咕泡教育
—gupaoedu—

高性能通信框架之Netty

GUPAO EDU

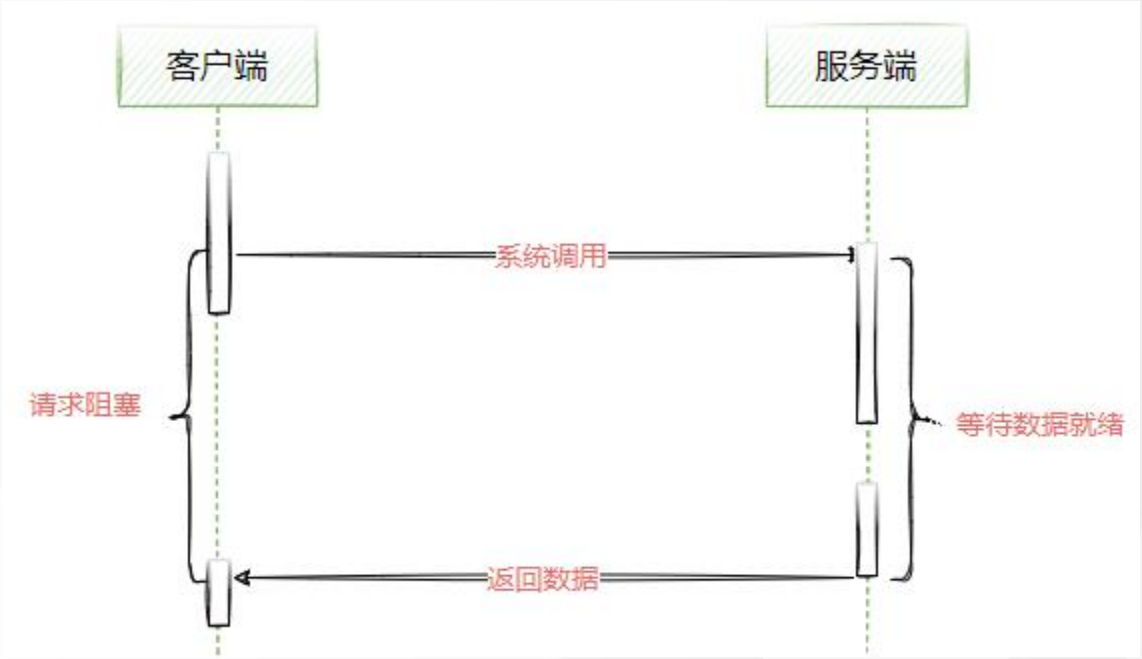
讲师：Mic

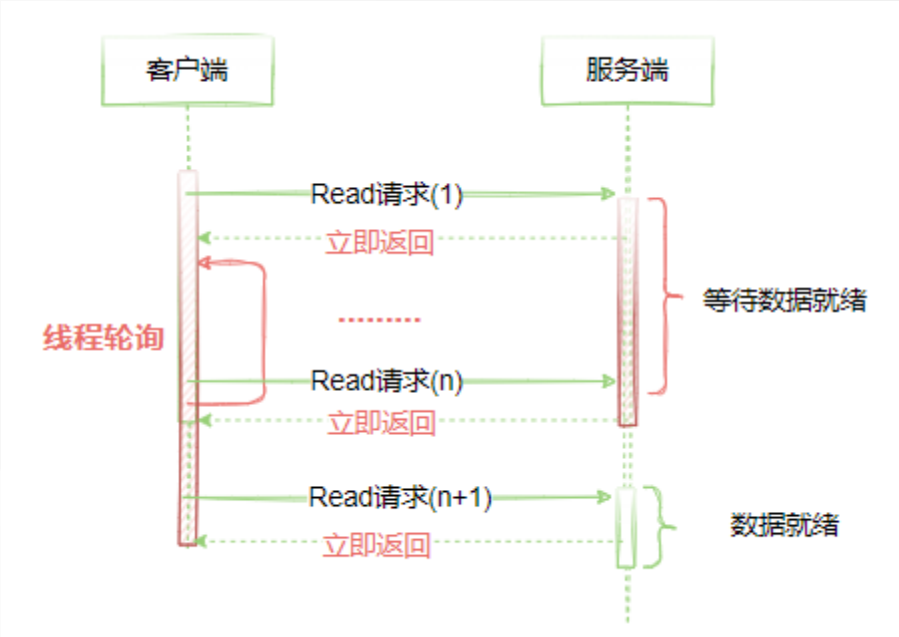
