

# neat

## Towards a Flexible Internet Transport Layer Architecture

Karl-Johan Grinnemo, **Tom Jones**, Gorry Fairhurst,  
David Ros, Anna Brunstrom and Per Hurtig

University of Aberdeen,  
Aberdeen, U.K.

{tom, gorry}  
@erg.abdn.ac.uk

Karlstad University,  
Karlstad, Sweden

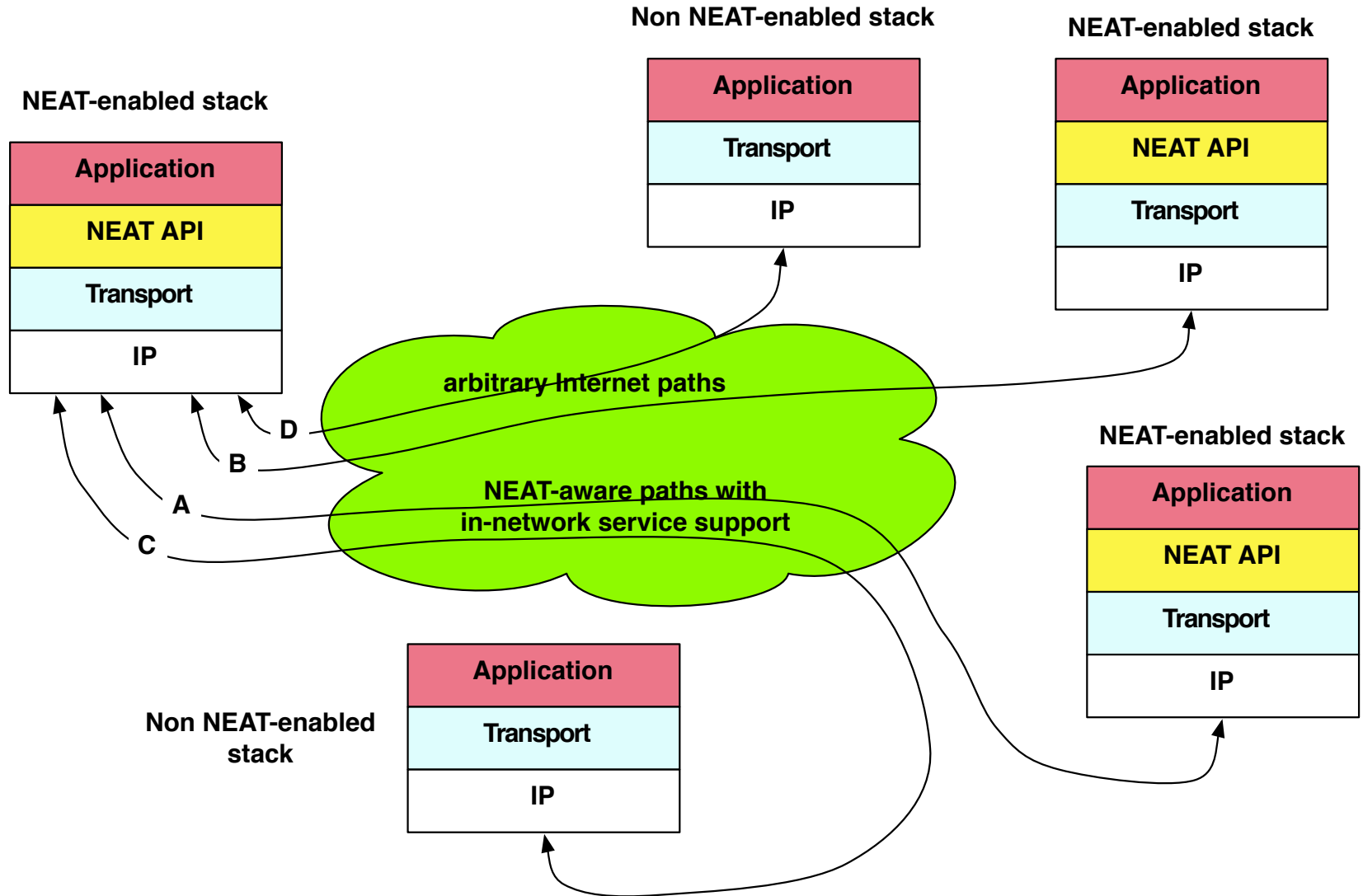
{karl-johan.grinnemo,  
anna.brunstrom,  
per.hurtig}@kau.se

