Hybrid Packet and Flow Processing with Flowlets

Yogesh Mundada, Rob Sherwood, Nick Feamster
yhm@cc.gatech.edu, rob.sherwood@stanford.edu, feamster@cc.gatech.edu

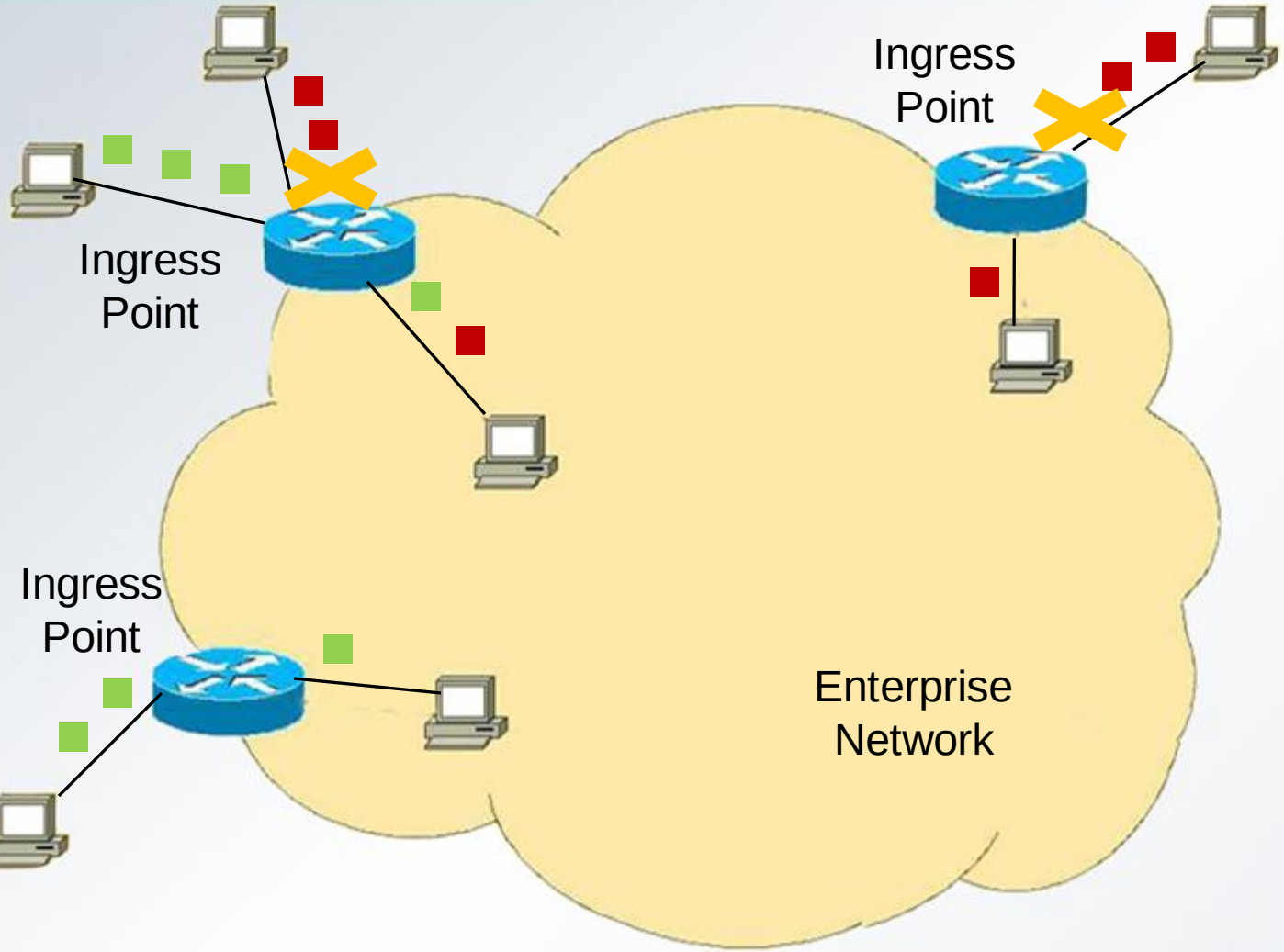
Motivation: Two Ways of Processing Traffic

- Process as an **individual packet**
  - Advantages: Flexible, More control, Micro-decisions
  - Disadvantages: Slow
- Process as an **aggregate flow**
  - Advantages: Faster, Macro-decisions
  - Disadvantages: Not enough control and flexibility

- **Flowlets:** Process packets on per-packet *and* per-flow granularity
  - Best of both worlds!

2

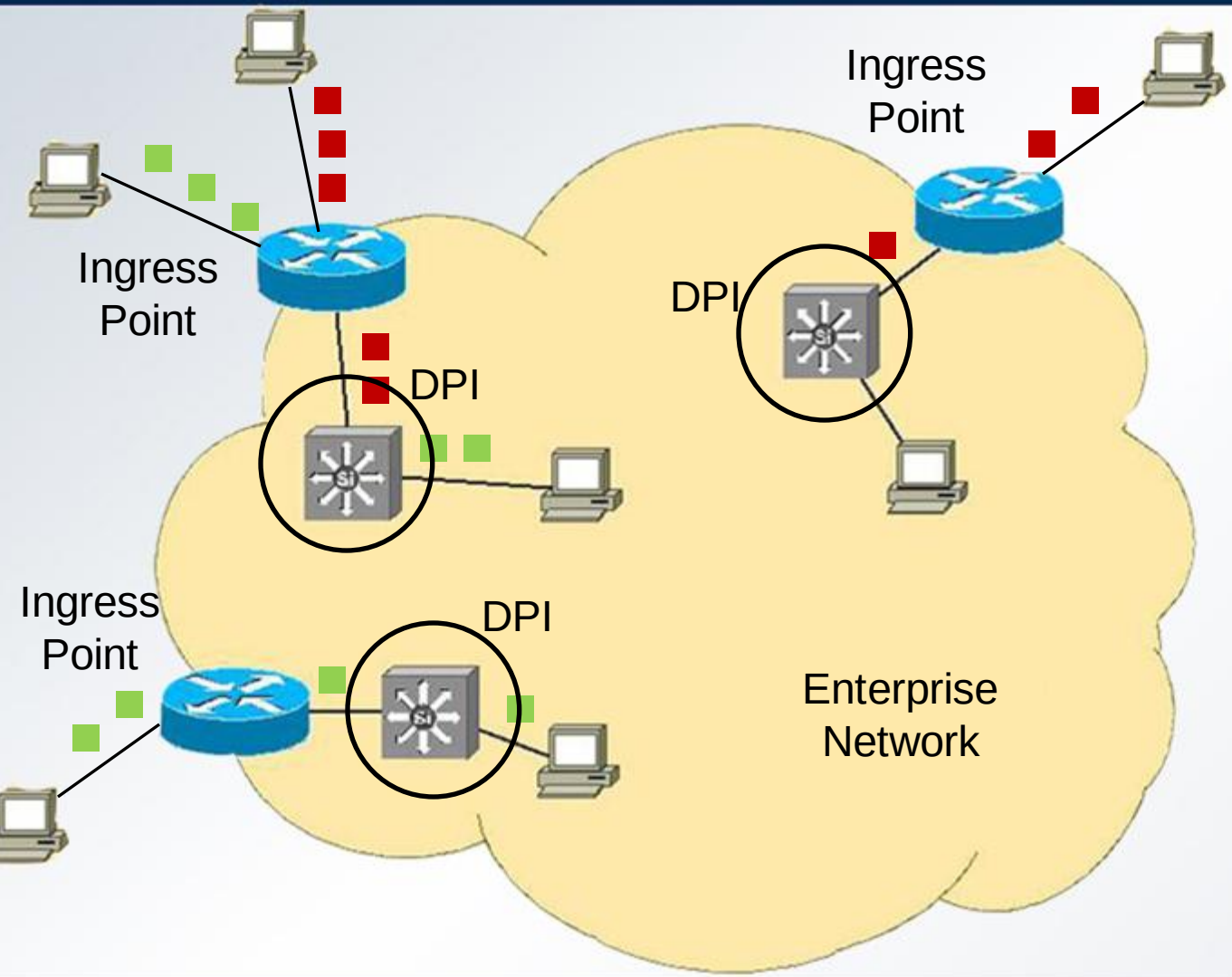
Example: Network Security (Flow-Based Only)