我们的愿景
推动每一次人才升级

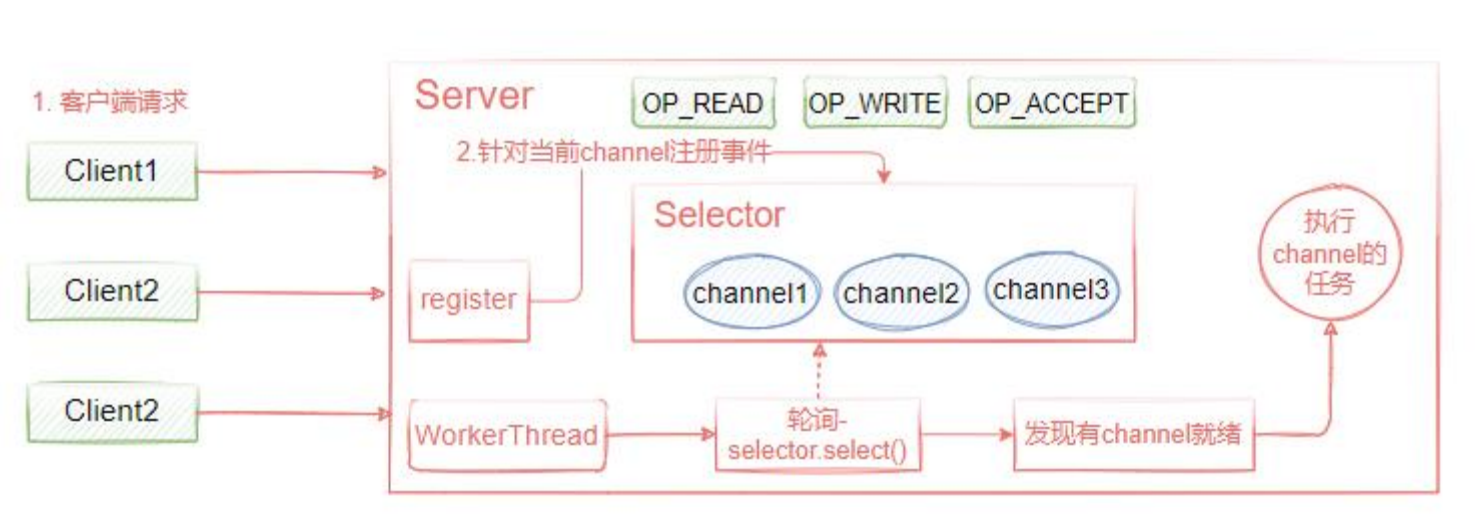
我们的使命
让每个人的职业生涯不留遗憾

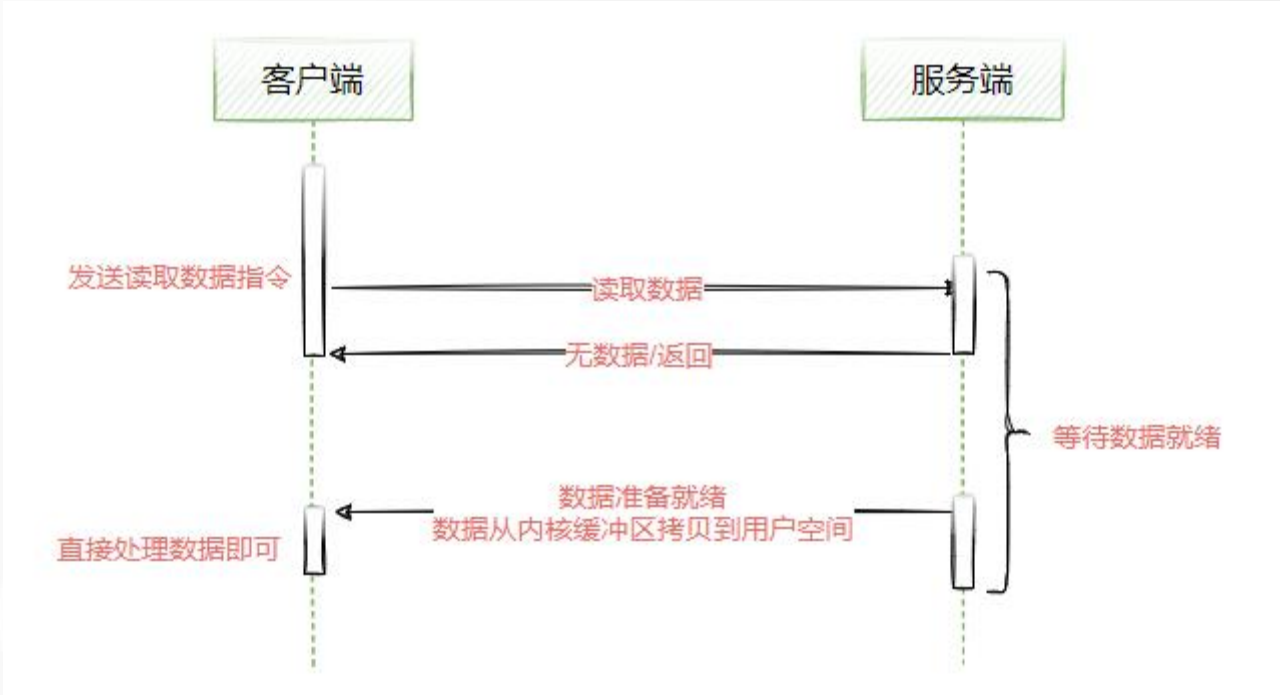
