

# Multi-Tenant Programmable Data Planes

P4.org Open Source Developer Days - January 4, 2022

Radostin Stoyanov

PhD Candidate - Computing Infrastructure Group

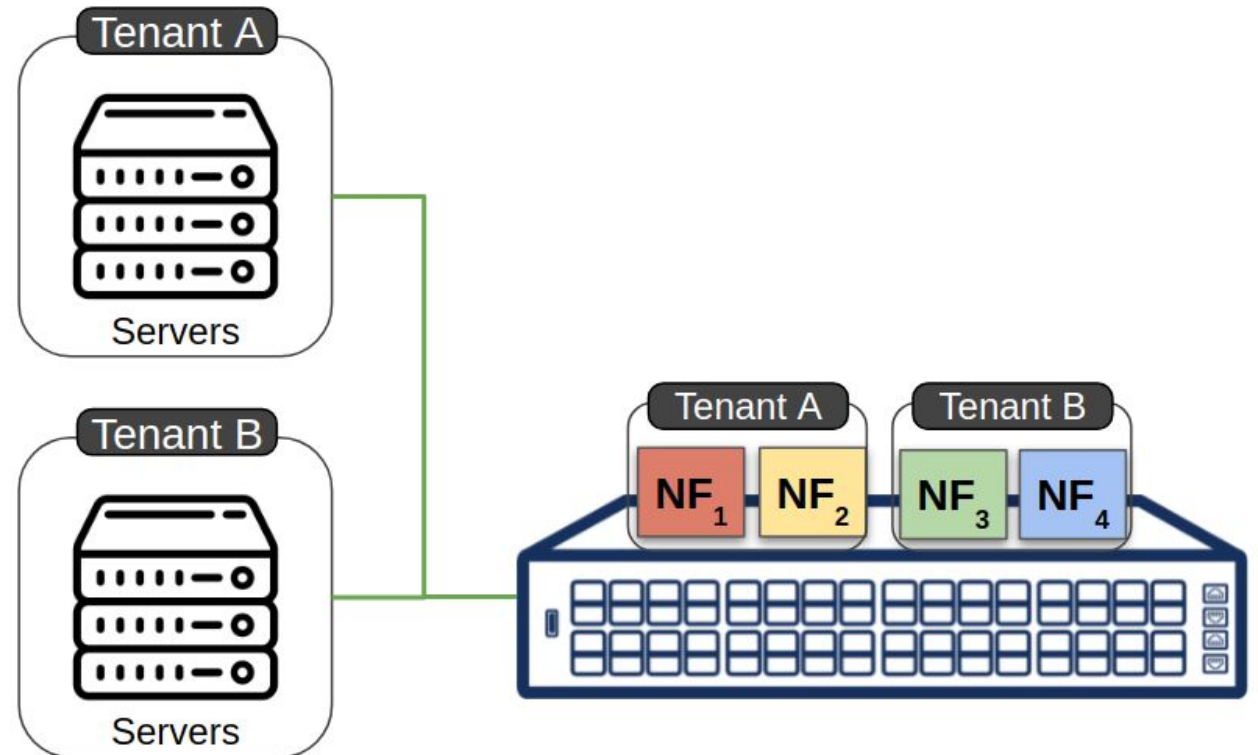
Advisor: Prof. Noa Zilberman

# Context and Motivation

Can we support multi-tenancy?

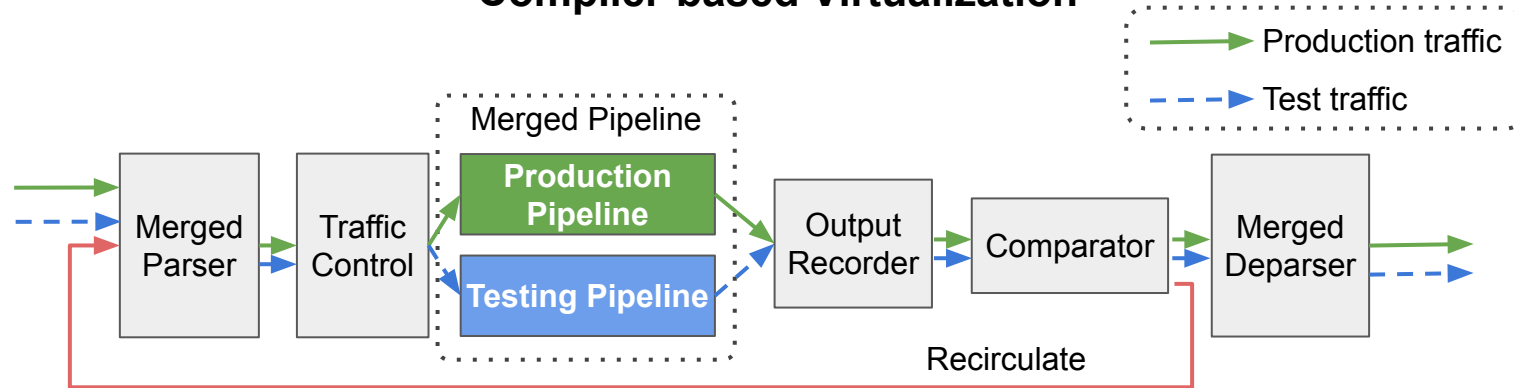
Requirements:

- Security isolation
- Performance isolation
- Resource isolation
- Runtime reconfigurability

