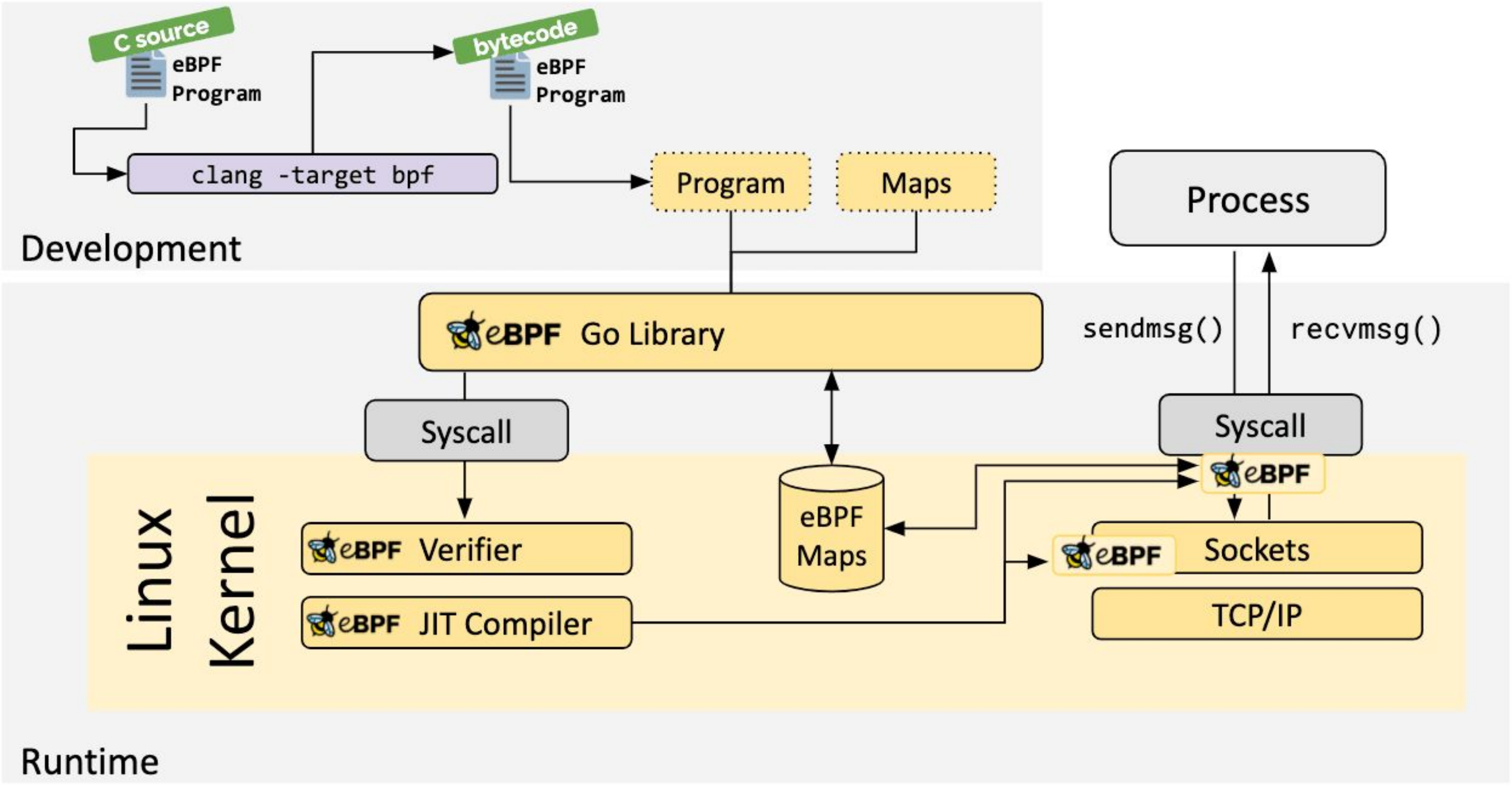
Writing Custom eBPF Programs for Observability

Sooter Saalu

What is eBPF?

- The Extended Berkeley Packet Filter is a framework for writing sandboxed programs that can be executed within your Linux kernel space without modifying the kernel itself

- It gives you a general-purpose tool that can observe, control and enhance kernel behavior


C source
eBPF Program
bytecode
eBPF Program
clang -target bpf
Program
Maps
Development
Process
eBPF Go Library
sendmsg()
recvmsg()
Syscall
Syscall
eBPF
Linux Kernel
eBPF Verifier
eBPF Maps
eBPF JIT Compiler
eBPF
Sockets
TCP/IP
Runtime

Why it's a game-changer for observability

- Kernel-level visibility
- No code instrumentation needed
- Low overhead

Real-world use cases

- Tracing syscalls =>>>> Falco example
- Network visibility =>>>> Netflix example
- Performance profiling =>>>> GroundCover example
- Security monitoring =>>>> Apple example

Demo: Packet Logger

Github link

Alternate ways

- Apart from native C ebpf programs:
  - Python —-- BCC (Single Python file with embedded C)
  - Go —-- Cilium library
  - Rust —-- Aya

