

# Automated Switch Validation with P4 Models

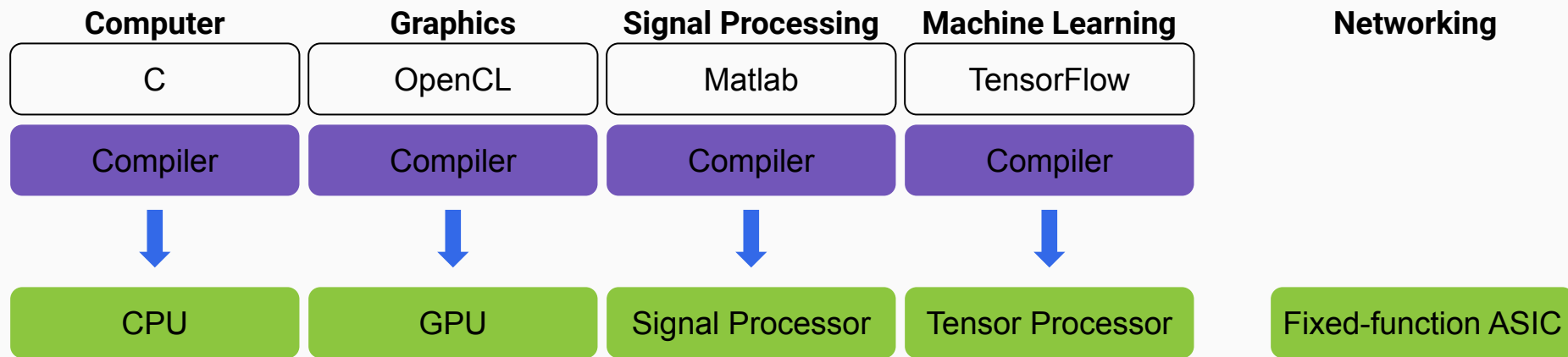
Afruz Bakhshiyeva & Luis Jira

# Agenda

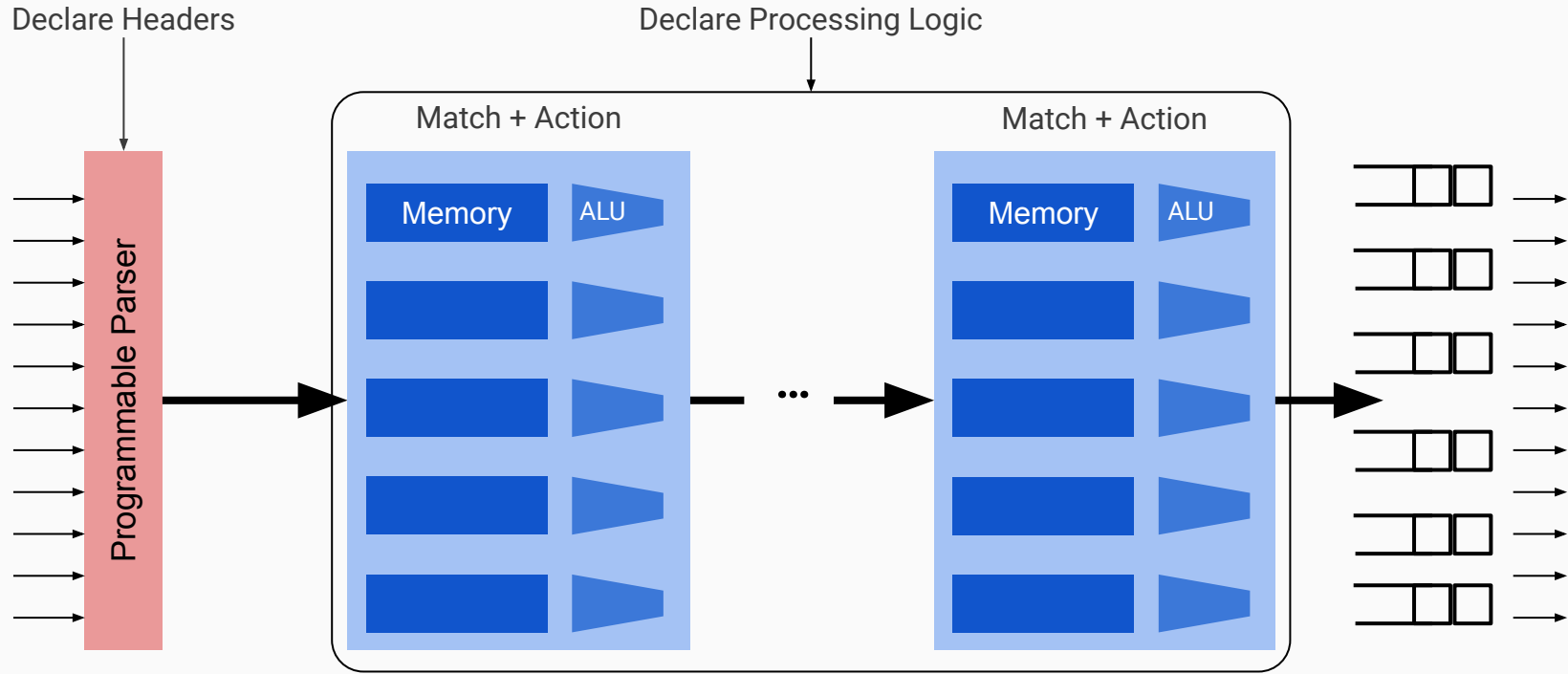
- P4 Introduction
- P4 for Formal Models
- Automated Validation based on P4 Models