本节课程内容安排

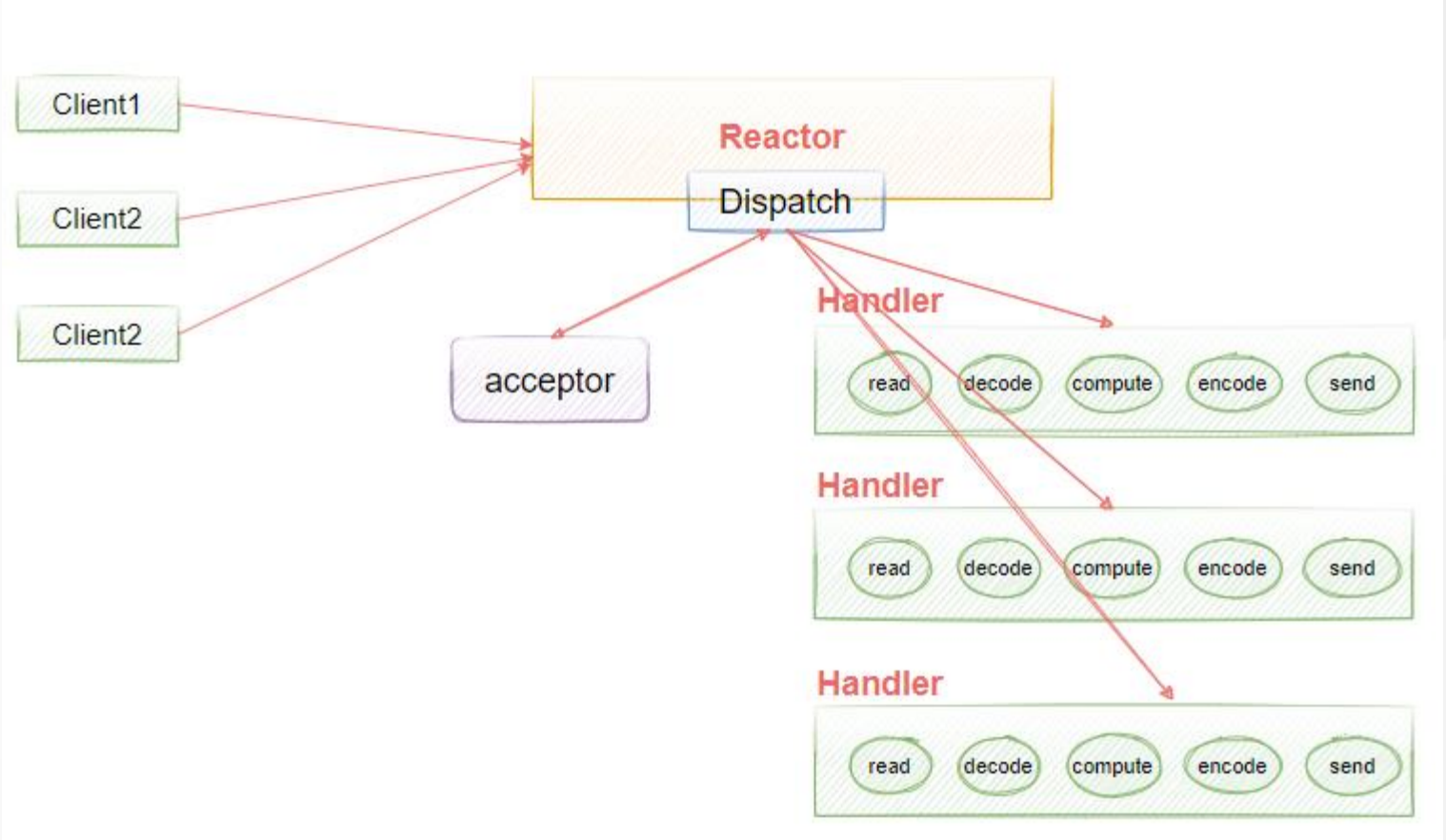
1. 从IO的发展理解Netty
2. 为什么选择Netty
3. Netty的生态
4. Netty的基本应用
5. Netty工作机制分析