1. **Routers forward traffic using only packet header fields**
2. **Identify malicious traffic by examining packet data at line speed**

- **Normal traffic**
- **Malicious traffic**
- **Suspicious traffic**

Example: Network Security (Packet-Based Only, Distributed)



1. Deploy DPI boxes at each Ingress point

Disadvantages

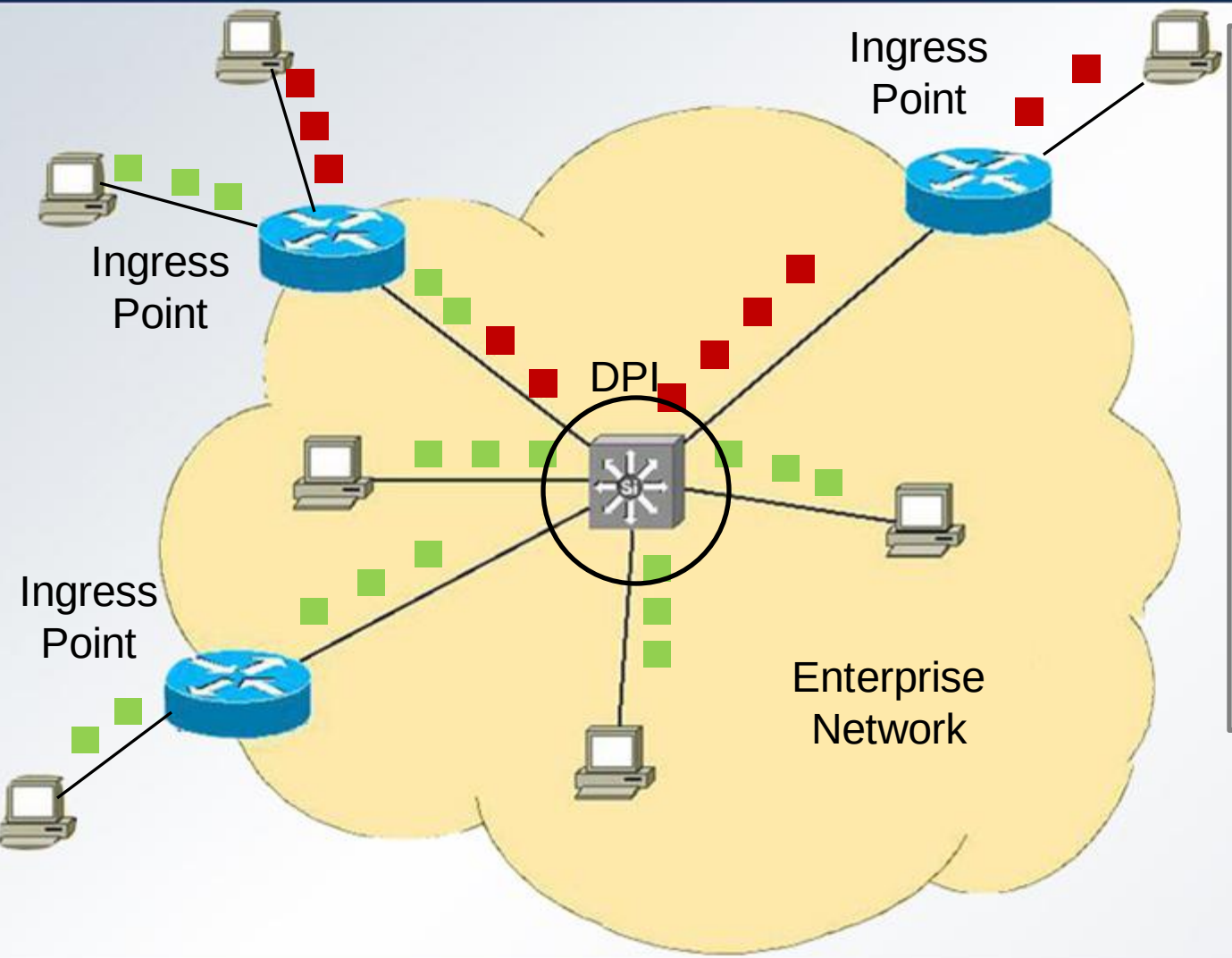
1. Distributed coordinated attacks
2. Space, Maintenance, Cost, Power

Normal traffic

Malicious traffic

Suspicious traffic

Example: Network Security (Packet-Based Only, Centralized)



