

# Reserved Name Processing Core

Scheduler: Round-Robin, Priority Queues, Tree Flavours

Scheduler Actors: Features, Timers, Async I/O

Streams Backends: Zero-copy, Message Passing

Linear Backends: Async I/O Disk Streams, Network Streams

Indexed Backends: Timers, Actors

Backpressured Message Bus/Buffers: Arc/Vec prealloc

Class: Low Latency, Real Time

# Respect Kernels History

Richard Rashid. Mach 3. NUMA, Bus Oriented Components  
Dave Cutler. Windows NT. True Async I/O on IoCompletionPort  
BeOS. Travis Geiselbrecht. SMP Scaling of OS services  
Microkernels: RT, Tiny codebases eCos/TRON, QNX, VxWorks  
Unikernels: Erlang, Mirage, HaLVM

## FOUNDATION

## Stream/List duality

```
pub enum List<Message> {  
    Nil,  
    Cons(Message, std::marker::PhantomData<List<Message>>) }
```

```
pub struct Stream<Message> {  
    head: Message,  
    tail: Box<Stream<Message>> }
```

```
pub trait Stream<Message> {  
    fn head(&mut self) -> Message;  
    fn tail(&mut self) -> Stream<Message>; }
```

## FUTURES

# Zero-copy and Message Passing

```
pub trait Future<Message,Error> {  
    fn poll(&mut self) -> Result<Message,Error>;  
    fn tail(&mut self) -> Future<Message,Error>; }  

```

```
pub trait Process<Protocol, State, Error> {  
    fn state(&mut self) -> State;  
    fn send(&mut self, Protocol) -> Result<State, Error>; }  

```

```
pub trait Discipline<Stream<Message>> {  
    fn select(&mut self, u64) -> Stream<Message>; }  

```

## ORDERS

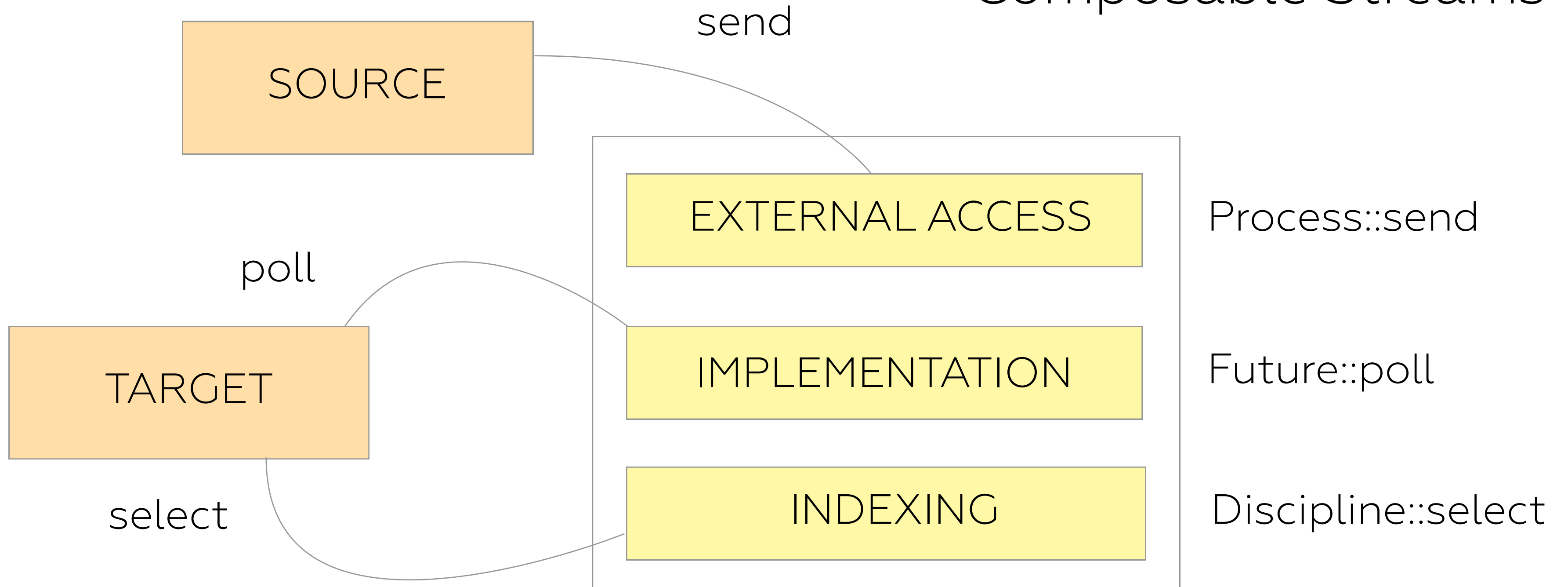
## Order Processing Protocol

```
pub struct OrderState {  
    state: OrderStatus,  
    id: ID,  
    price: Price,  
    size: Size,  
    side: Side, }
```

```
pub enum OrderProtocol {  
    Request, Execute, Reject,  
    UnsolicitedCancel,  
    Cancel,  
    Ack, Replace, }
```

```
type OrderProcess = Process<OrderProtocol, OrderState, Error>;
```