# P4 Introduction

# Why does P4 exist?



# Programmable Switch Architecture



From Nick McKeown, "Programmable Forwarding Planes Are Here To Stay", IEEE ICC 2018

# P4 Parsing

Parser

Control

Deparser

```
1 header Ethernet_h{
2     bit<48> dstAddr;
3     bit<48> srcAddr;
4     bit<16> etherType;
5 }
6
7 struct headers {
8     Ethernet_h ethernet;
9 }
10
11 parser MyParser(packet_in packet, out headers hdr) {
12     state start {
13         packet.extract(hdr.ethernet);
14         transition accept;
15 }
```

# P4 Control

Parser

Control

Deparser

```
1 control MyControl(inout headers hdr,  
2                     inout standard_metadata_t standard_metadata) {  
3  
4     action drop() {  
5         mark_to_drop(standard_metadata);  
6     }  
7  
8     action forward(bit<9> egress_port) {  
9         standard_metadata.egress_spec = egress_port;  
10    }  
11  
12    table dmac {  
13        key = {hdr.ethernet.dstAddr: exact;}  
14        actions = {forward; drop;}  
15        size = 256;  
16        default_action = drop;  
17    }  
18  
19    apply {  
20        dmac.apply();  
21    }  
22 }
```

# P4 Deparser

Parser

Control

Deparser

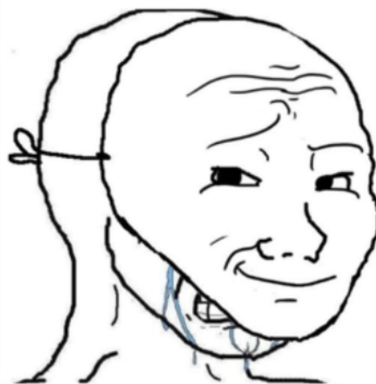
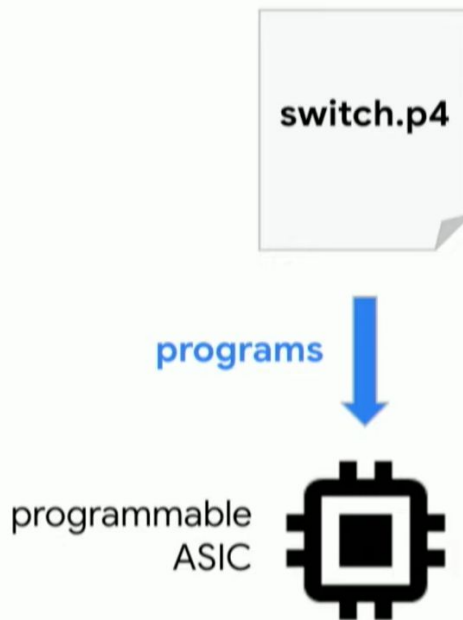
```
1 control MyDeparser(in headers hdr, packet_out packet) {  
2     apply {  
3         packet.emit(hdr.ethernet);  
4     }  
5 }
```



