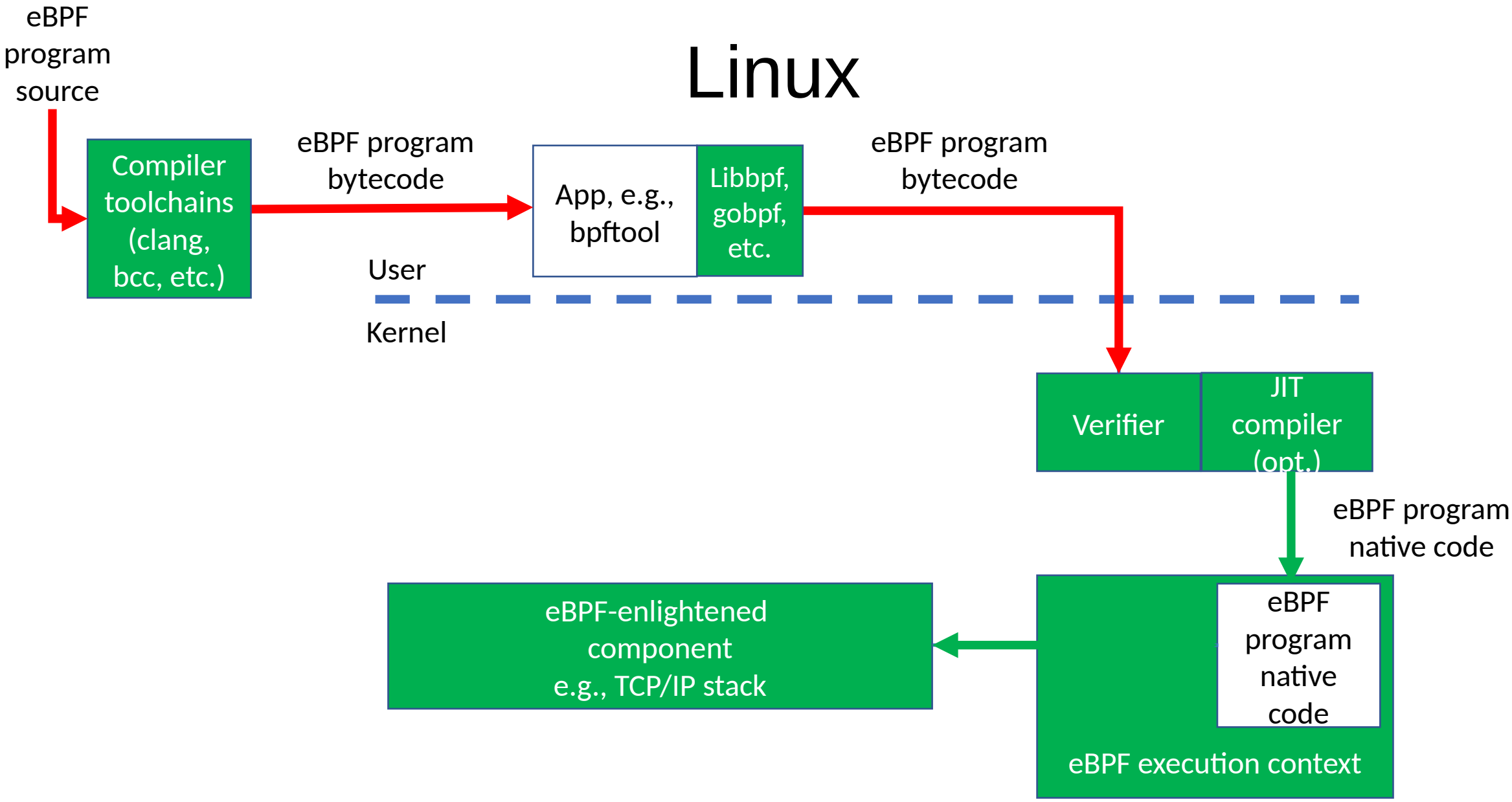
Host or Guest VM

Inside or outside containers

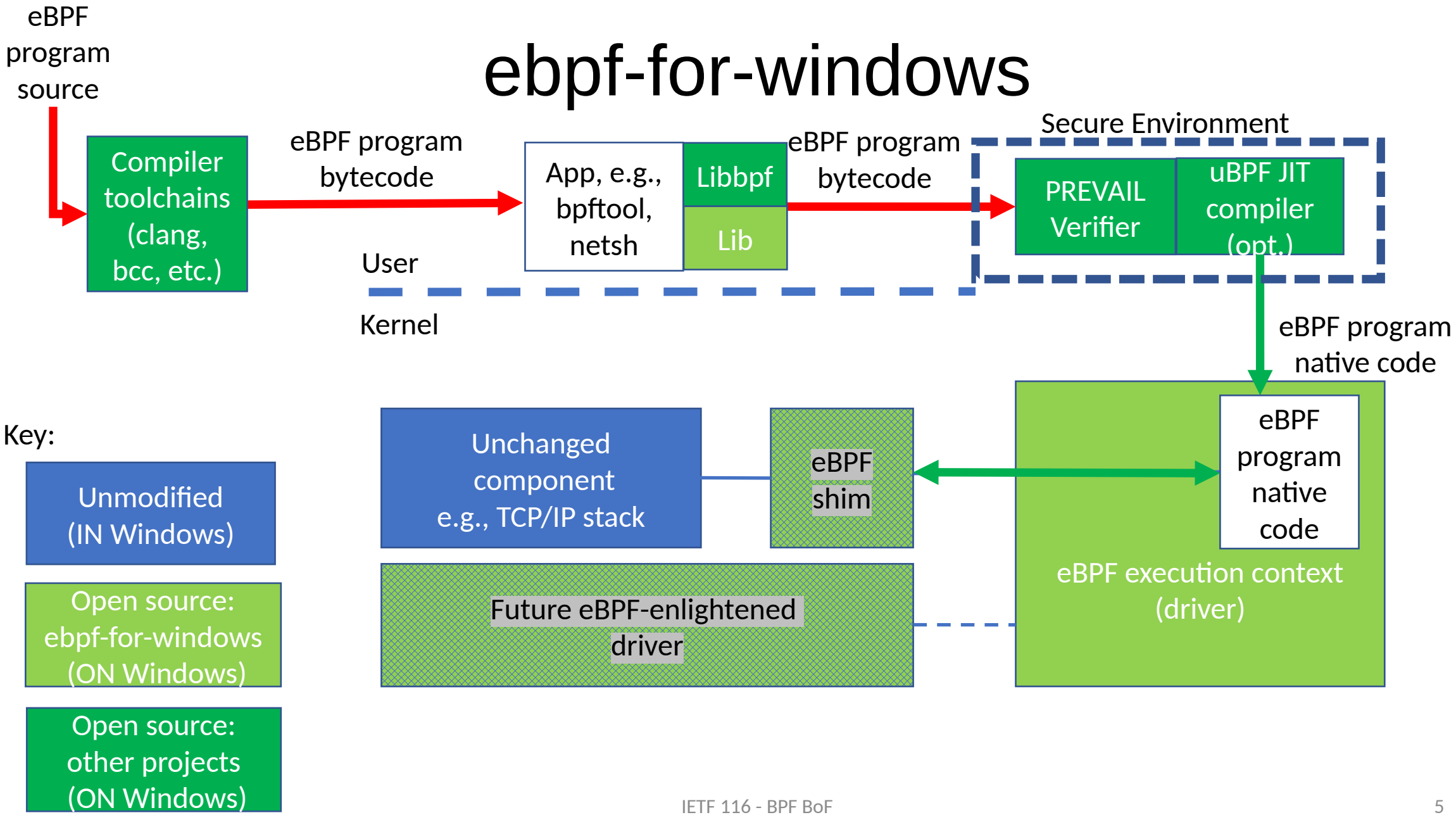
Native (JIT-compiled) code or interpreted

All using common toolchains and APIs

Linux


eBPF program source
Compiler toolchains (clang, bcc, etc.)
eBPF program bytecode
App, e.g., bpftool
Libbpf, gobpf, etc.
eBPF program bytecode
User
Kernel
Verifier
JIT compiler (opt.)
eBPF program native code
eBPF program native code
eBPF execution context
eBPF-enlightened component e.g., TCP/IP stack

ebpf-for-windows



Goal: Cross-plat BPF program source compat

- Pre-BTF ELF format + func/line info from BTF
- PREVAIL verifier: loops, helper functions, etc.
- Program types: 4 (XDP, CGROUP_SOCK_ADDR, SOCK_OPS, …)
- Map types: 13 (hash, array, queue, stack, LPM, ringbuf, …)
- Helpers: 20 (maps, tail call, ringbuf, printk, csum_diff, pid, …)

- BPF ISA conformance test suite: same tests for both Linux and Windows
  - Atomics support not done yet for PREVAIL/uBPF

- E.g.: Cilium layer-4 load balancer port uses >95% of code unmodified

Questions?