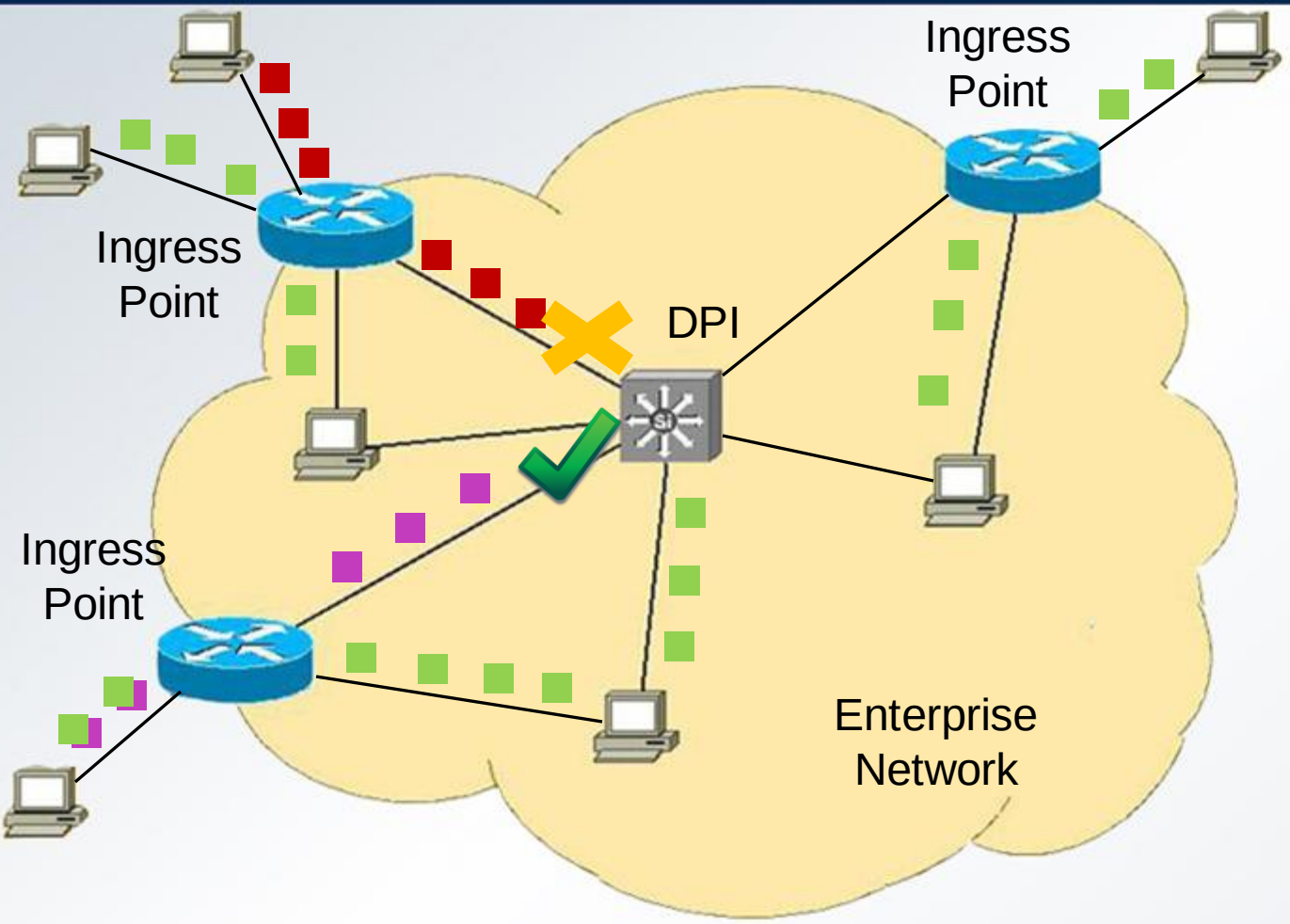
1. Redirect traffic through a centralized DPI

Advantages
1. Distributed attacks are detected
2. Less cost, power, maintenance

Disadvantages
1. Not scalable
2. Single point of failure

Normal traffic
Malicious traffic
Suspicious traffic

Better Solution: Both Packets and Flows



1. Redirect only suspicious traffic to the centralized DPI box.

Desired Properties

- Distributed monitoring
- Centralized decision making
- Packet by packet and Flow based processing

Normal traffic

Malicious traffic

Suspicious traffic

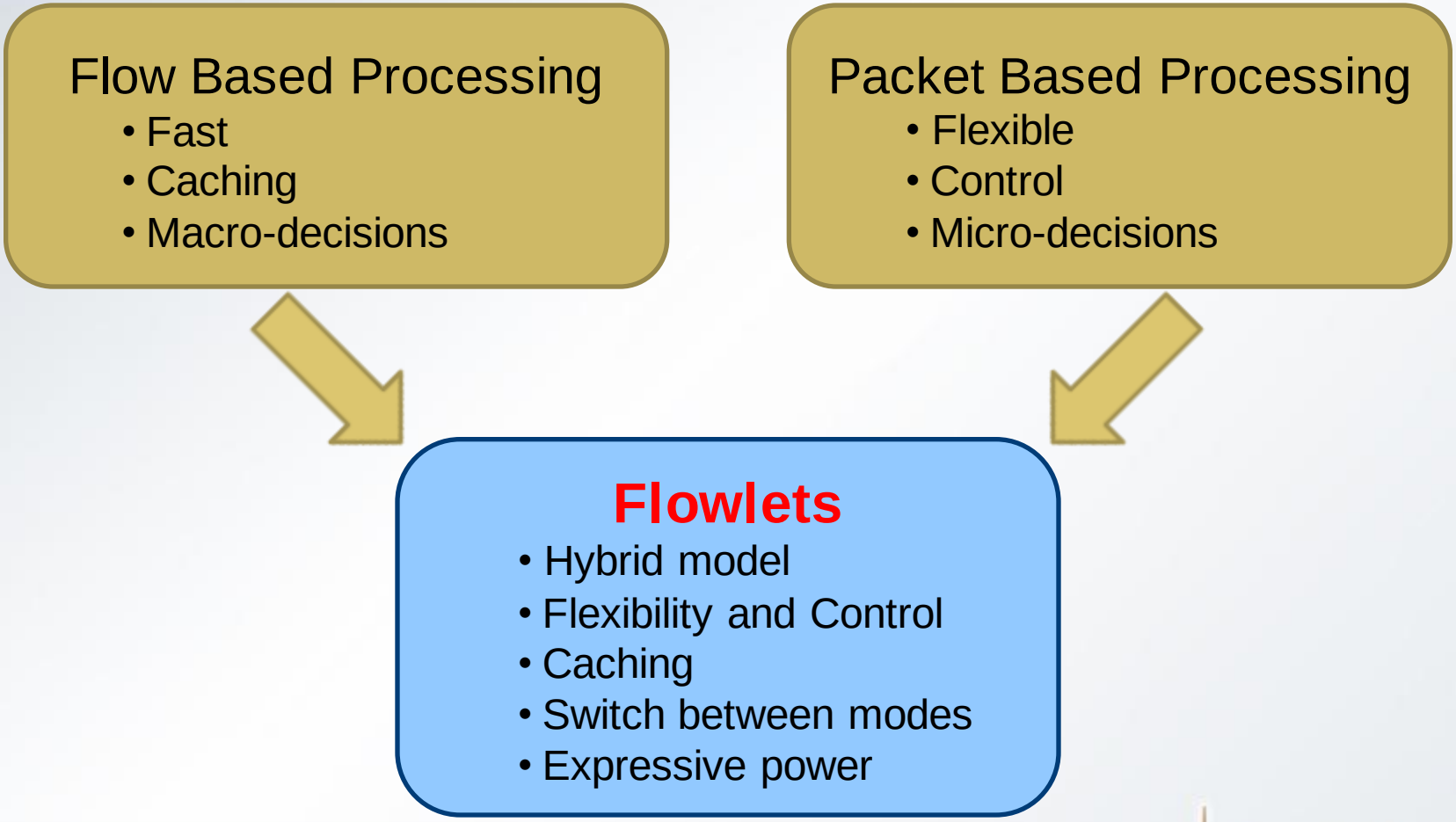
Georgia Institute of Technology

Neither Flow nor Packet-Based Processing is Sufficient

	Examples	Speed	Caching	"Macro" Decisions	"Micro" Decisions	Control/ Flexibility
Flow-based processing	MPLS, Forwarding tables, Openflow					
Packet-based processing	Active Networks, DPI, Click					

We need both !!