# P4 for Formal Models

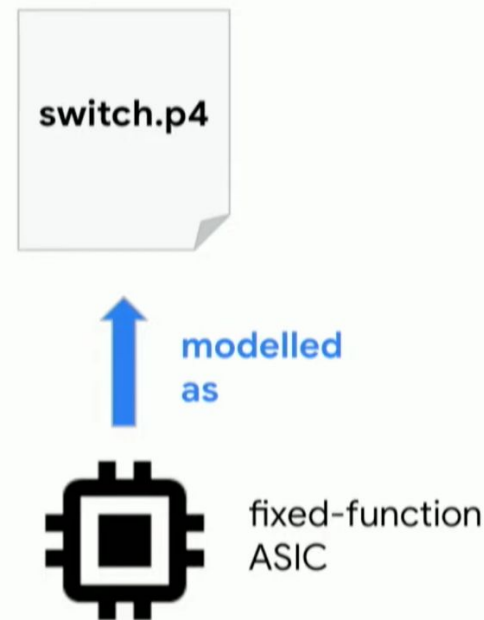
# Google's unorthodox (mis)use of P4 programming language

## P4 as intended

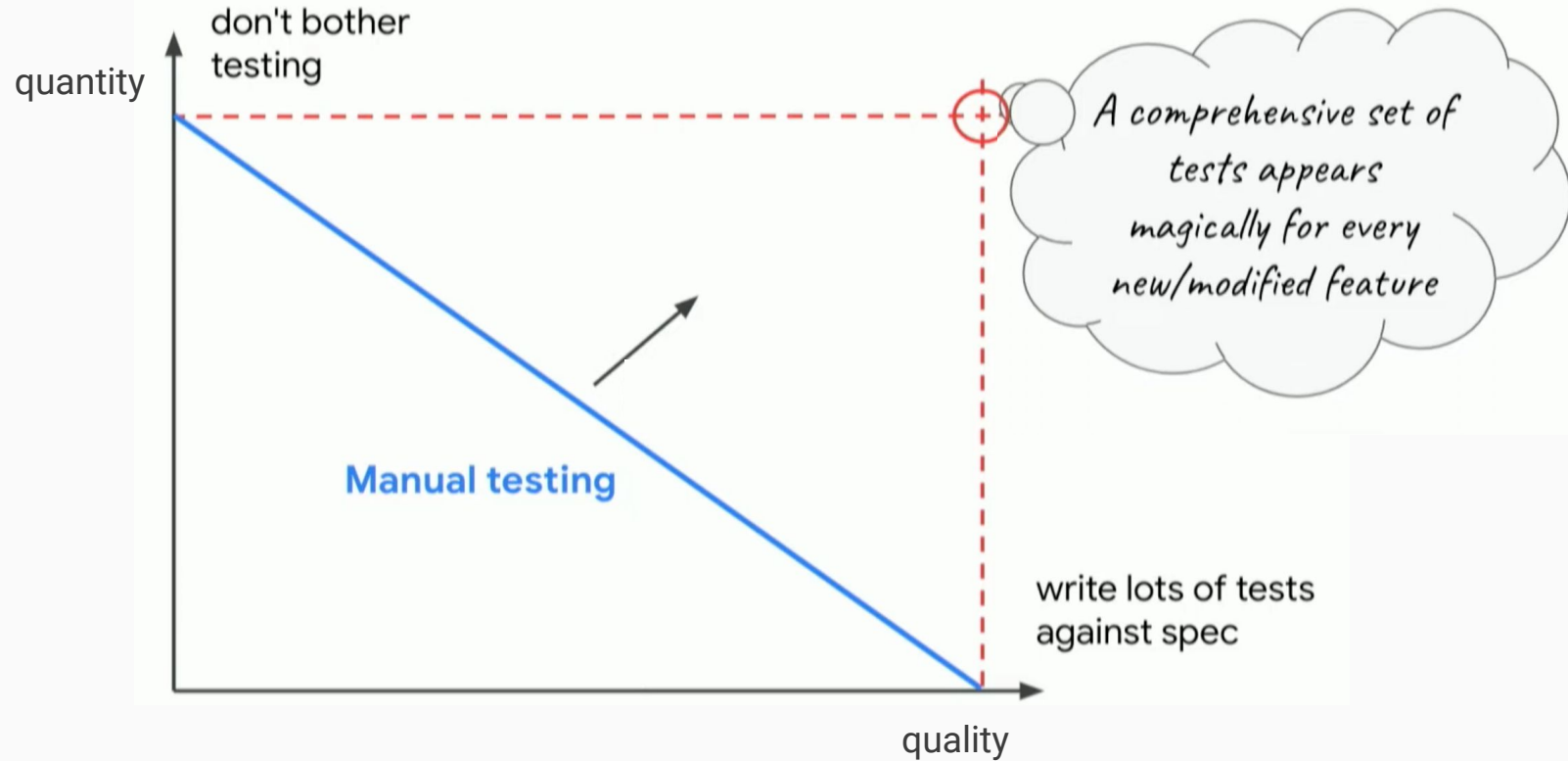


**WHY???**

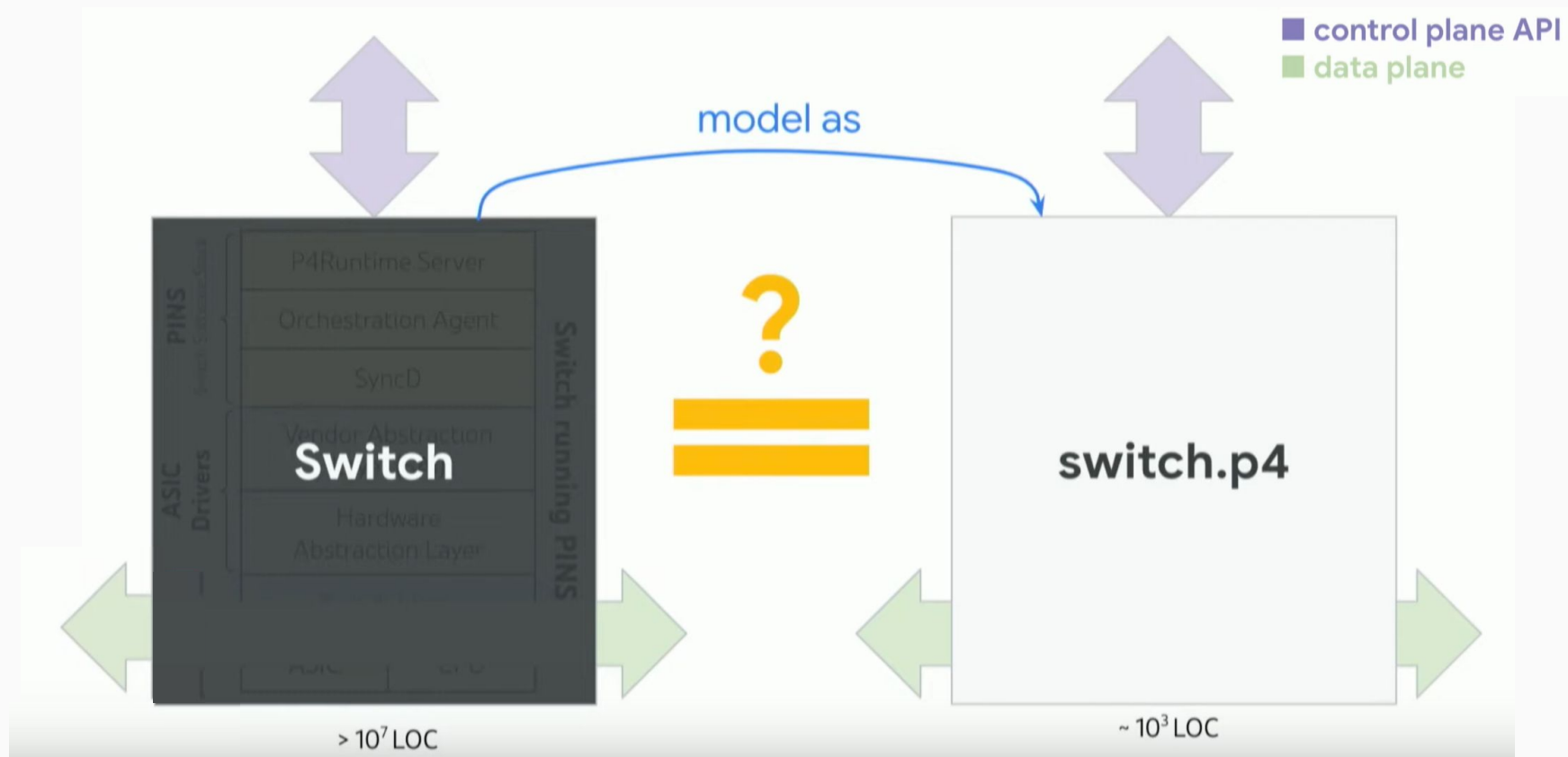