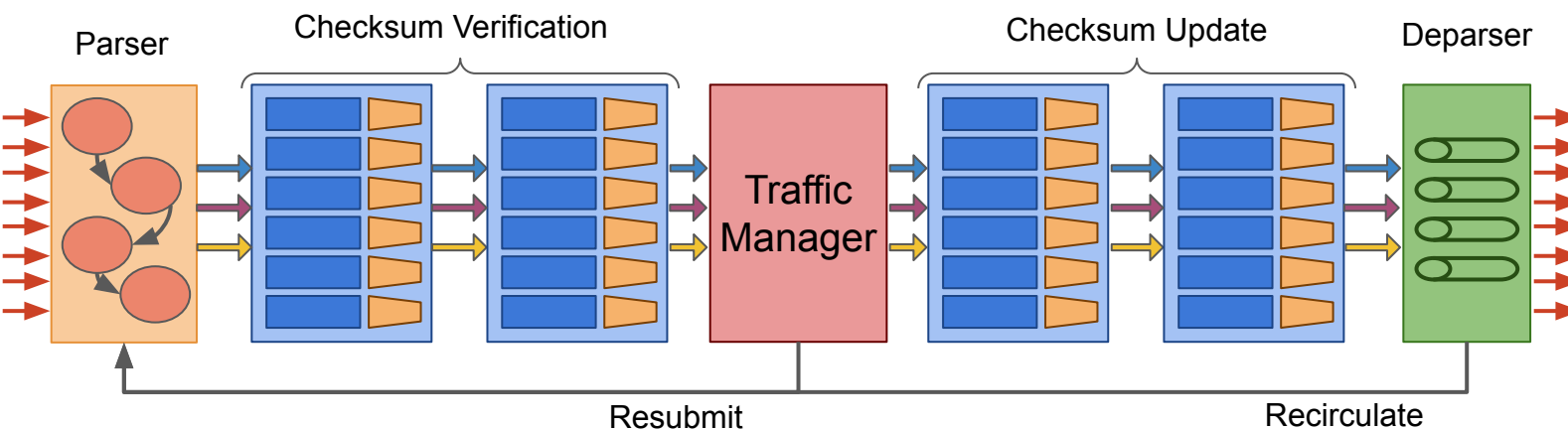


# Data Plane Virtualization

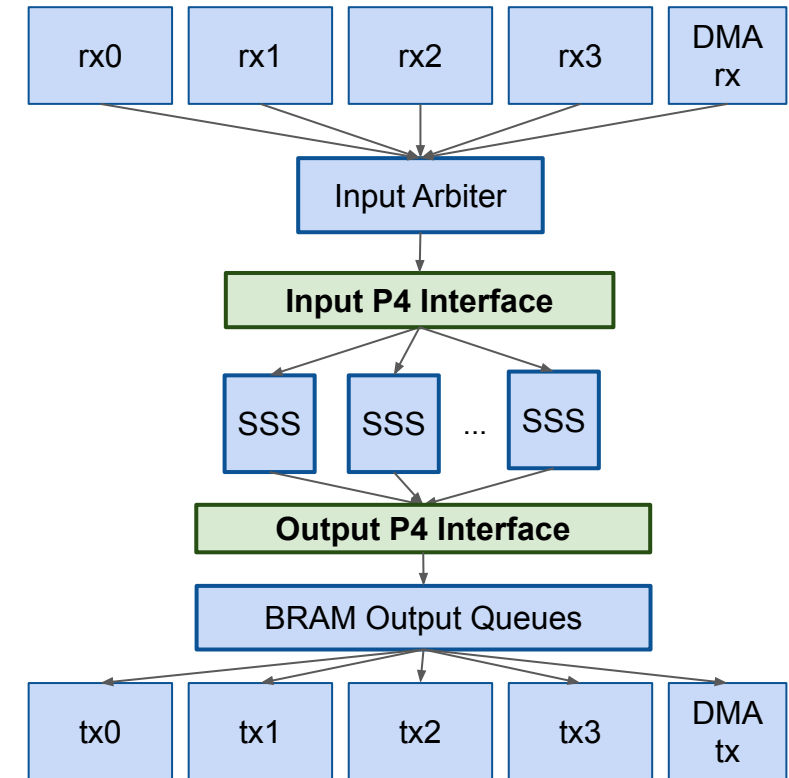
## Compiler-based Virtualization



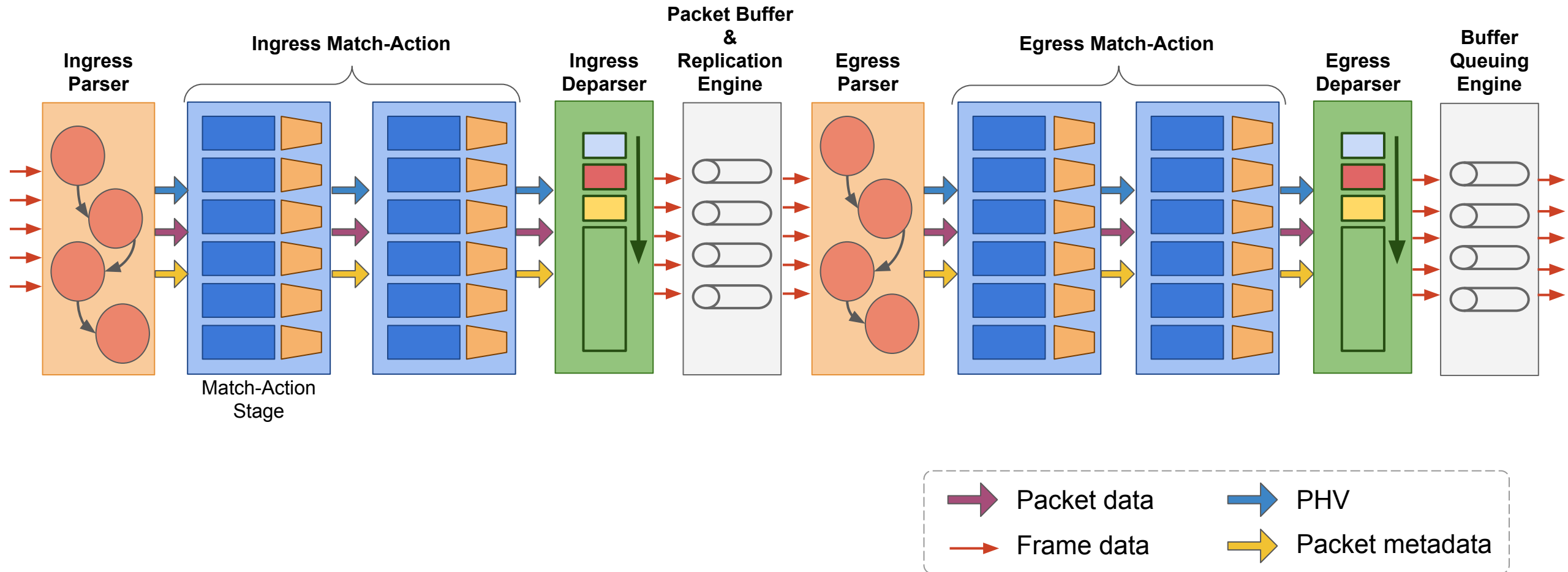
## Recirculation-based Virtualization



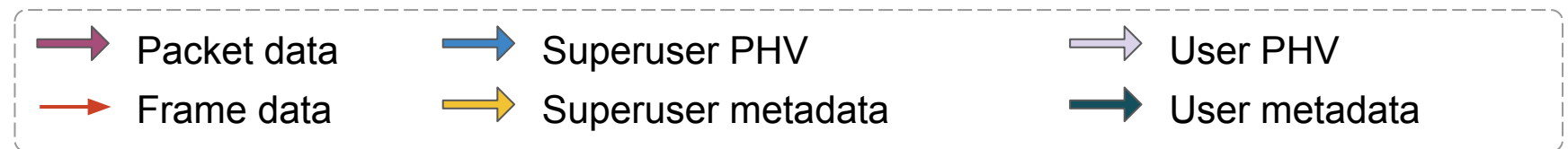