Simula Research  
Laboratory, Oslo,  
Norway

dros@simula.no



# The Network has Ossified



# The API has Ossified

```
getaddrinfo(); // Look up host
socket();      // Create a socket

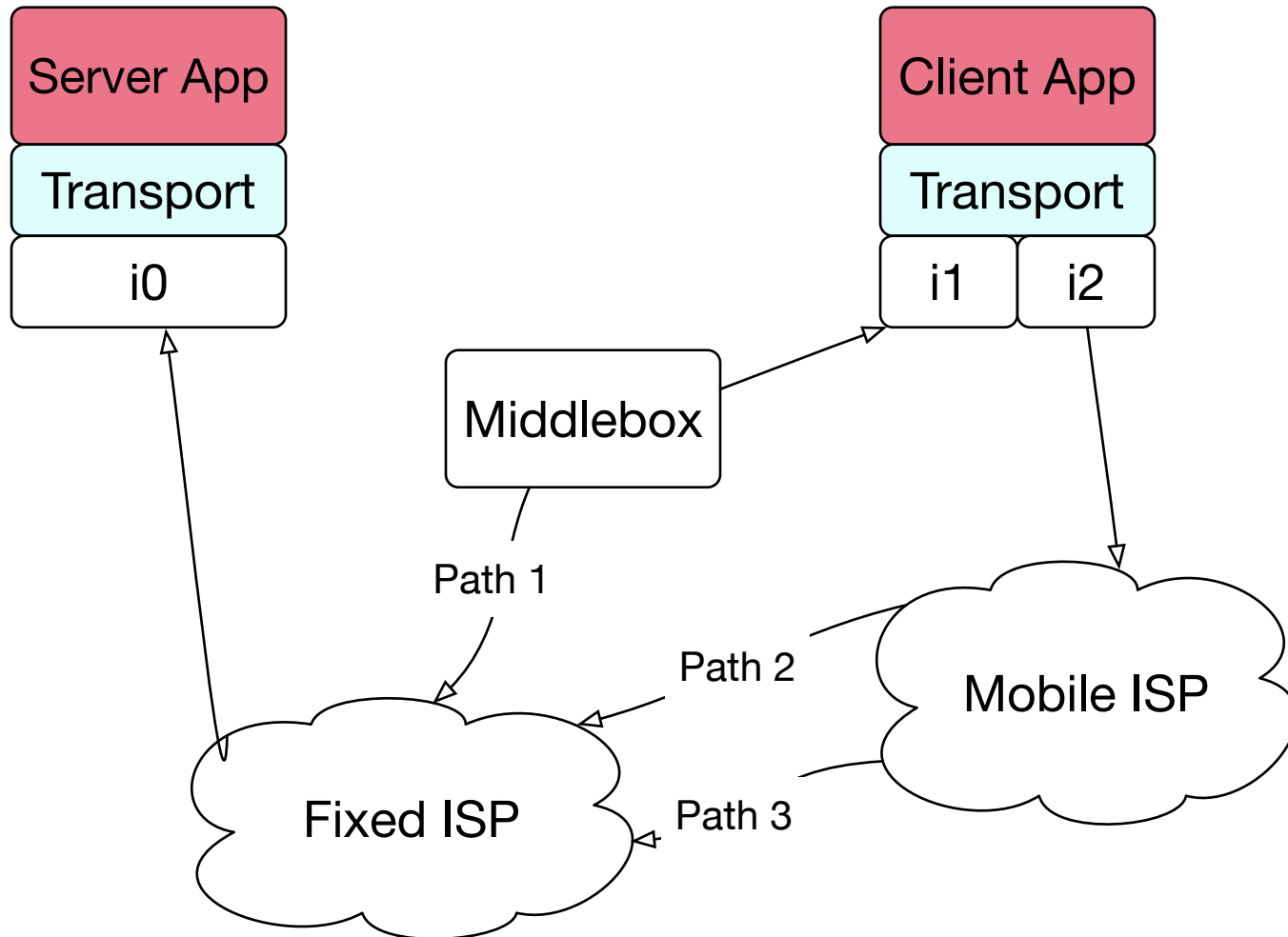
setsockopt(); // Configure the socket
getsockopt(); // Check parameters

connect();    // Start connection

send();
recv();
```