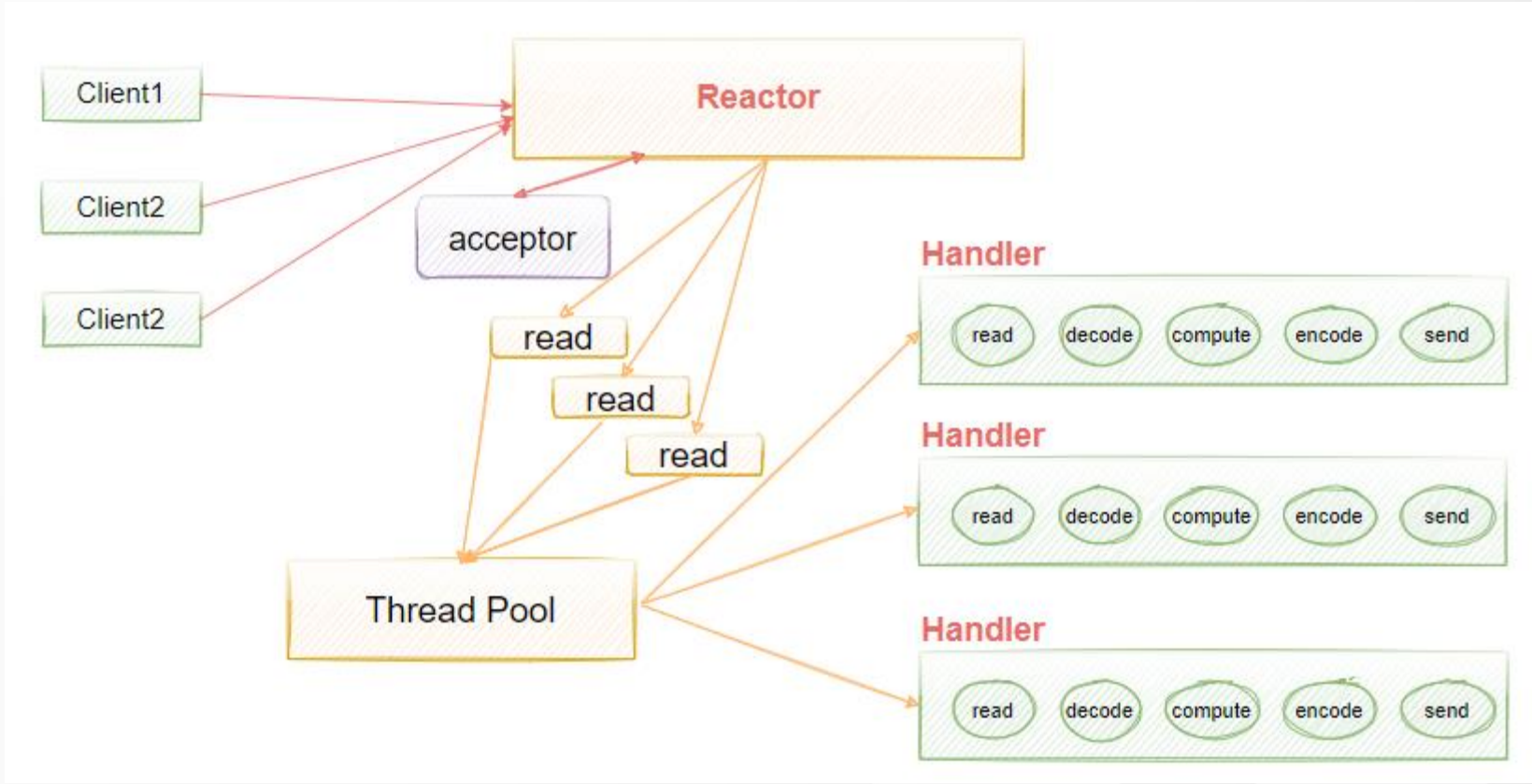

客户端
服务端
系统调用
请求阻塞
等待数据就绪
返回数据

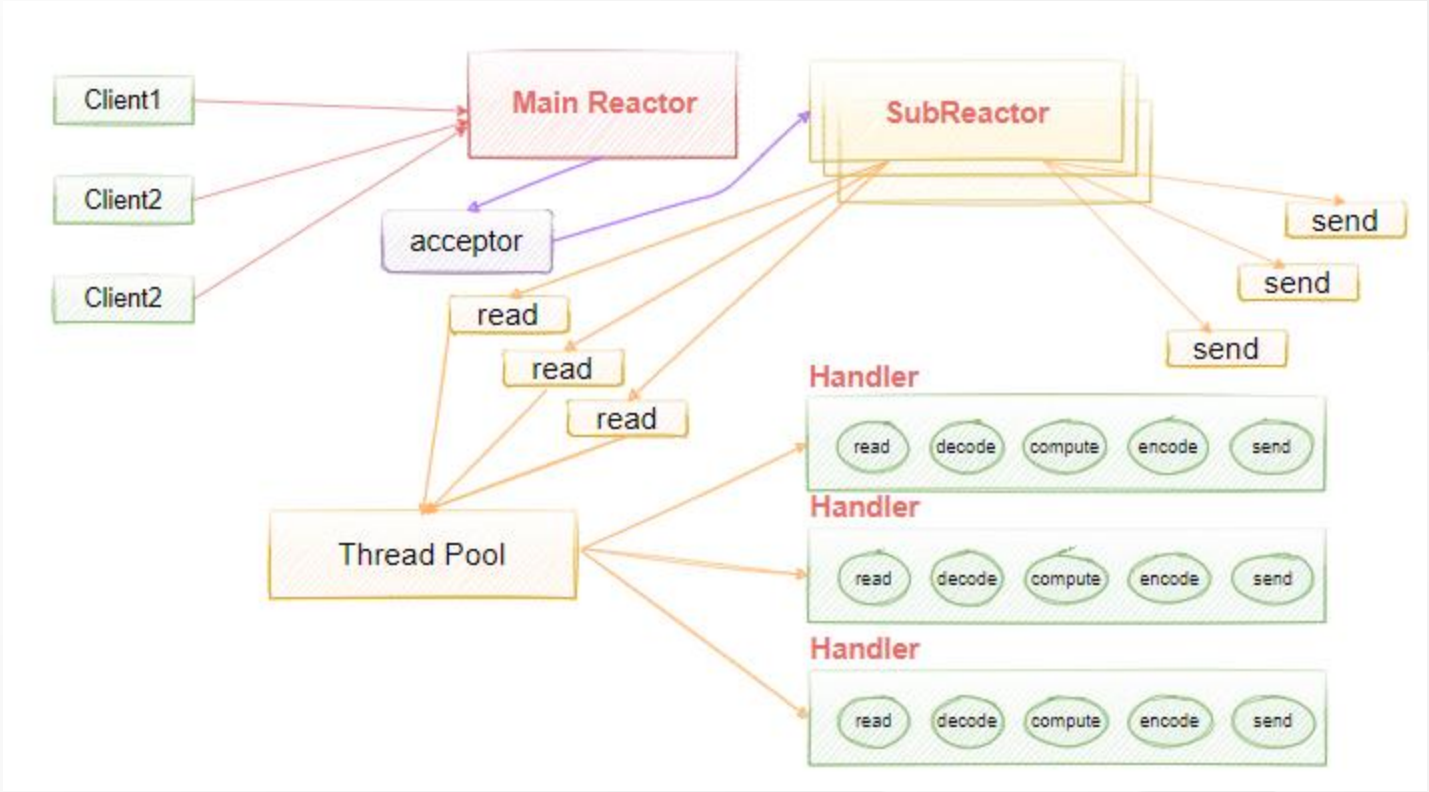

客户端
服务端
Read请求(1)
立即返回
线程轮询
.....
等待数据就绪
Read请求(n)
立即返回
Read请求(n+1)
立即返回
数据就绪


1. 客户端请求
Client1
Client2
Client2
Server
OP_READ
OP_WRITE
OP_ACCEPT
2.针对当前channel注册事件
register
Selector
channel1
channel2
channel3
WorkerThread
轮询-
selector.select()
发现有channel就绪
执行
channel的
任务

