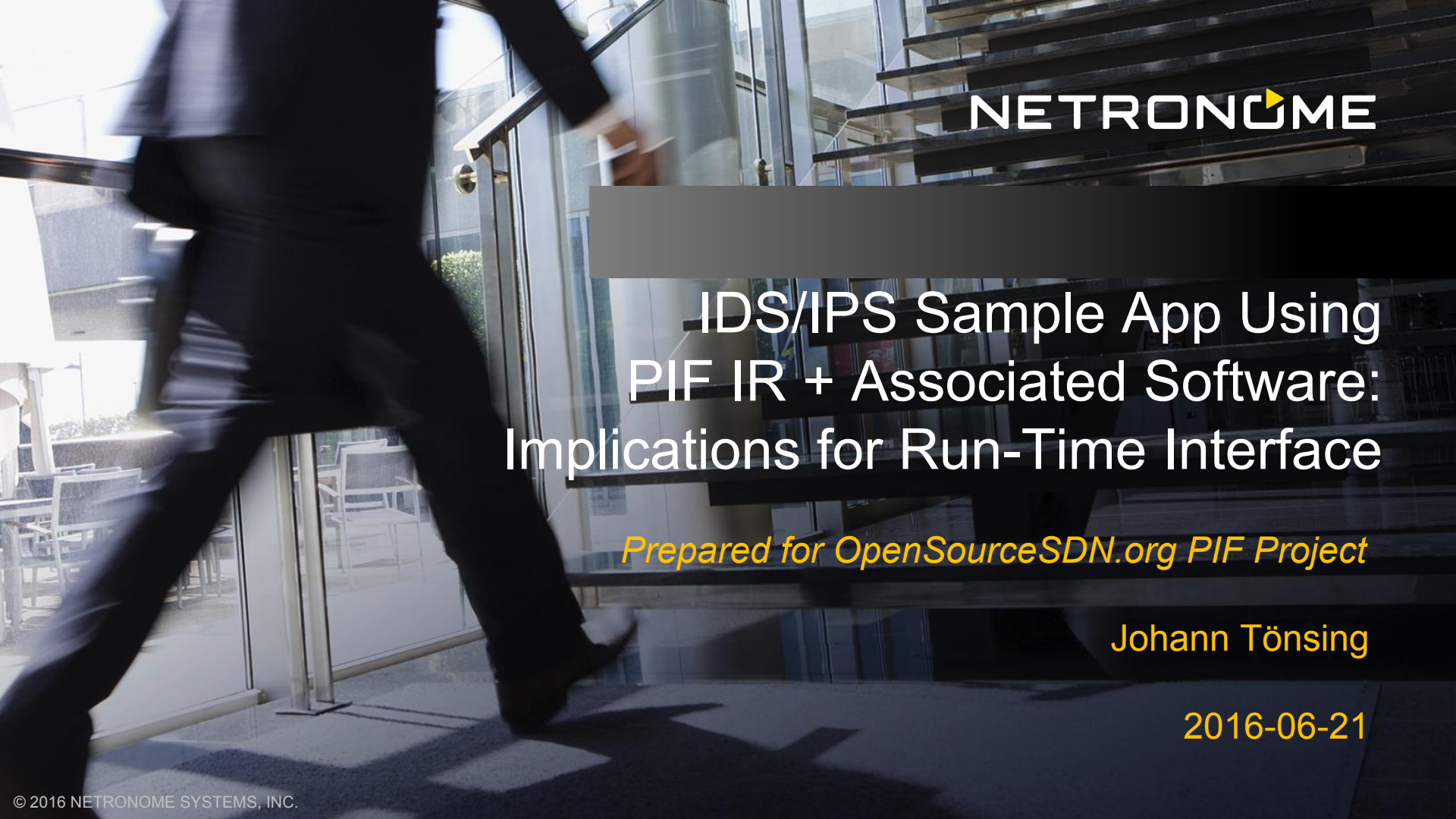


NETRONOME

# IDS/IPS Sample App Using PIF IR + Associated Software: Implications for Run-Time Interface

*Prepared for OpenSourceSDN.org PIF Project*

Johann Tönsing

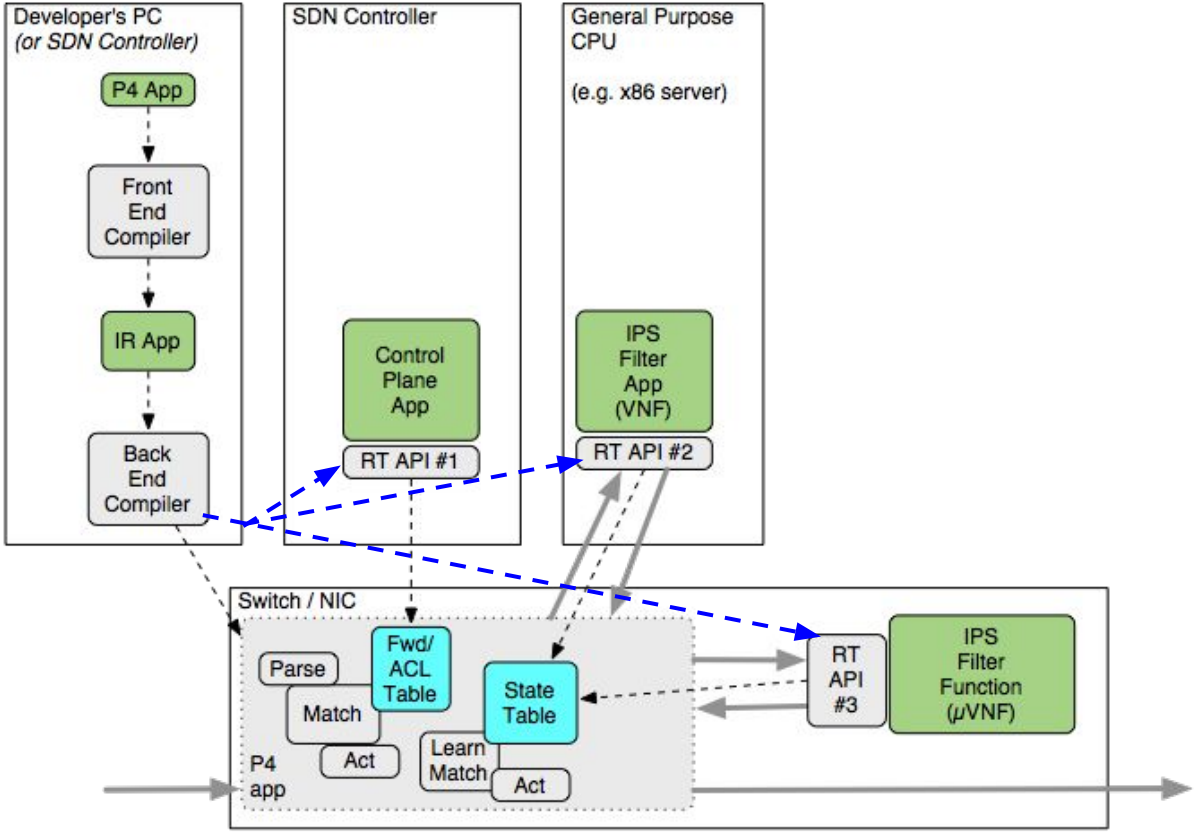
2016-06-21

# Requirements

NETRONOME

- Basic datapath
  - Forwarding (L2/L3 or SDN style)
  - ACLs
  - Steering traffic to other functions (see below)
- Intelligent filtering functionality - software / firmware / ...
  - Near fastpath (medium speed)
  - Far from fastpath (lower speed)
- Statefulness
  - Learning of microflows
    - Create/remove entries in state table
    - Timeout handling
  - Software to update policy (actions associated with state)
    - Per learnt microflow - e.g. adjust policy per TCP connection
    - Per larger “milliflow” - e.g. ban malicious node’s IP

# Design




API #1 = slow, decoupled, generic

API #2 = medium speed, closely coupled, generic/generated

API #3 = fast, directly attached, generated

All support data structure access and packet I/O aspects

# Notes of Discussion

NETRONOME

Q: Could #2 be implemented in terms of / via #3 to simplify implementations?