Main Idea: Combine Packet and Flow Processing

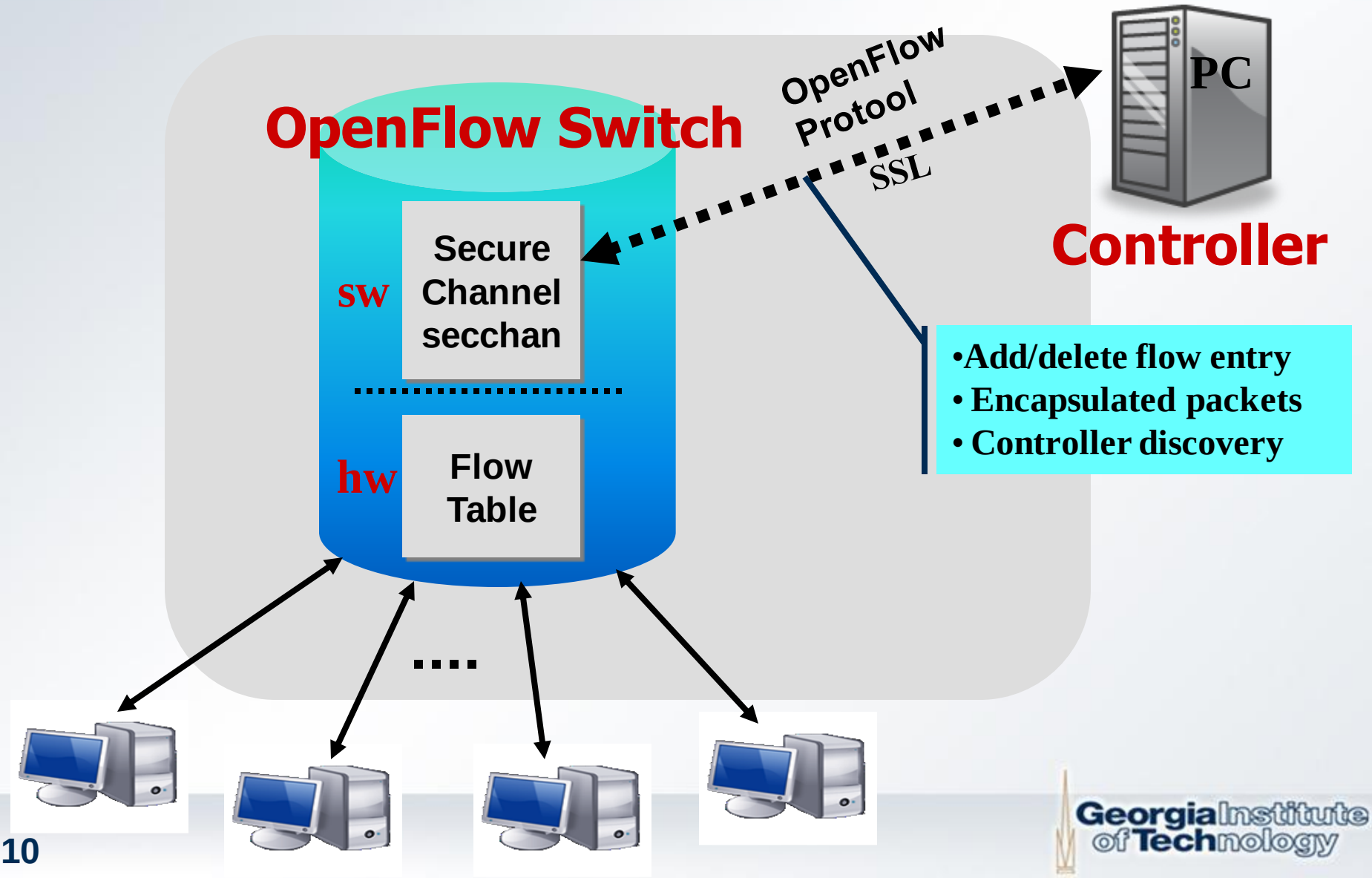


Talk Outline

- Motivation
- Example: Enterprise Security
- Flowlets: Hybrid Processing
- Background
  - OpenFlow
  - Click
- OpenFlow Click Element
- Demonstration
- Implementation Details and Decisions
- Future work and Summary