# Fixing Interface Selection



# NEAT Application

```
static struct neat_flow_operations ops;  
static struct neat_ctx *ctx = NULL;  
static struct neat_flow *flow = NULL;
```

```
ctx = neat_init_ctx()  
flow = neat_new_flow(ctx)
```

```
prop = NEAT_PROPERTY_UDP_REQUIRED | NEAT_PROPERTY_IPV6_REQUIRED;  
neat_set_property(ctx, flow, &prop)
```

```
ops.on_writable = on_writable;  
ops.on_readable = on_readable;  
ops.on_error = on_error;
```

```
neat_set_operations(ctx, flow, &ops)  
neat_open(ctx, flow, argv[argc - 2], argv[argc - 1])
```

```
neat_start_event_loop(ctx, NEAT_RUN_DEFAULT);
```



# NEAT Application

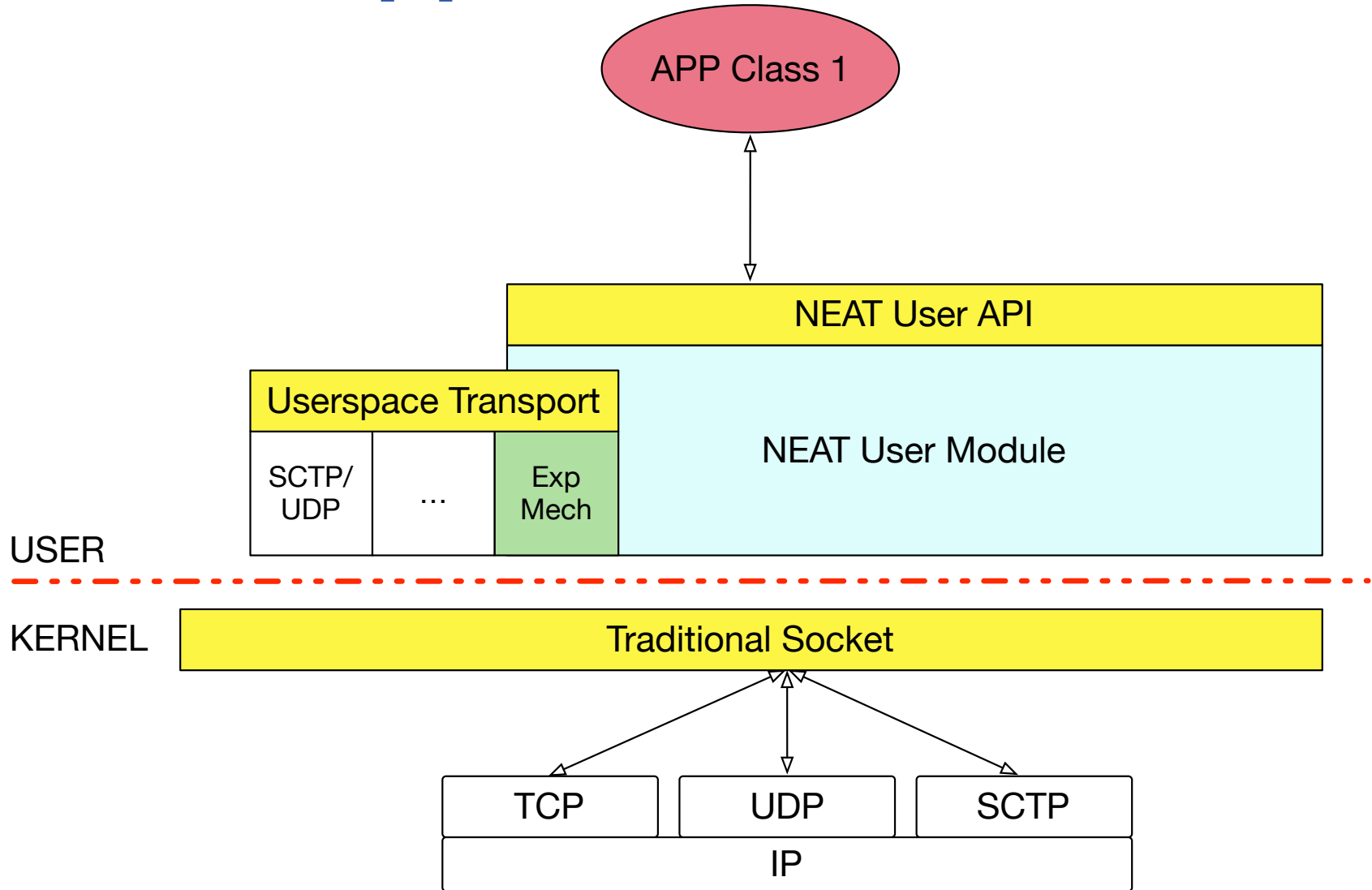
```
static neat_error_code
on_writable(struct neat_flow_operations *opCB)
{
    neat_write(opCB->ctx, opCB->flow, buf)
    return NEAT_OK;
}
```

```
static neat_error_code
on_readable(struct neat_flow_operations *opCB)
{
    neat_read(opCB->ctx, opCB->flow, buf)
    return NEAT_OK;
}
```