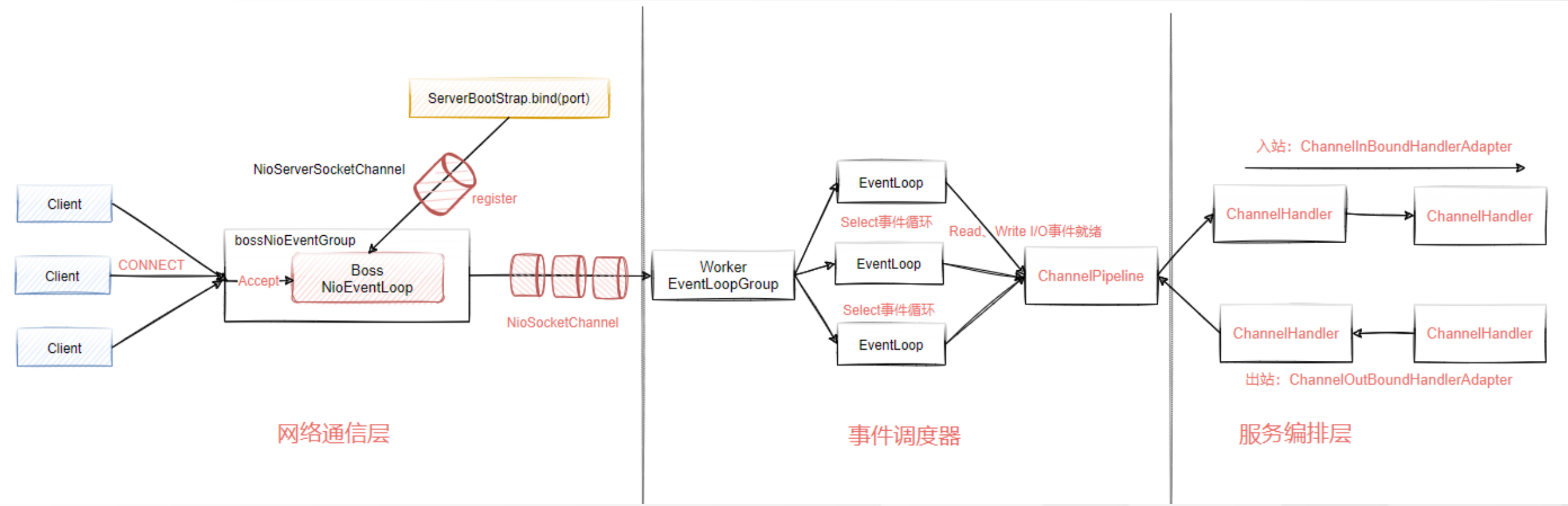

客户端
服务端
发送读取数据指令
读取数据
无数据/返回
等待数据就绪
数据准备就绪
数据从内核缓冲区拷贝到用户空间
直接处理数据即可

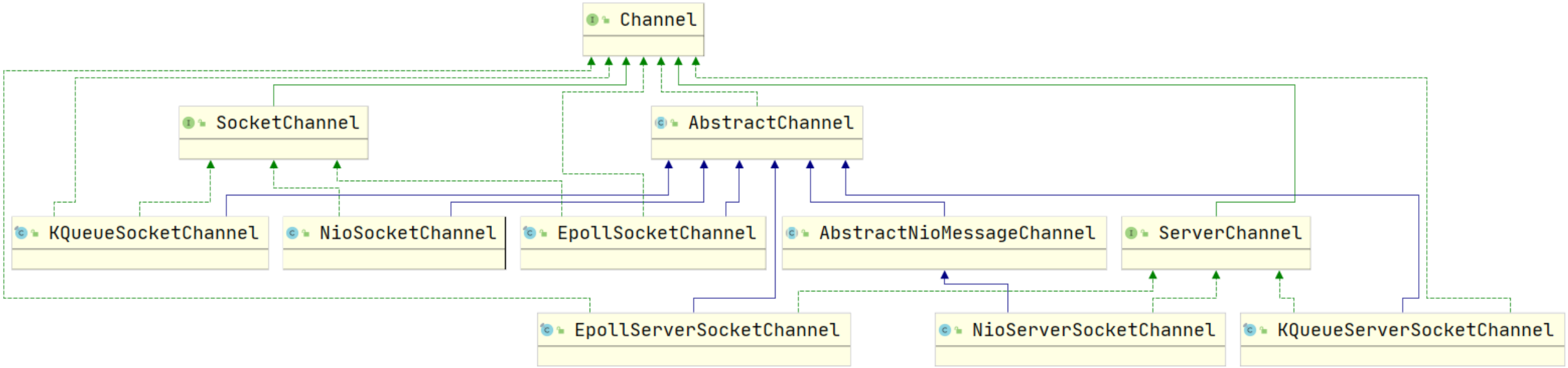

Client1
Client2
Client2
Reactor
Dispatch
acceptor
Handler
read
decode
compute
encode
send
Handler
read
decode
compute
encode
send
Handler
read
decode
compute
encode
send