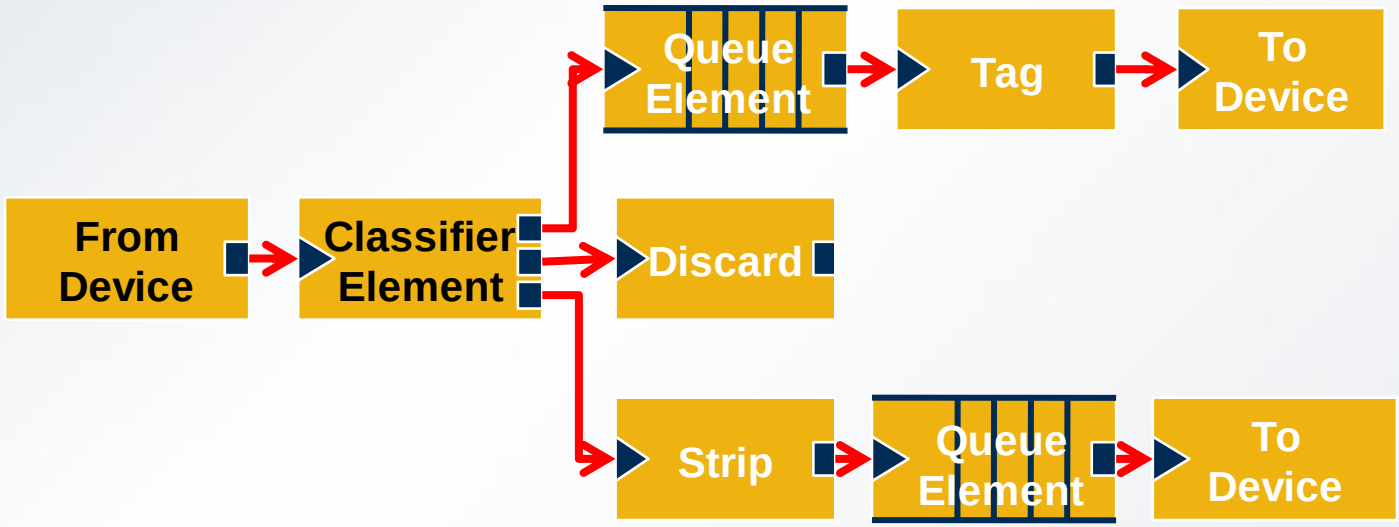
9

Flow-Based Processing: OpenFlow



Packet-Based Processing: Click

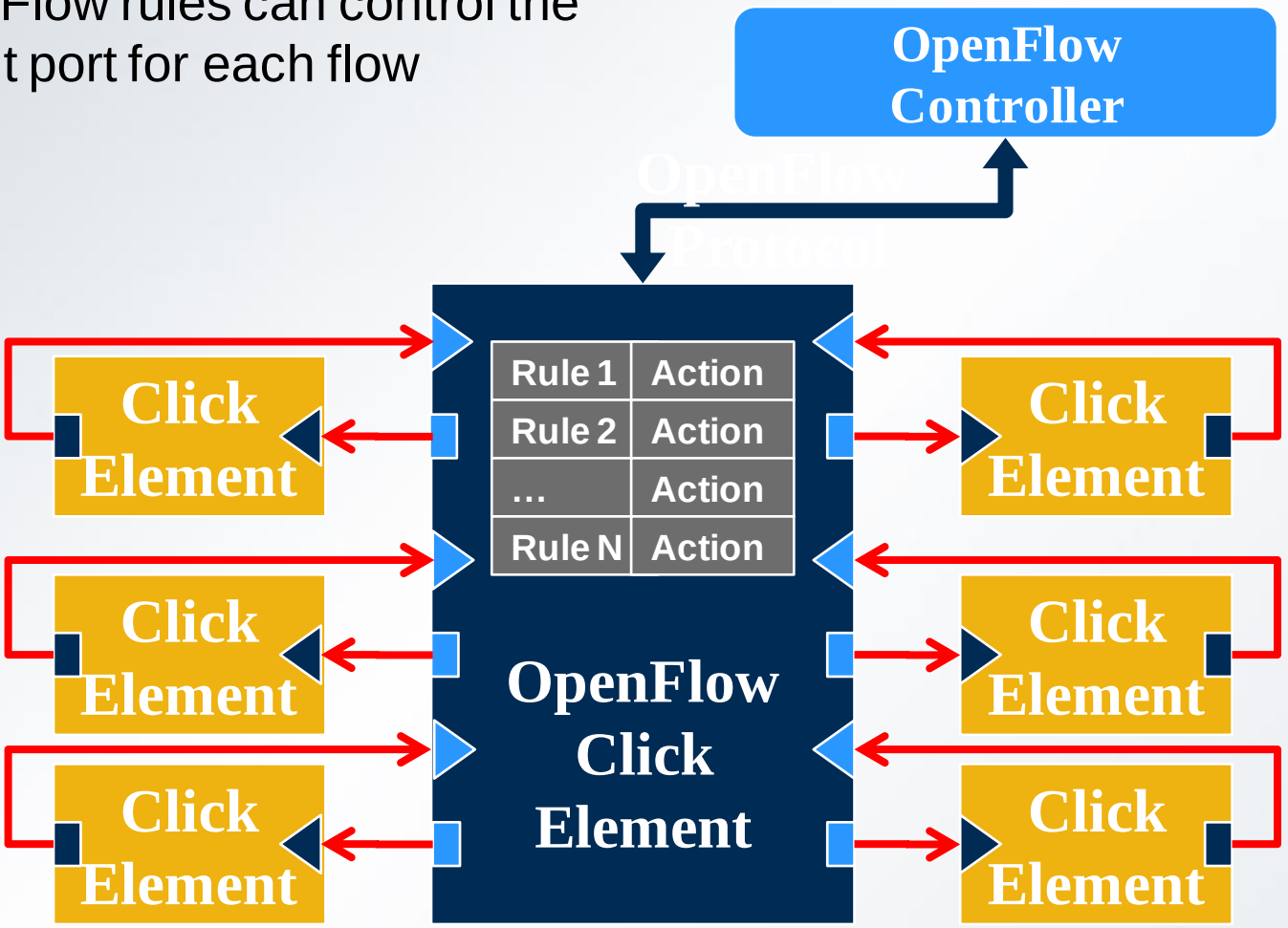
- Easy to program
- Intuitive configuration language
- Provides more control
- Large elements library