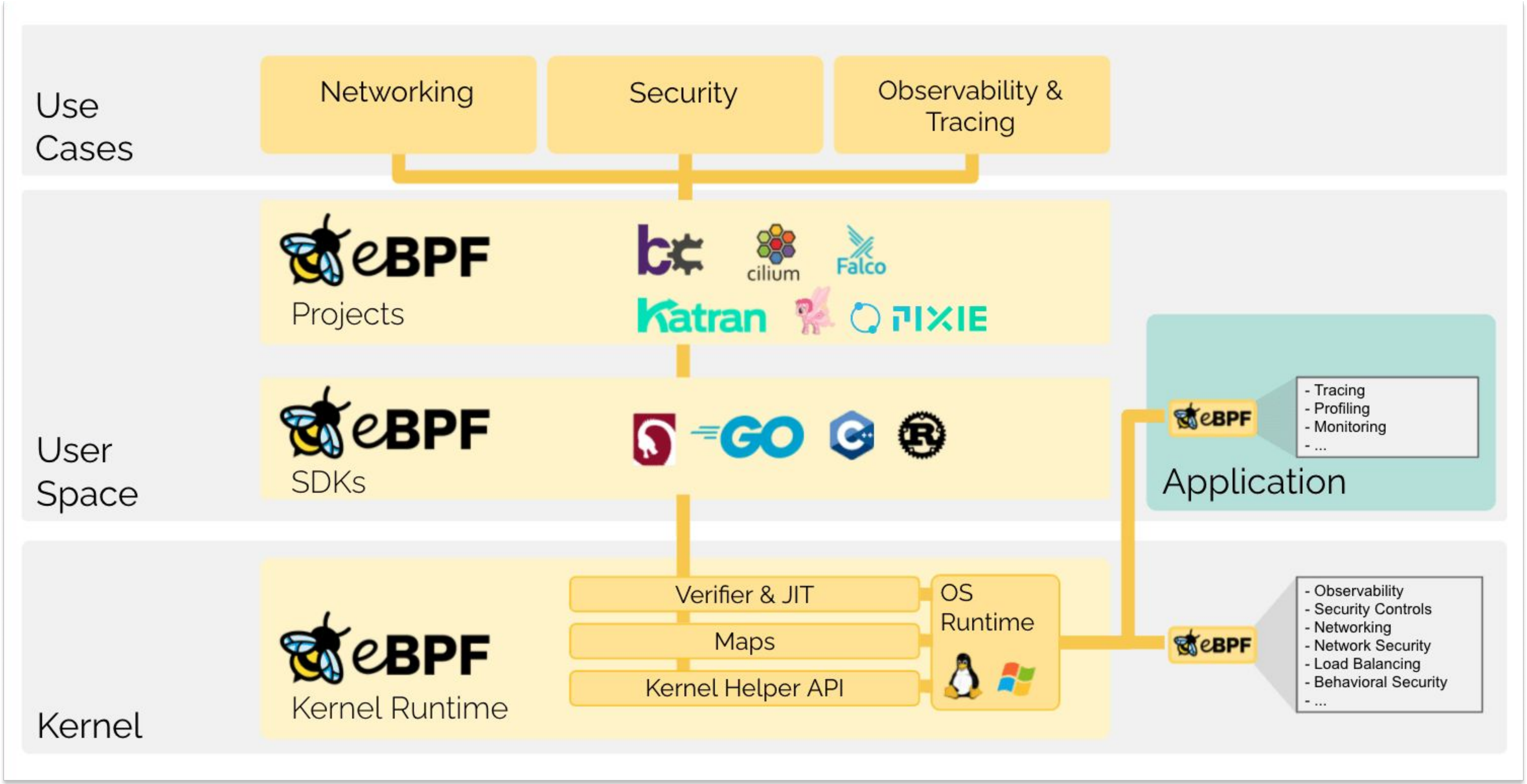
- Easier with Abstractions however Performance!! - Larger footprint, delayed execution

Demo: Tracing openat() Syscalls with BCC

Github link

Extending ebpf

- OpenTelemetry eBPF collector
- eBPF Prometheus exporter
- Grafana Beyla (eBPF based auto instrumentation tool)



Use Cases

Networking

Security

Observability & Tracing

User Space

eBPF Projects

eBPF SDKs

Application

eBPF

- Tracing
- Profiling
- Monitoring
- ...

Kernel

eBPF Kernel Runtime

Verifier & JIT

Maps

Kernel Helper API

OS Runtime

eBPF

- Observability
- Security Controls
- Networking
- Network Security
- Load Balancing
- Behavioral Security
- ...

eBPF ecosystem

- Observability and Monitoring:
  - Pixie, Parca, Pyroscope, DeepFlow
- Networking:
  - Cilium, Calico, Katran, Kyanos
- Security and Runtime Enforcement:
  - Falco, Tetragon, Tracee, Kubescape
- eBPF Program Management:
  - L3AF, bpfman, BumbleBee

More information

- eBPF documentary website
- eBPF internals (slides)
- Learning eBPF (Book)
- eBPF for FAANG

Thank You

CONF42