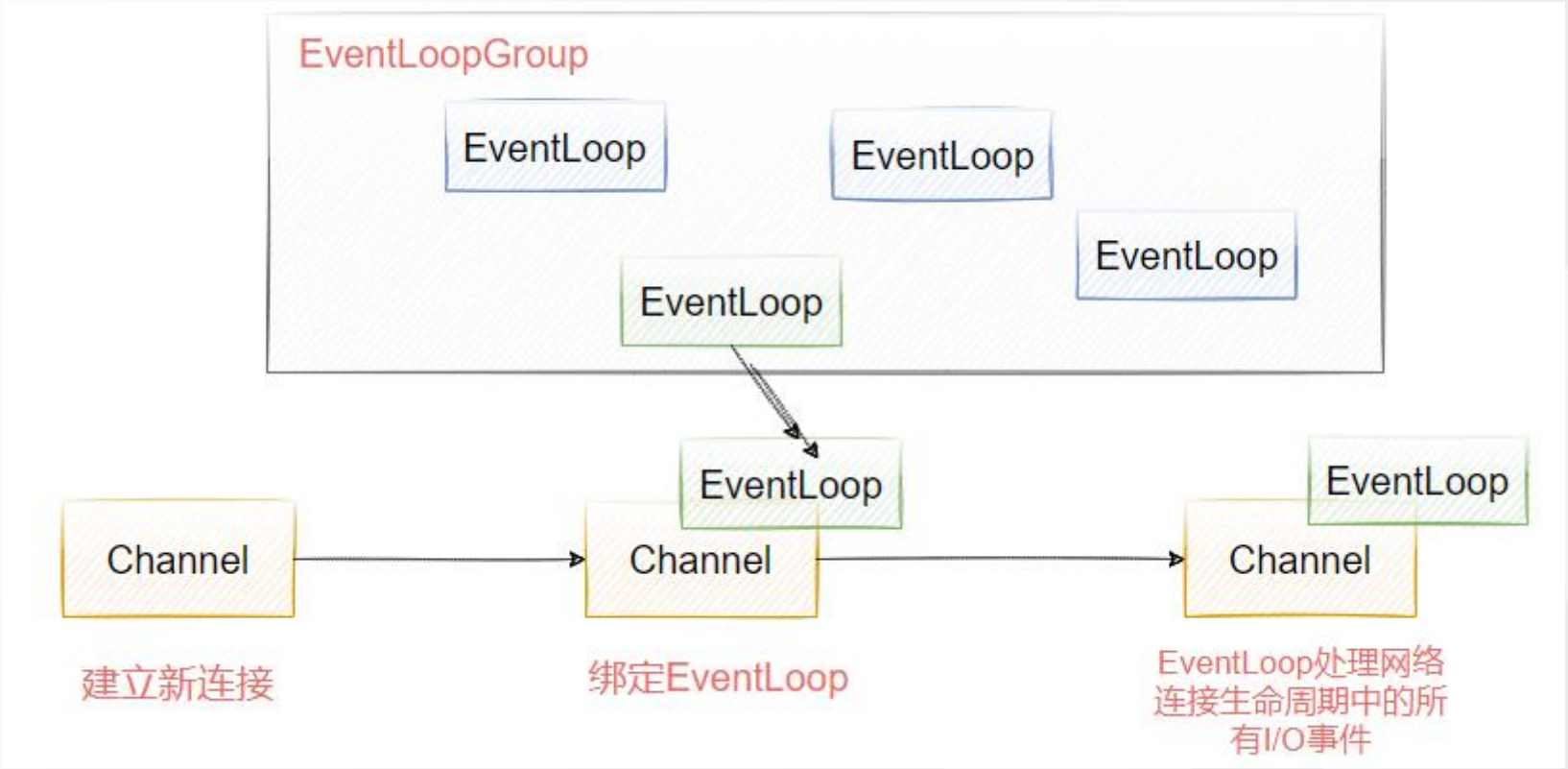

Client1
Client2
Client2
Reactor
acceptor
read
read
read
Thread Pool
Handler
read
decode
compute
encode
send
Handler
read
decode
compute
encode
send
Handler
read
decode
compute
encode
send