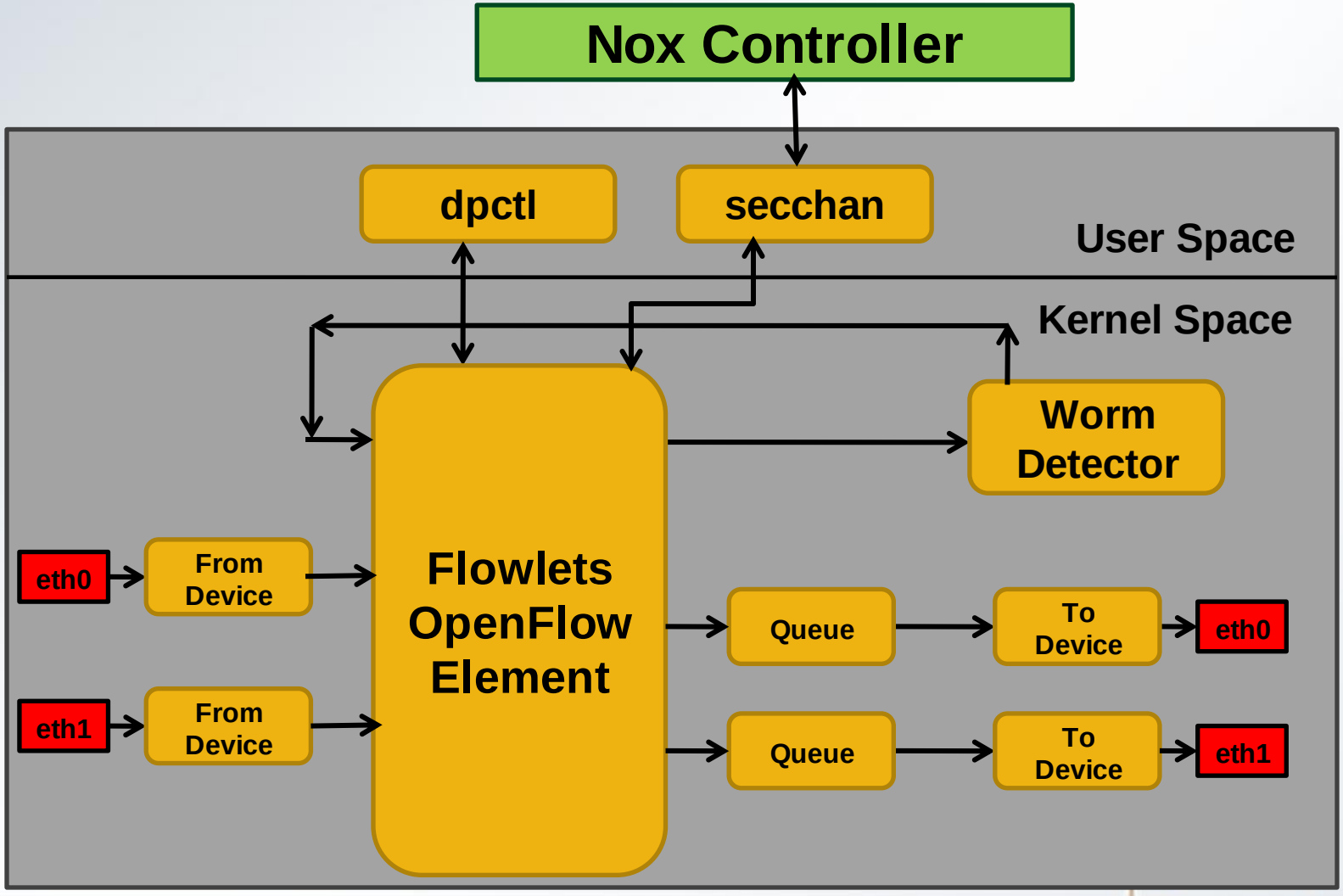
11

Hybrid: OpenFlow Click Element

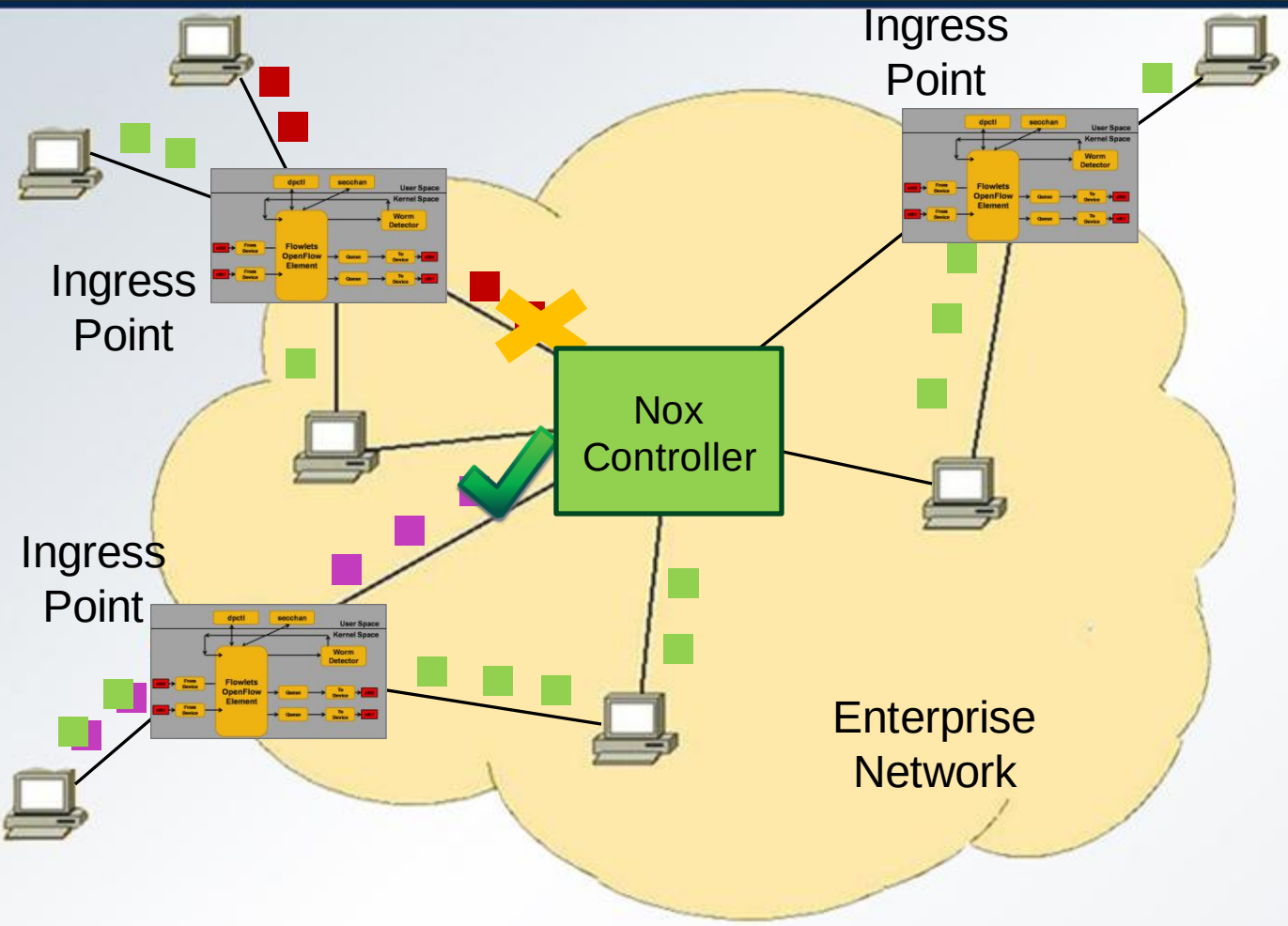
OpenFlow rules can control the output port for each flow