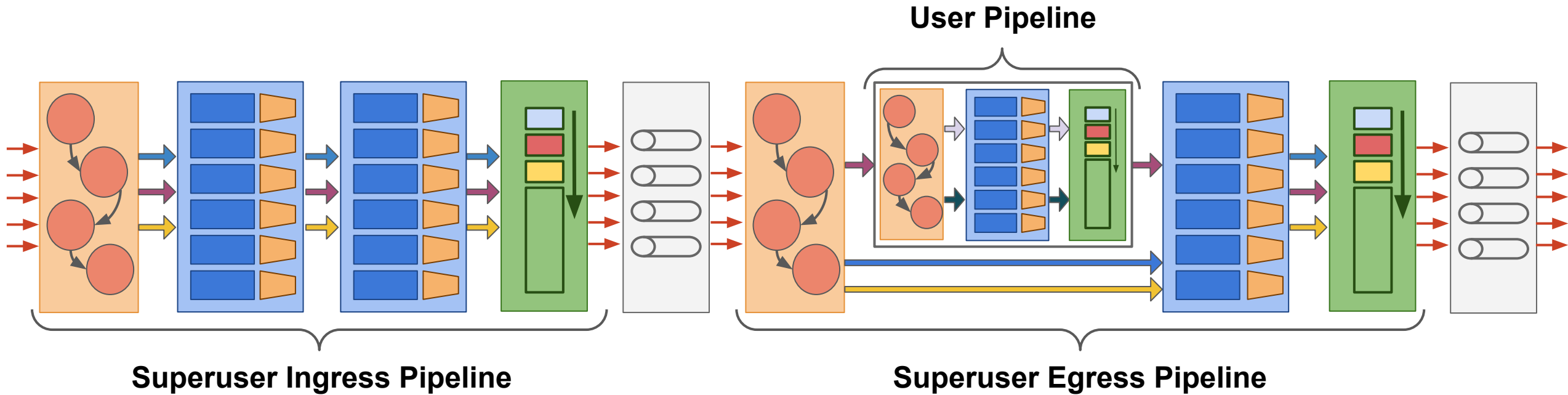
## Architecture-driven Virtualization



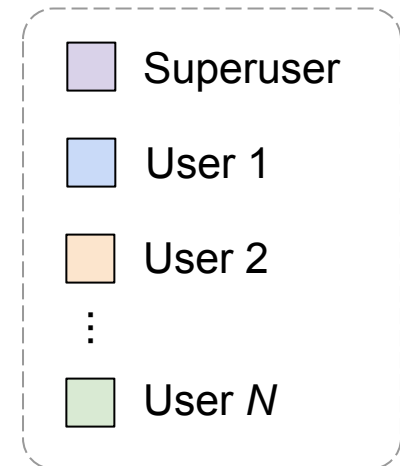
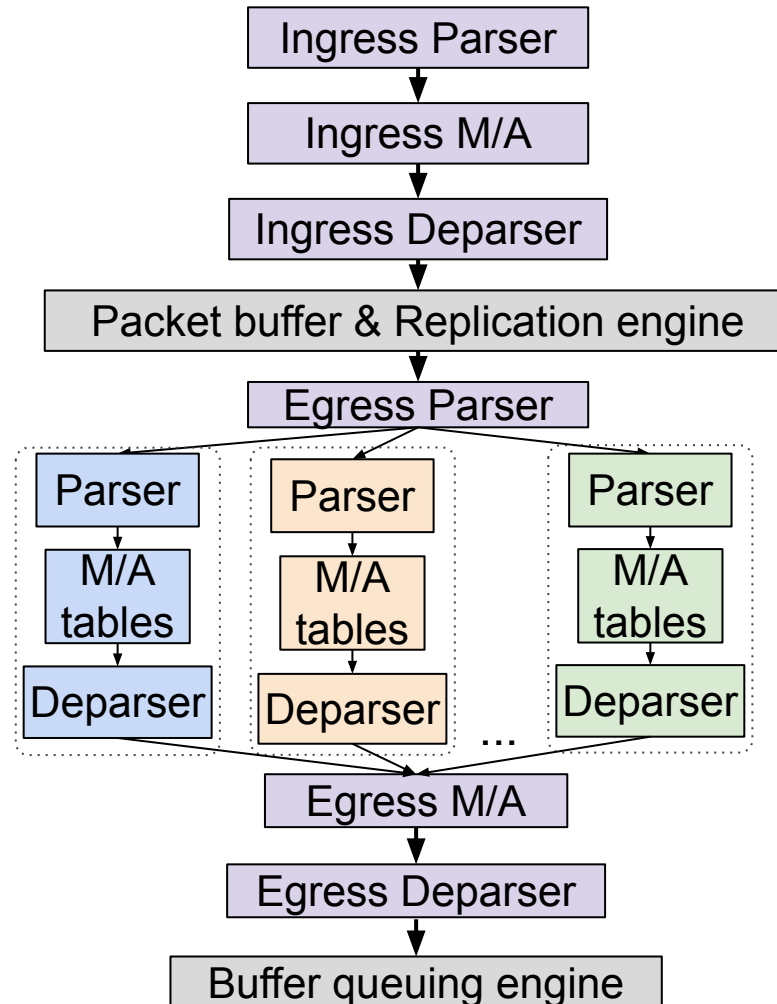
# Portable Switch Architecture (PSA)