A: Yes - e.g. for switch platforms, the control processor would normally be involved in #2 and #3 anyway. It however depends on the implementation whether this is helpful - e.g. keeping #2 in the same path as the packets can ensure ordering of control messages relative to the packets. This is applicable to some platforms where #2 is processed close to the fastpath which also implements the P4/PIF code.

(Implementing #2 via #1 is not possible as #1 is the lower performing one.)

Q: Why does the diagram not show packets going to/from the SDN controller (packet in/out)?

A: Such an exception path could certainly exist, but this path was omitted in the interest of brevity as it does not introduce new concepts.

Note: Supporting statefulness is already on the IR to-do list (issue tracker).

# Next Steps

NETRONOME

- Make provision for stateful data structures
  - Instantiate and configure via program: key, timeout behavior
  - Action in program: learning / unlearning from P4 / PIF or associated software
  - API: query / populate from associated software
  - Events: timeout notification to associated software (TBD also P4/PIF?)
- Make provision for multiple incarnations of run-time API
  - Performance / closeness
    - Extremely generic - remote, decoupled (protocol) - slow
    - Somewhat generic - nearby, closely coupled (protocol/API) - medium performance
    - Generated - directly coupled (API) - fast
  - Operations
    - Data structure access (R/W/…)
    - Event handling - publish / subscribe
    - Packet I/O