Example: Enterprise Security



Enterprise Security with Flowlet Processing



1. Redirect only suspicious traffic to the centralized DPI box.

Desired Properties

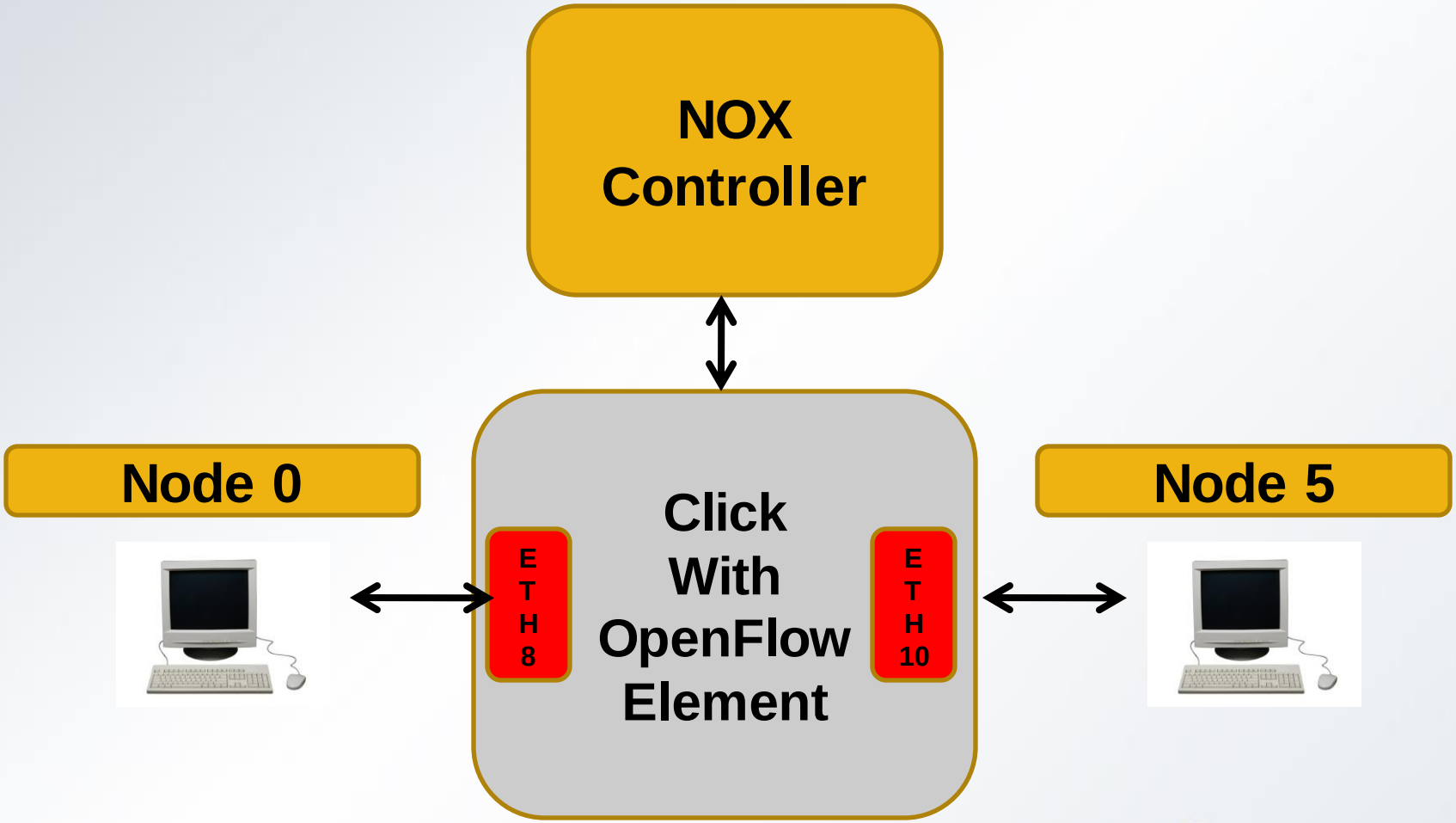
- Distributed monitoring
- Centralized decision making
- Packet by packet processing
- Flow based processing

Normal traffic

Malicious traffic

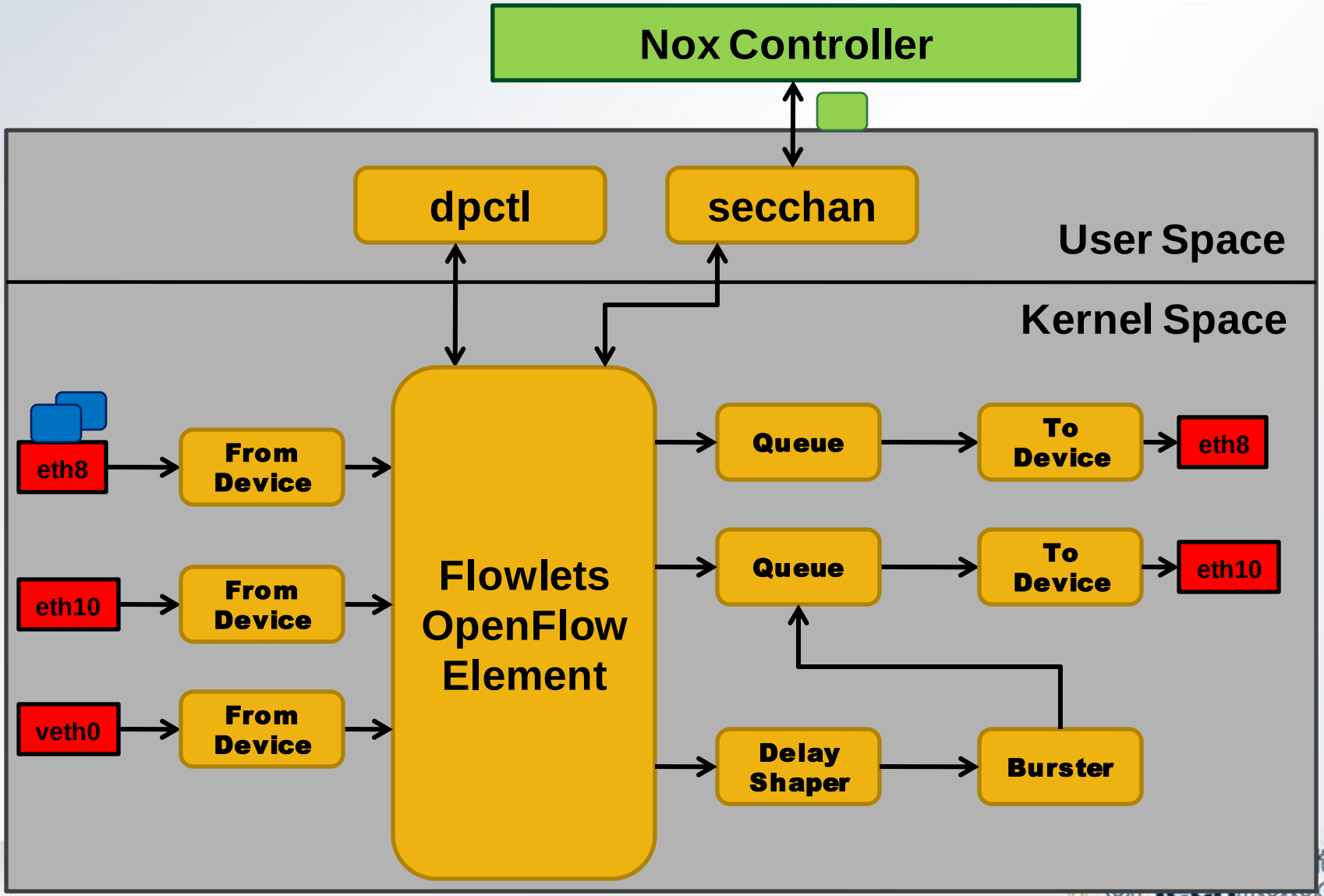
Suspicious traffic

Demonstration



15

Demonstration



Demo Click Configuration

```
q1 :: Queue;
q2 :: Queue;
q3 :: Queue;
s :: Ofswitch;

FromDevice(eth8, PROMISC true) -> [0]s;
FromDevice(eth10, PROMISC true) -> [1]s;
FromDevice(veth0, PROMISC true) -> [2]s;

s[0] -> Print("Received from eth8", MAXLENGTH 100) -> q1;
s[1] -> Print("Received from eth10", MAXLENGTH 100) -> q2;
s[2] -> Print("Received from veth0", MAXLENGTH 100) -> q3 ->

DelayShaper(2) -> b::Burster(0.1) -> q2;

q1 -> ToDevice(eth8);
q2 -> ToDevice(eth10);
```