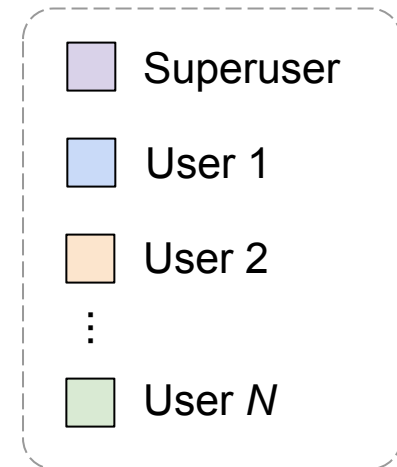
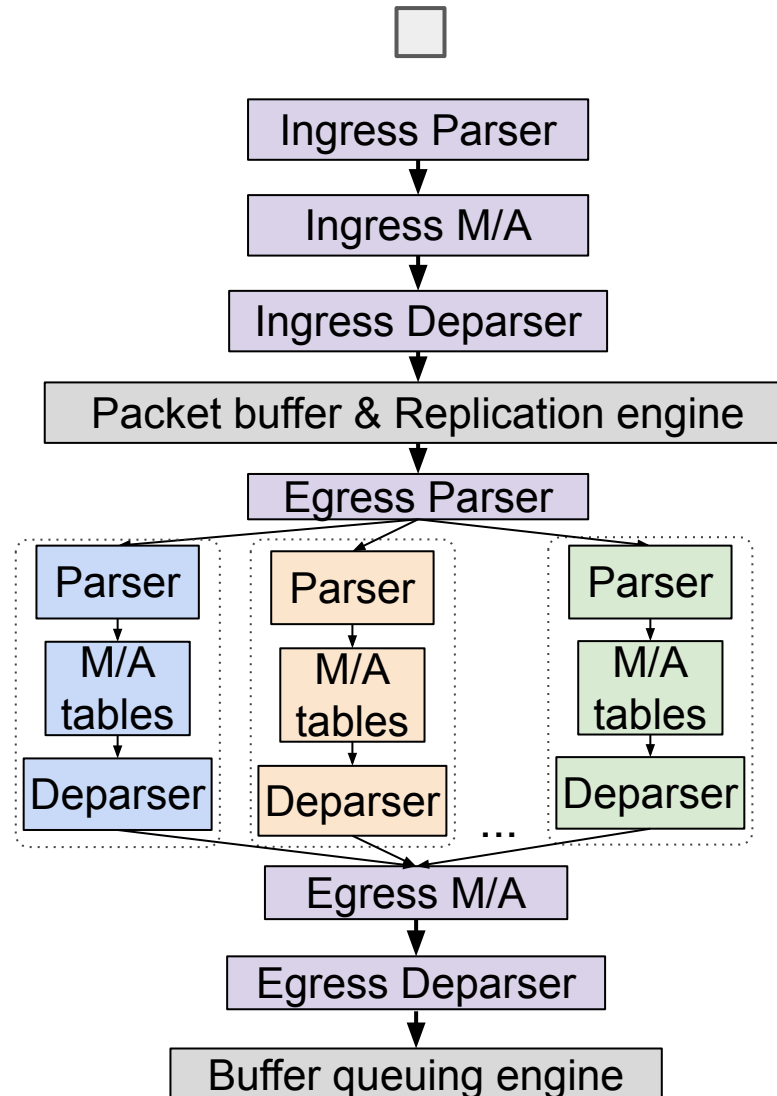
# Multi-Tenant Programmable Switch