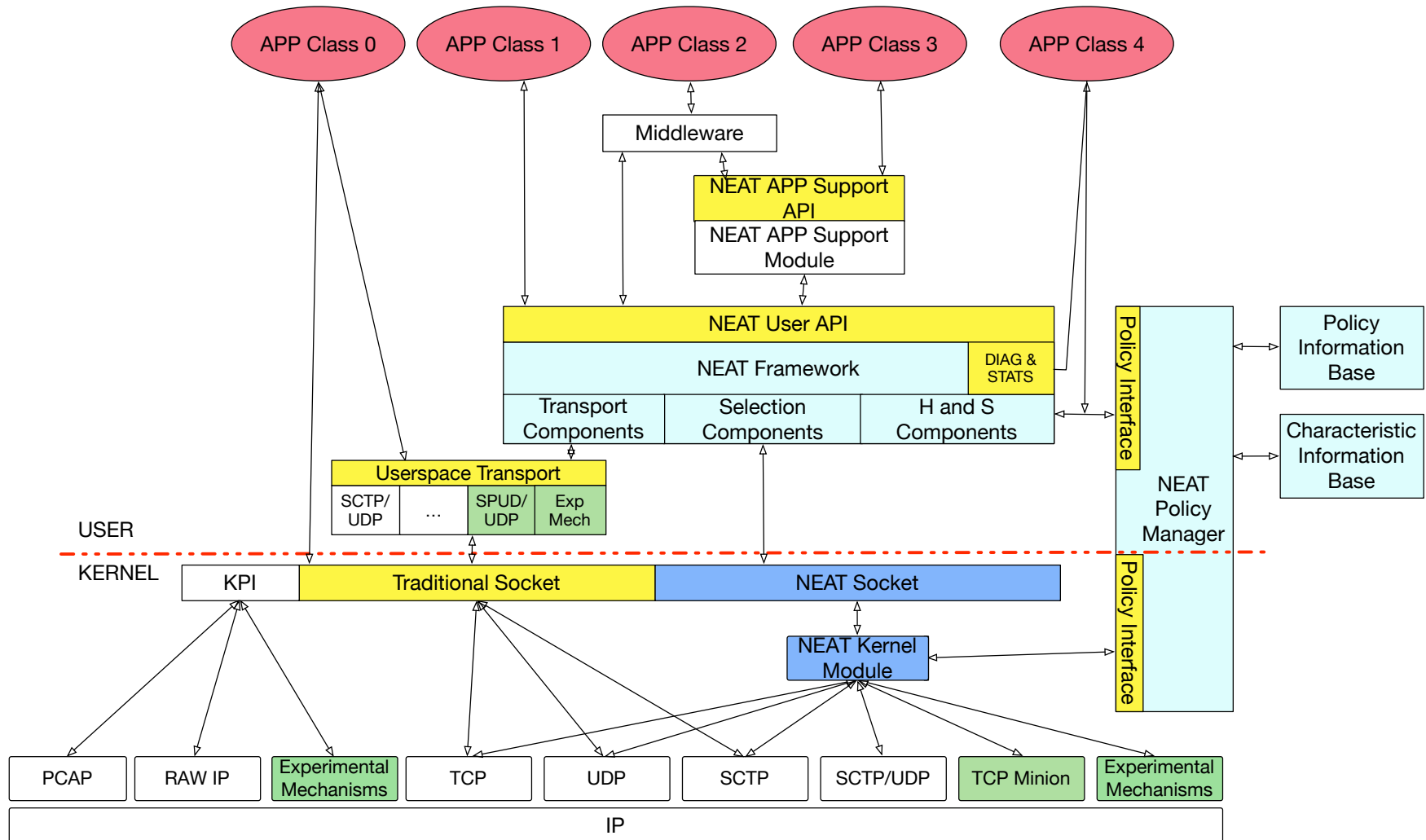
<https://github.com/NEAT-project/neat/blob/master/examples/client.c>