# Composable Streams



# Polymorphic Disciplines

SCHED

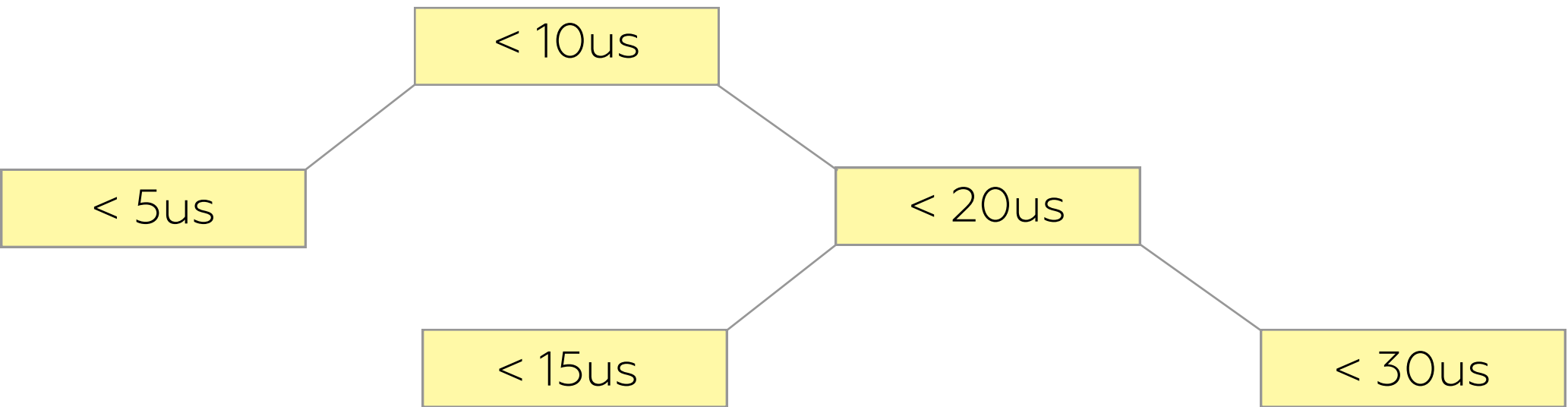
DISK

Discipline::select

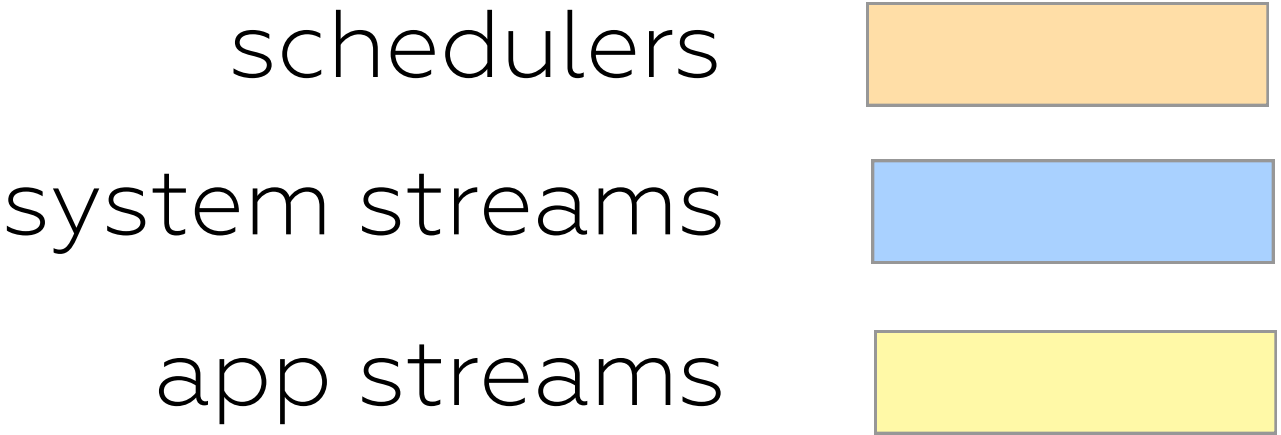
Circular Buffers

TIMERS

Discipline::select



Linear: MQ, EXT, DISK, NET  
Trees: TIMERS  
Priority Queues: TASKS, IRQ



# Node Components