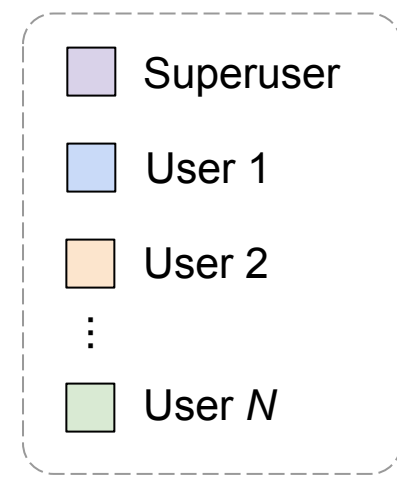
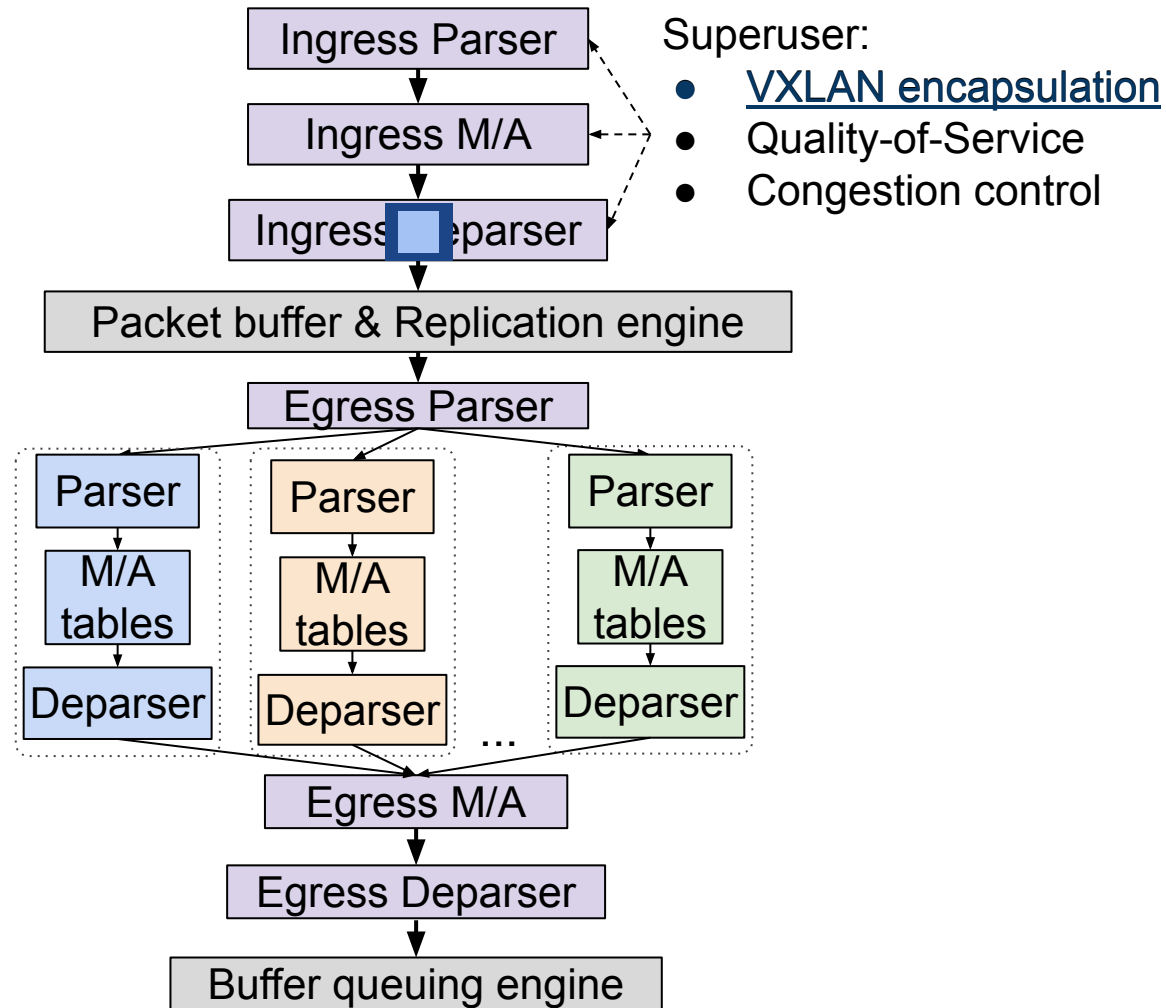
# MTPSA Packet Flow



# MTPSA Packet Flow

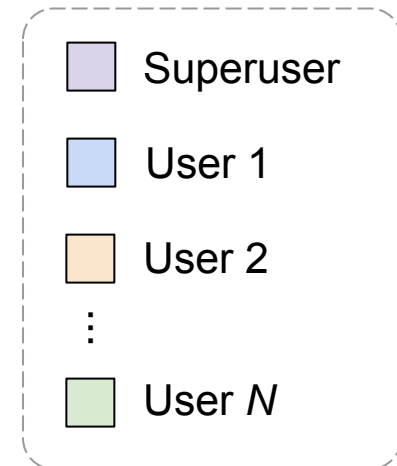
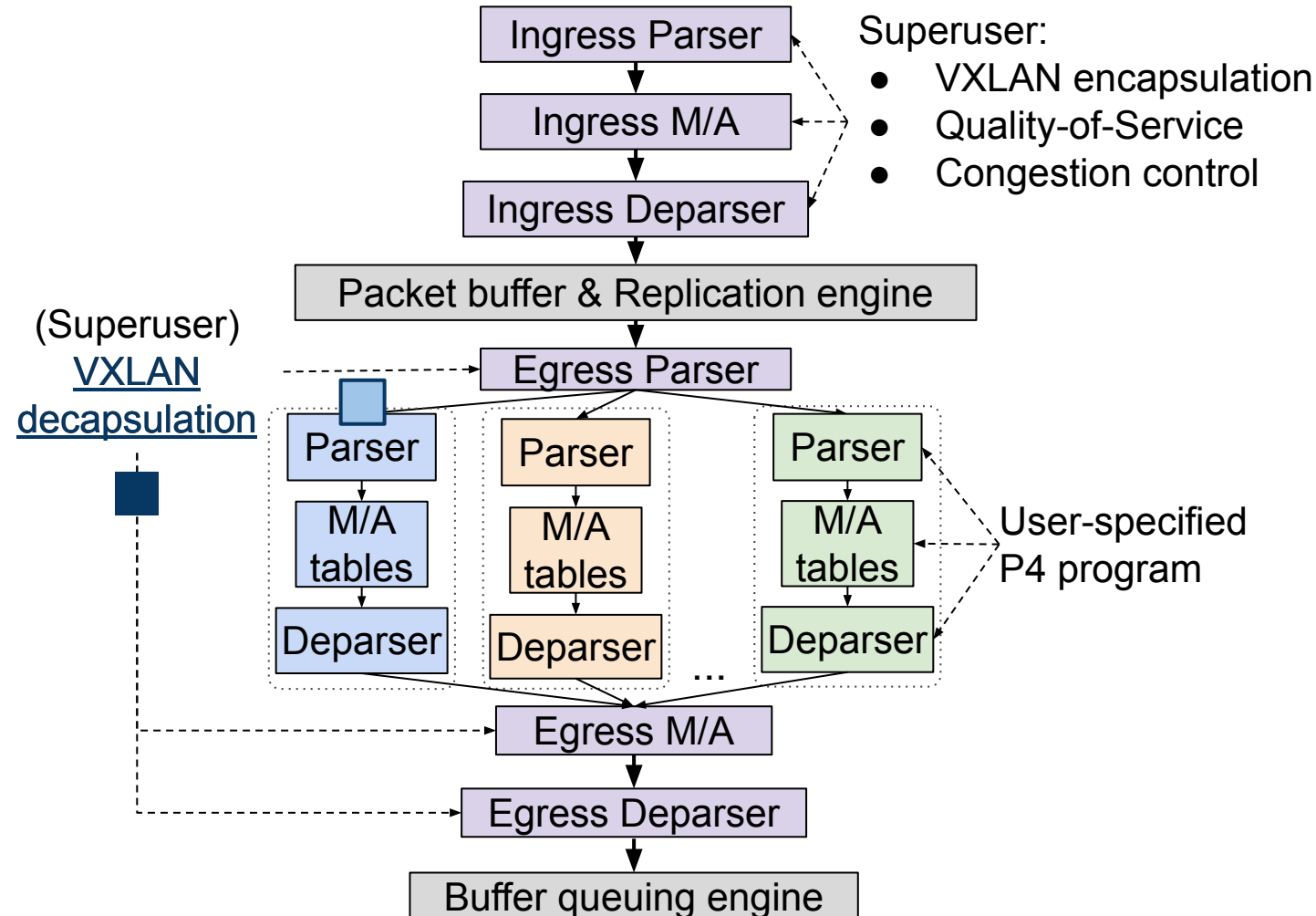