## P4 at Google



# State of the Art



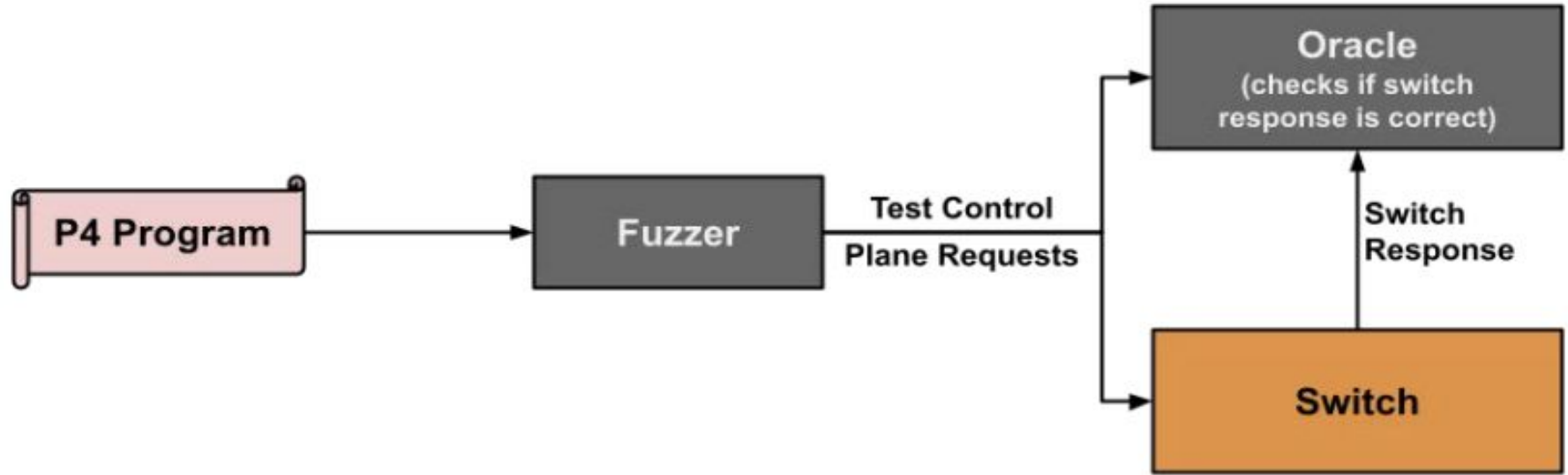
# Switch Abstraction