# NEAT Application

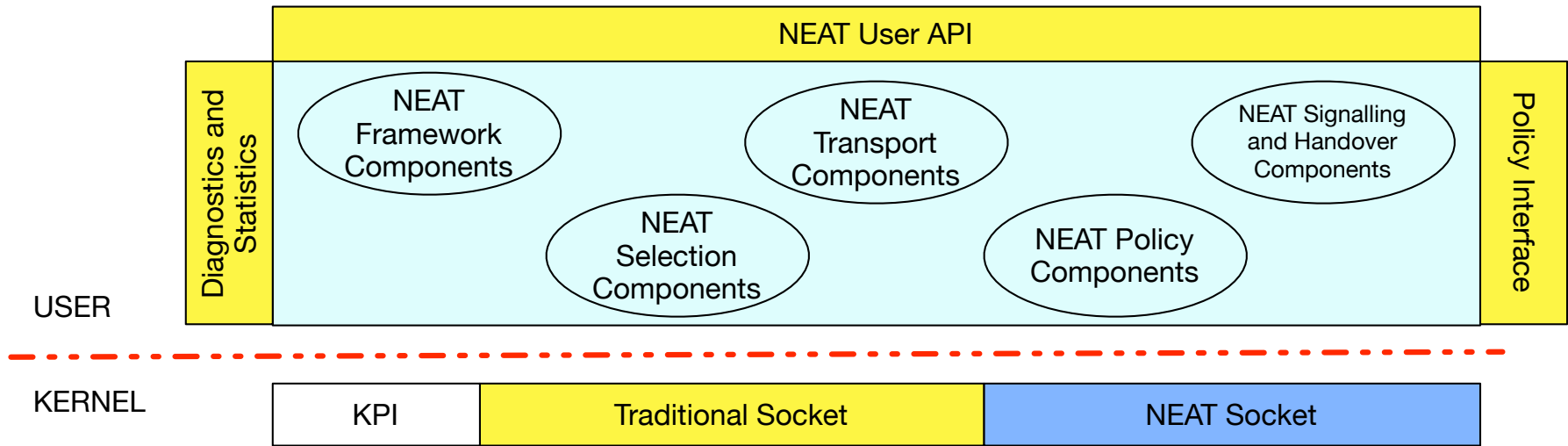


# The NEAT System



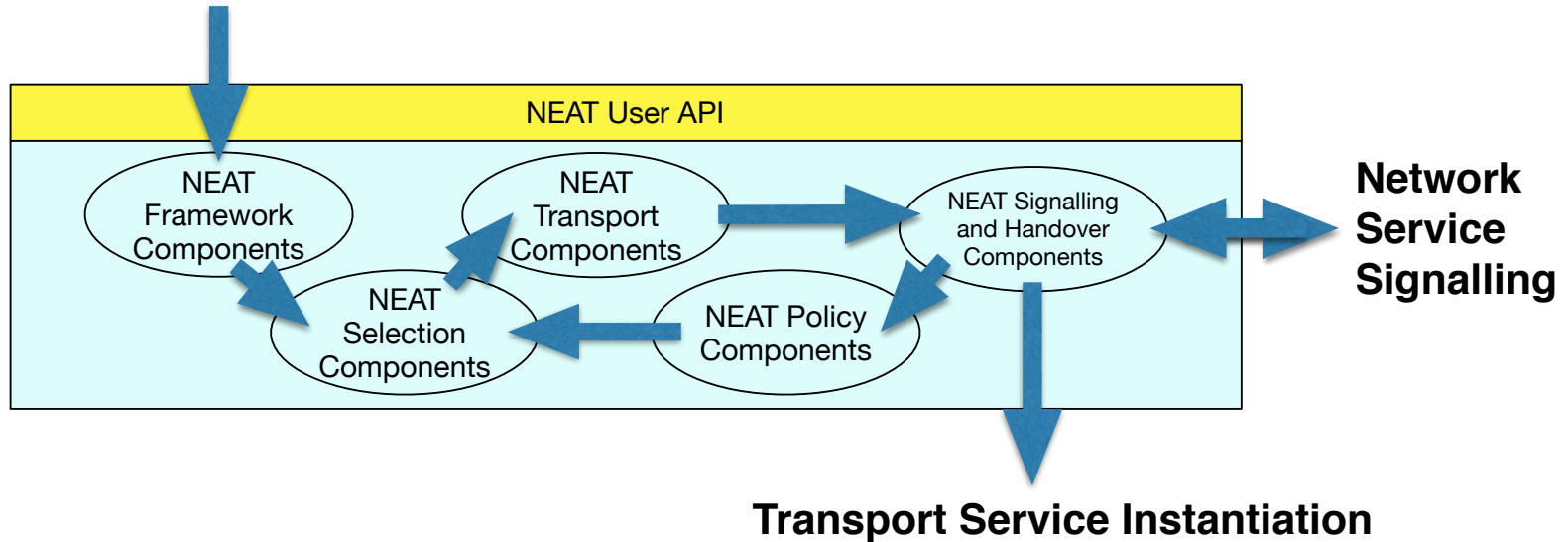


# The NEAT User Module



# The NEAT User Module

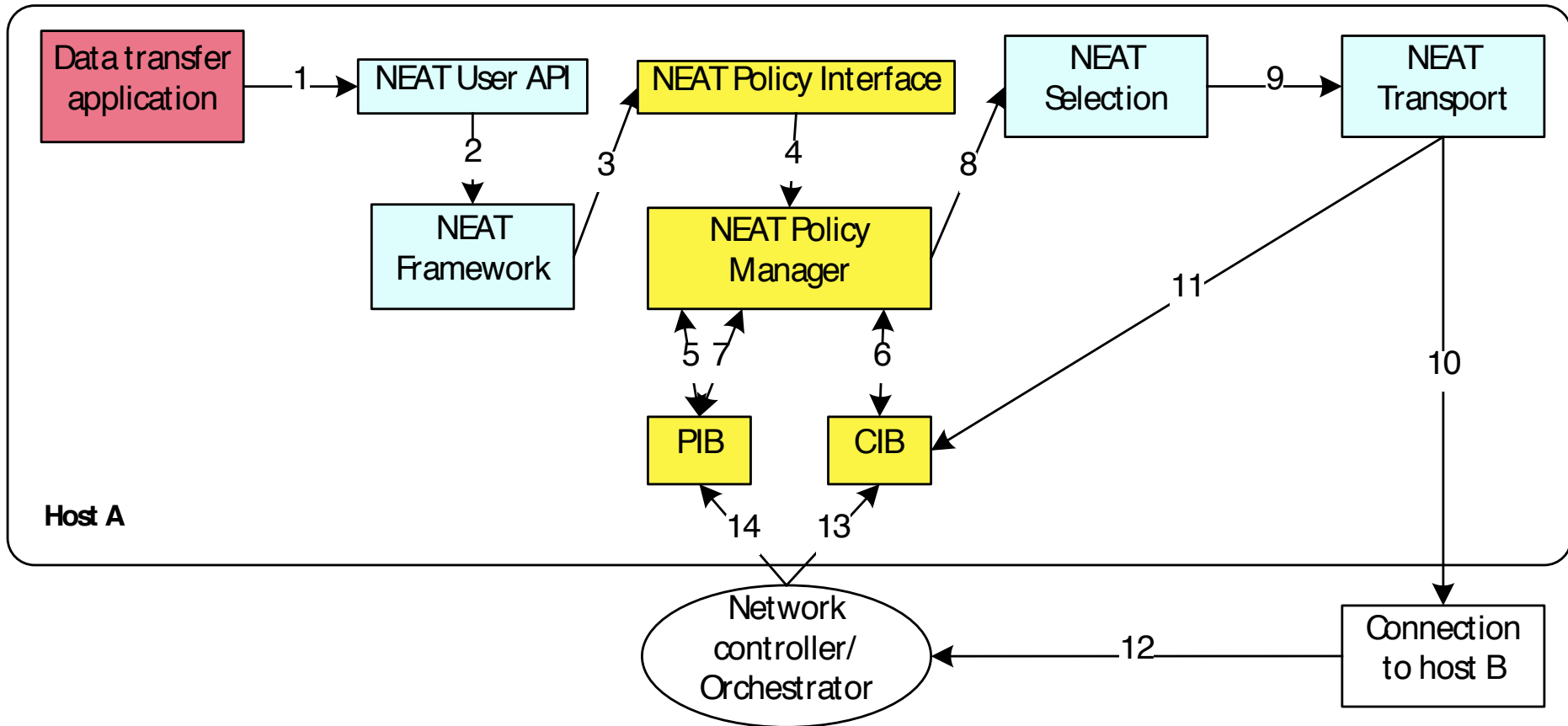
**NEAT Flow Endpoint**



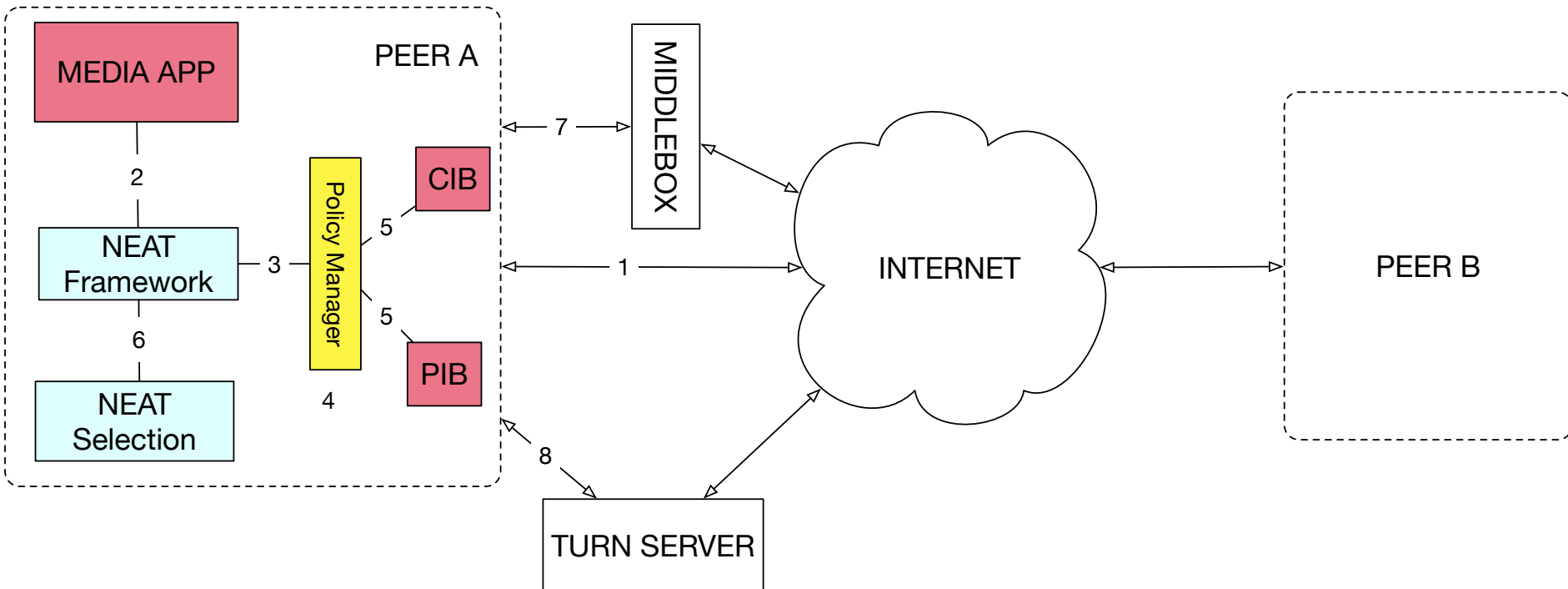
## 5 Groups of components:

- ▶ NEAT Framework Component: API, Logic
- ▶ NEAT Selection Components: Choose candidates
- ▶ NEAT Policy Components: Policy and Characteristics
- ▶ NEAT Transport Components: Instantiate transports
- ▶ NEAT Handover and Signalling Components

# NEAT Example: Transfer Application in SDN Environment



# NEAT Example: Multimedia Application



# Resources

<https://neat-project.org>

<https://github.com/neat-project>

# Thank you for listening

## Questions

<https://neat-project.org>

<https://github.com/neat-project>

[tom@erg.abdn.ac.uk](mailto:tom@erg.abdn.ac.uk)

**neat**

NEAT is funded by the European Union's Horizon 2020 research and innovation programme under grant agreement no. 644334.