# MTPSA Packet Flow

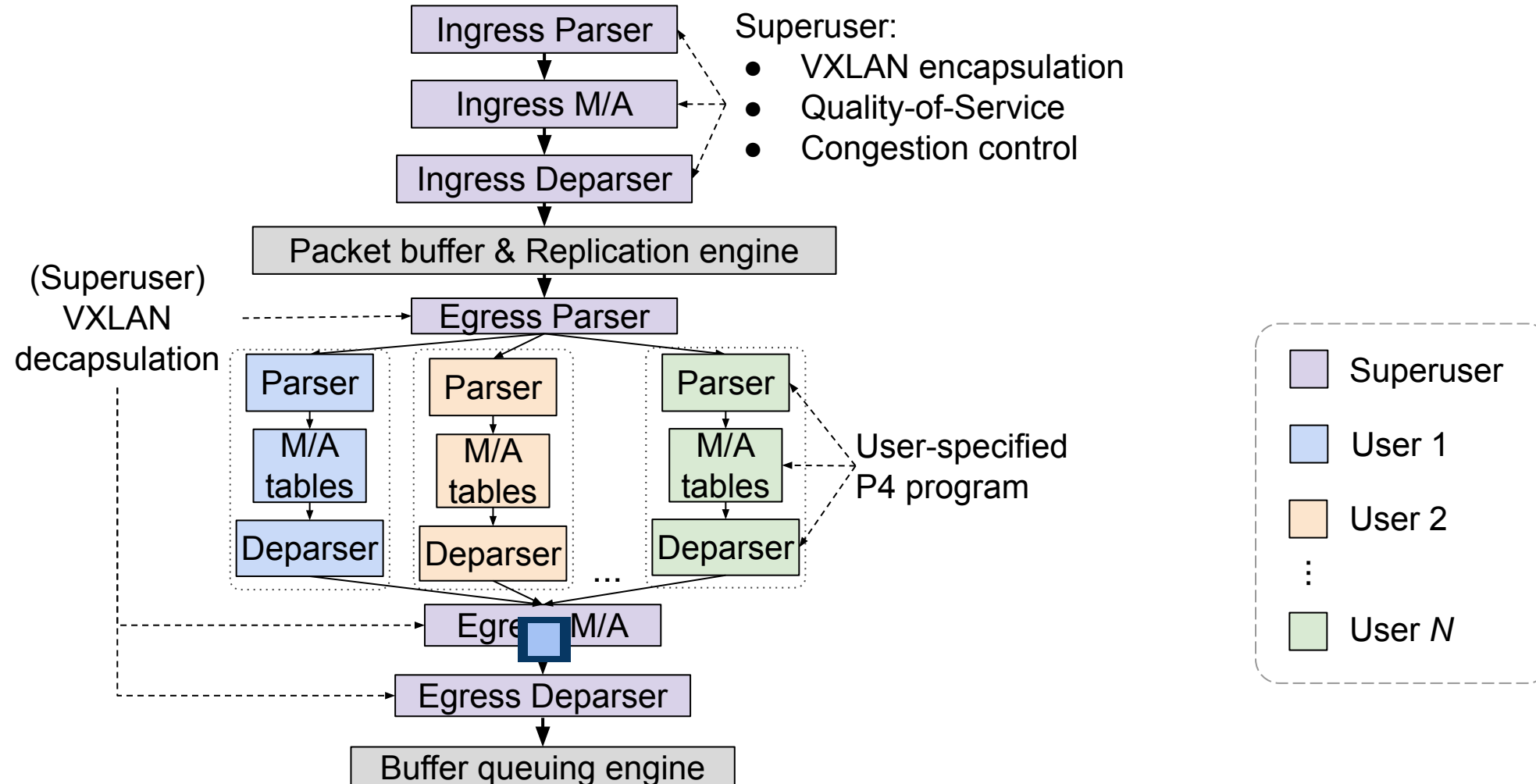




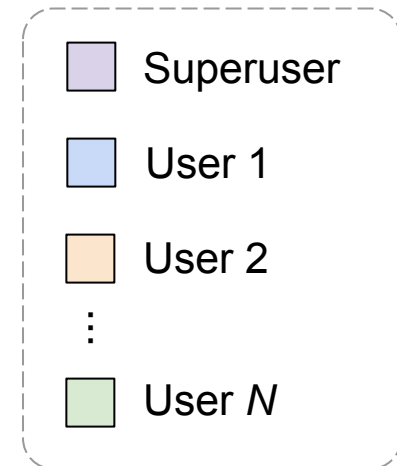
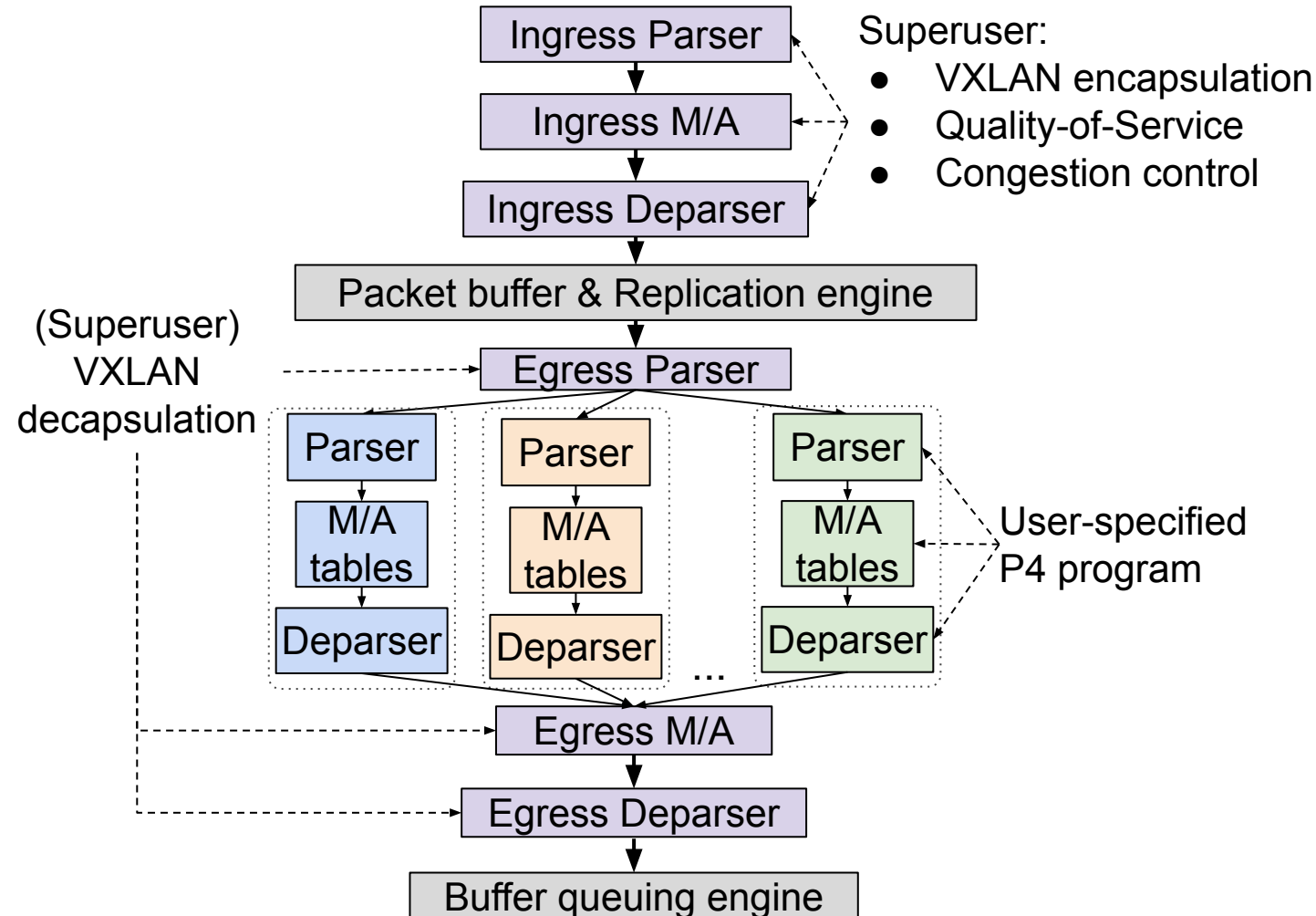
# MTPSA Packet Flow



# MTPSA Packet Flow



# MTPSA Packet Flow

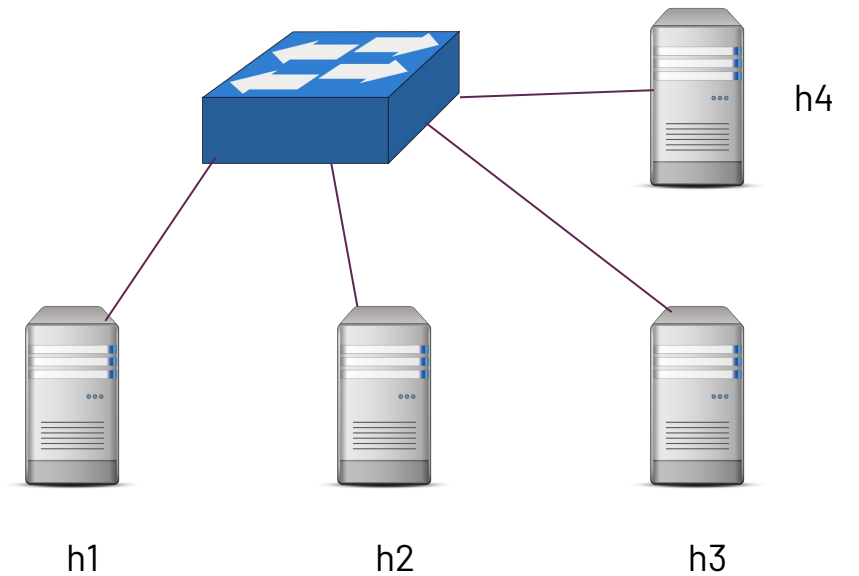


# MTPSA Live Demo 1

Topology: 4x Hosts + 1x Switch

User 1: Load balancer

User 2: P4 Calculator



# User Permissions

## Superuser Ingress

```
action set_user_id(bit<16> user_id) {  
    ostd.user_id = user_id;
```

```
    if (user_id == 3) {  
        // Disable counter extern in User 3  
        ostd.user_permissions = 0x01;  
    }  
}
```

```
table set_user_id_table {  
    key = {  
        istd.ingress_port: exact;  
    }  
    actions = {  
        set_user_id;  
        drop;  
    }  
    size = 1024;  
    default_action = drop();  
}
```

# User Permissions

## Superuser Ingress

Superuser defines permissions given to users to perform certain operations

```
action set_user_id(bit<16> user_id) {  
    ostd.user_id = user_id;
```

```
    if (user_id == 3) {  
        // Disable counter extern in User 3  
        ostd.user_permissions = 0x01;  
    }  
}
```

```
table set_user_id_table {  
    key = {  
        istd.ingress_port: exact;  
    }  
    actions = {  
        set_user_id;  
        drop;  
    }  
    size = 1024;  
    default_action = drop();  
}
```

# User Permissions

## Superuser Ingress

```
action set_user_id(bit<16> user_id) {
    ostd.user_id = user_id;
    if (user_id == 3) {
        // Disable counter extern in User 3
        ostd.user_permissions = 0x01;
    }
}

table set_user_id_table {
    key = {
        istd.ingress_port: exact;
    }
    actions = {
        set_user_id;
        drop;
    }
    size = 1024;
    default_action = drop();
}
```