Implementation Decisions

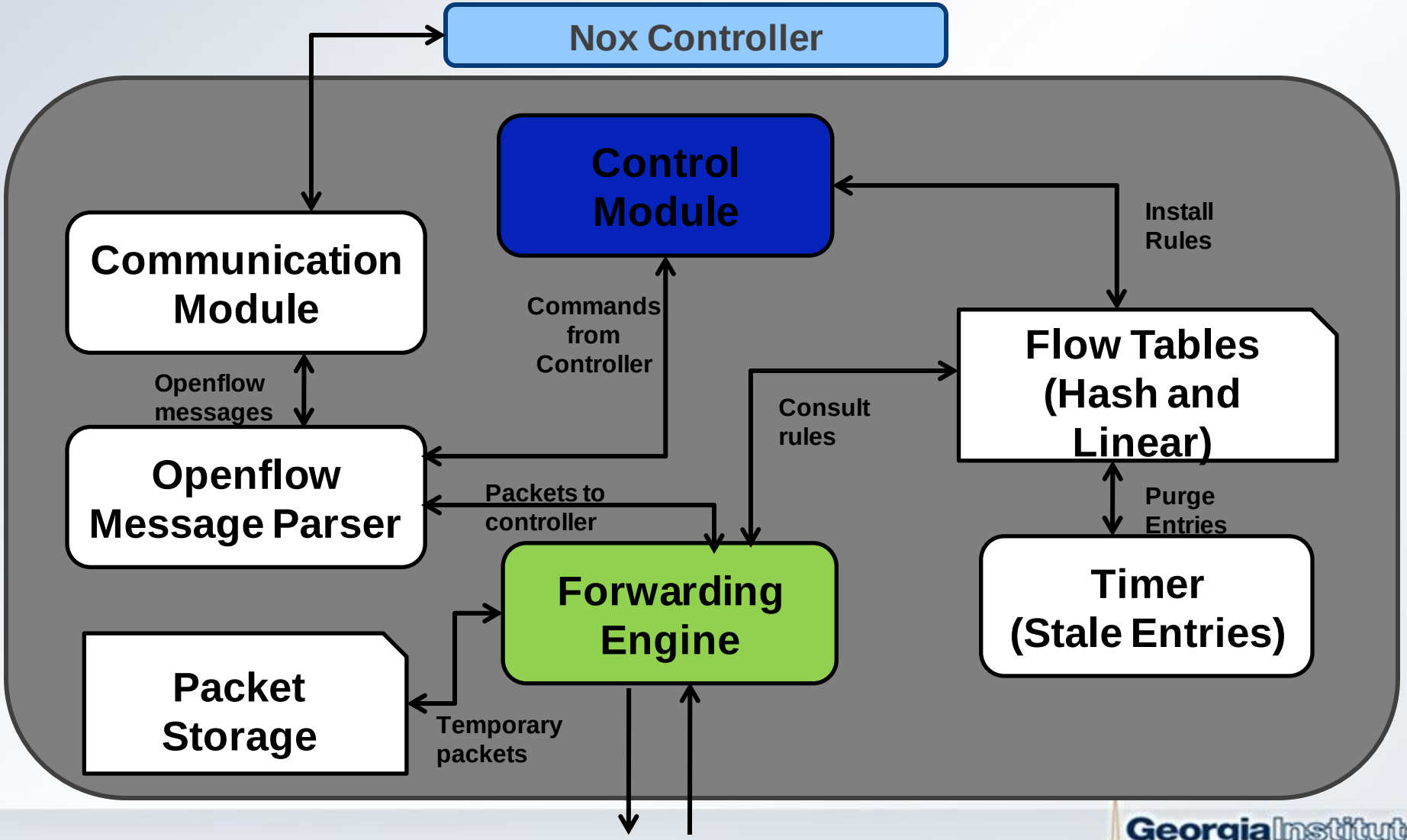
1. Implement new Click element using existing OpenFlow switch source code.

2. Implement element as kernel module.

3. Minimize changes to existing OpenFlow code base.

18

Element Architecture



19

Other Applications of Flowlets

- Loop detection: TTL zero
- Inserting/Deleting extra bits from packet headers (Splicing)
- Packet sampling
- Duplicate packet detection

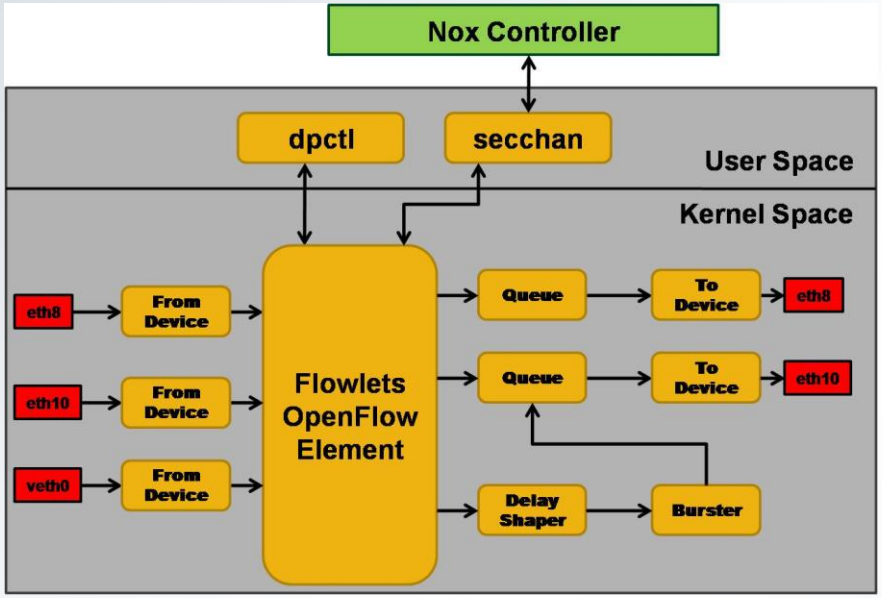
Future work

- User-space element
- Click vendor-specific action
- Dynamic port addition/deletion
- Dynamic element load/unload
- Dynamic sub-graph load/unload

21

Summary

http://www.openflowswitch.org/wk/index.php/OpenFlowClick



Flowlets

- Hybrid model
- Flexibility with control
- Caching
- Switch between modes
- Expressive power

Georgia Institute of Technology

23

Challenges

- Locking
- Refactoring code in correct modules
- Memory allocation
- Multi-threaded code
- Debugging
- Mixing C/C++ code

24