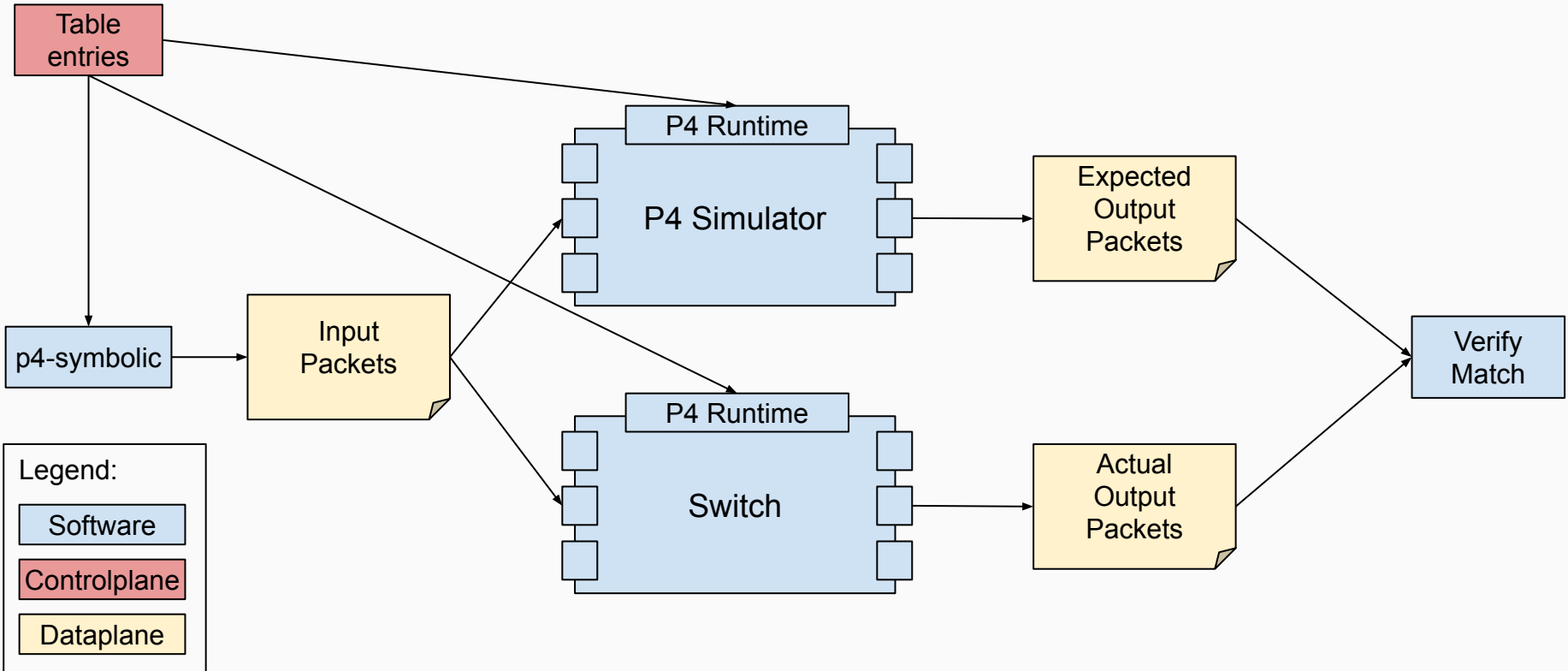
From Steffen Smolka, "SwitchV: Automated SDN Switch Validation with P4 Models", SIGCOMM'22