\$ mtpsa\_switch\_CLI

```
RuntimeCmd: switch_context 1
Obtaining JSON from switch...
Done
RuntimeCmd: counter_read ingress.port_data 0
ingress.port_data[0]= (4042 bytes, 28 packets)
```

Users can read  
only their own  
counters

# User Permissions

## Superuser Ingress

```
action set_user_id(bit<16> user_id) {
    ostd.user_id = user_id;
    if (user_id == 3) {
        // Disable counter extern in User 3
        ostd.user_permissions = 0x01;
    }
}

table set_user_id_table {
    key = {
        istd.ingress_port: exact;
    }
    actions = {
        set_user_id;
        drop;
    }
    size = 1024;
    default_action = drop();
}
```

\$ mtpsa\_switch\_CLI

```
RuntimeCmd: switch_context 1
Obtaining JSON from switch...
Done
RuntimeCmd: counter_read ingress.port_data 0
ingress.port_data[0]= (4042 bytes, 28 packets)
```

Users can read  
only their own  
counters

```
RuntimeCmd: switch_context 3
Obtaining JSON from switch...
Done
RuntimeCmd: counter_read ingress.port_data 0
ingress.port_data[0]= (0 bytes, 0 packets)
```

User 3 does not  
have sufficient  
permissions to  
use counters



# MTPSA Live Demo 2

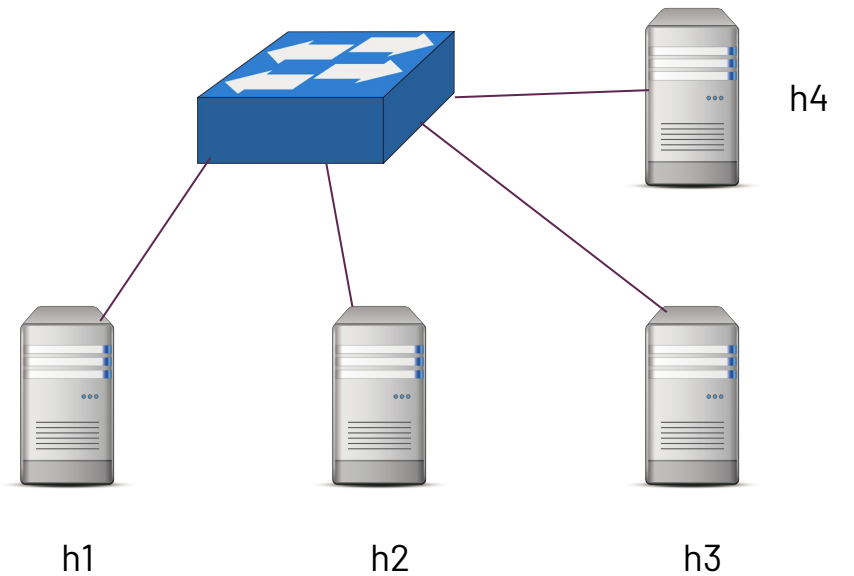
Topology: 4x Hosts + 1x Switch

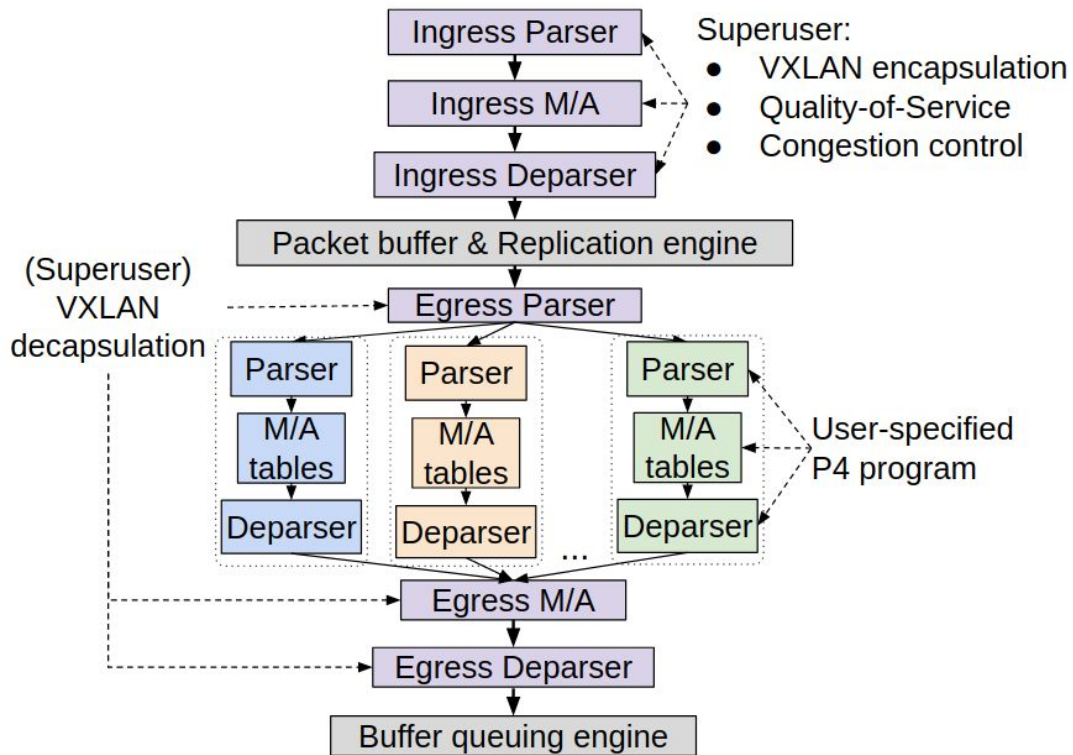
User 1: L3 Forwarding + Counter

User 2: L3 Forwarding

User 3: L3 Forwarding + Counter

User 4: L3 Forwarding





# Summary & Questions?

<https://github.com/mtpsa>

<https://eng.ox.ac.uk/computing/projects/in-network-computing/mtpsa>