# Automated Validation from P4 Model

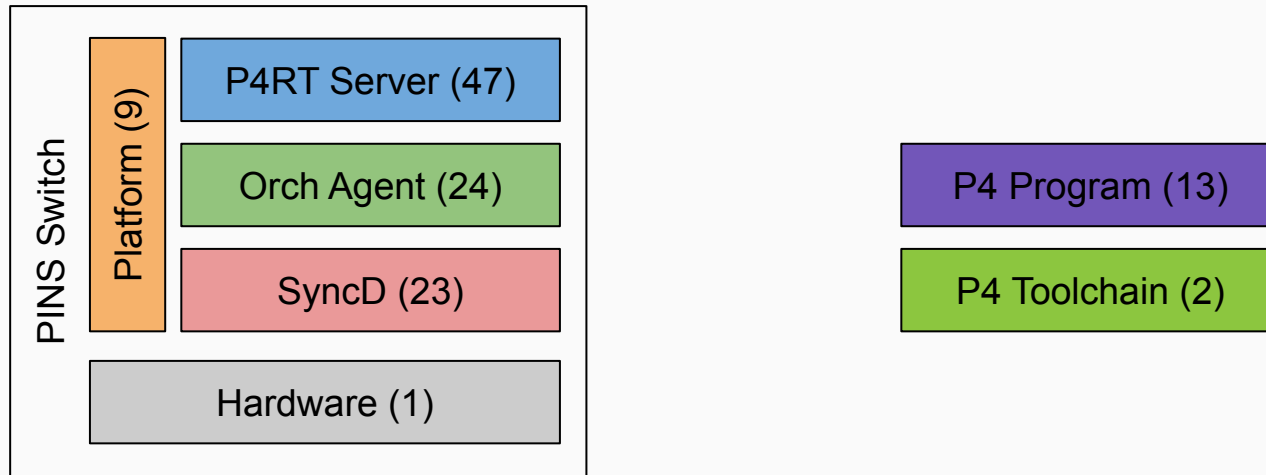
# Automated Control Plane API Validation



# Automated Data Plane Validation



# Bugs Found Across All Layers





# In a Nutshell

- P4 is a data plane programming language
- P4 can model a fixed-function switch
- Can automatically generate test inputs for:
  - Control Plane API
  - Data Plane
- Automatic validation eliminates dilemma of quantity vs quality
- Dare to think outside of the box! Tools can be used outside of their scope of use!



*Automation may be a good thing, but don't forget that it  
began with Frankenstein! ~Anonymous*

Thank You!

Q & A

# References

- [Kinan Dak Albab et al. 2022. SwitchV: automated SDN switch validation with P4 models. In Proceedings of the ACM SIGCOMM 2022 Conference \(SIGCOMM '22\). \[ACM\]](#)
- [SwitchV: Automated SDN Switch Validation with P4 Models \(TS 5, SIGCOMM'22\) \[YouTube\]](#)
- [Leveraging P4 to Automatically Validate Network Switches - Stefan Heule, Google - ONF Connect 19 \[YouTube\]](#)
- [P4<sub>16</sub> Language Specification](#)
- [<https://www.svd.se/a/EoBWK/aret-utan-sommar-fodde-varldsberomt-monster>](#)

# Grading Criteria

- timing: The presentation's length is between 6:30-7:30 minutes (hard limit)
- well-structured: Structure is announced and graphically visible
- motivation: The presentation contains a good, motivating introduction telling why this presentation is important
- technical: The presentation contains one part that is deeply technical
- code: The presentation contains valuable and readable code snippets
- originality: The presentation contains one part that is original (eg less than 100 results on Google on this topic)
- reflection: The presentation contains a reflective part
- sota: There is one good slide positioning the presentation in the state of the art
- take-home: The last slide contains a good and concise take-home message
- loudly: The speakers talk loudly and clearly
- engagement: The speakers engage with the audience
- humour: The speakers are fun, have humour
- readable-slides: The slides don't have too much text
- illustration: The slides contain nice illustrations