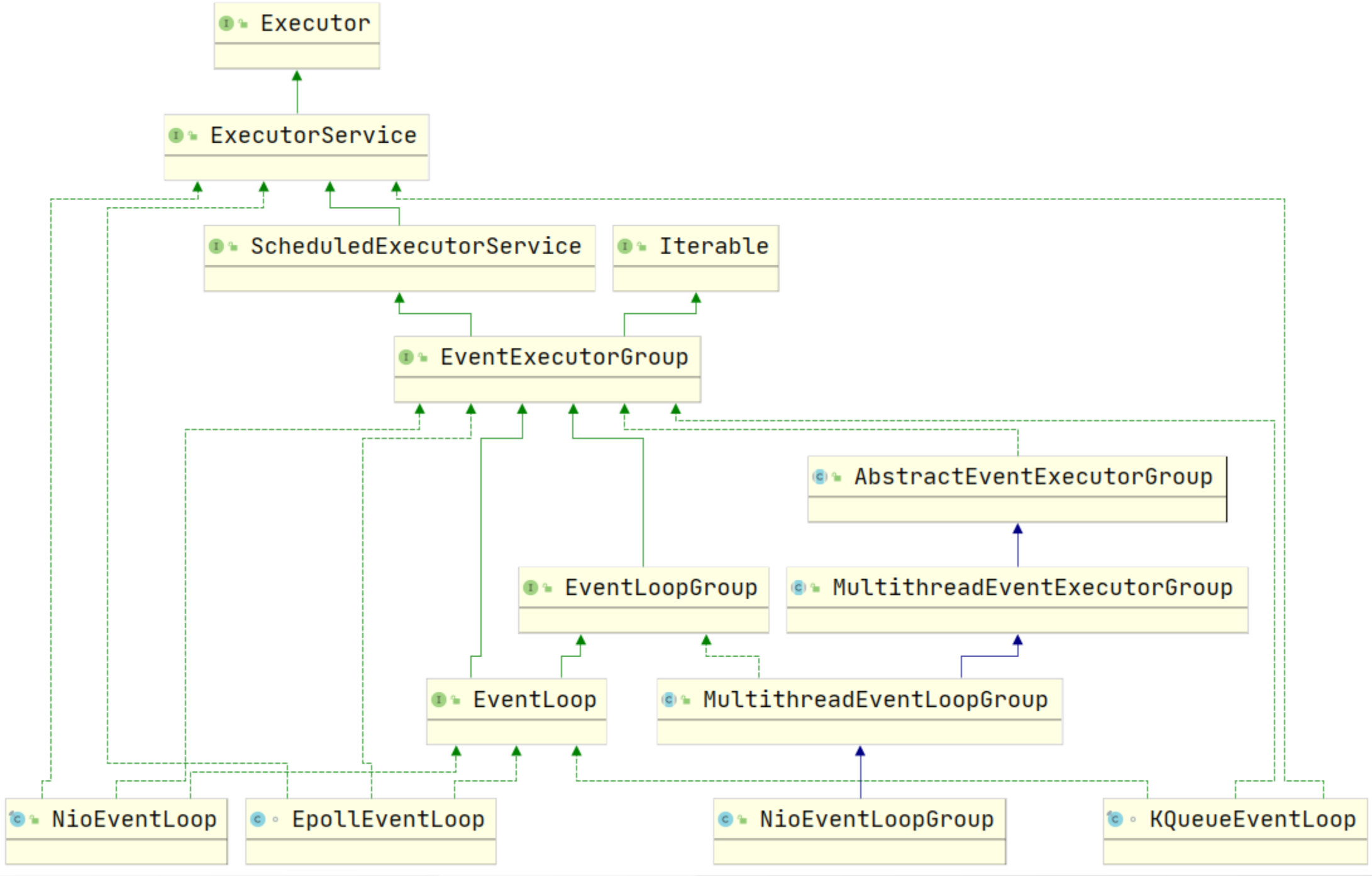

Client1
Client2
Client2
Main Reactor
acceptor
SubReactor
read
read
read
send
send
send
Handler
read
decode
compute
encode
send
Handler
read
decode
compute
encode
send
Handler
read
decode
compute
encode
send
Thread Pool

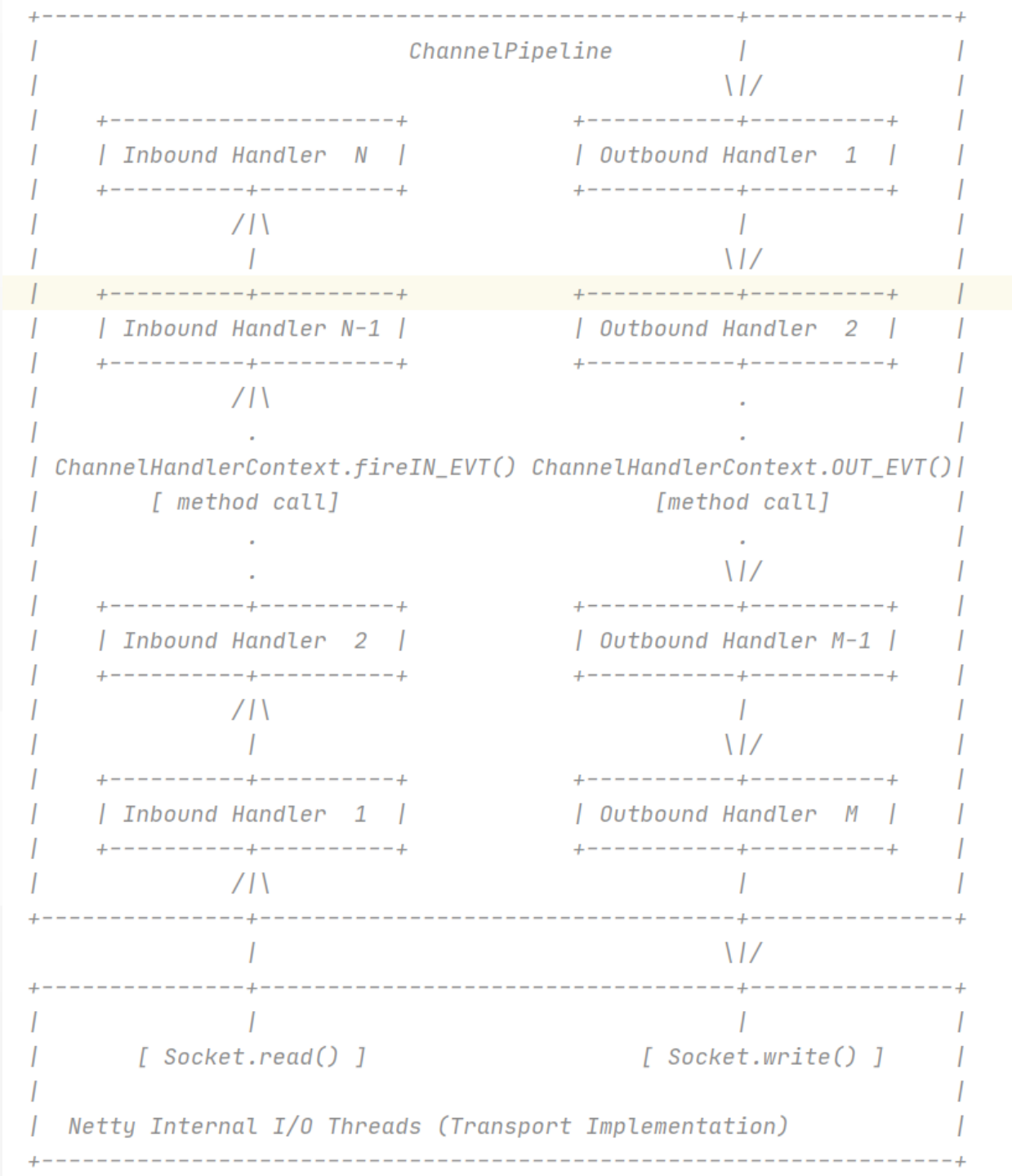

Netty的生态
Netty核心组件的使用和说明
拆包和粘包
Netty多协议开发应用
内存管理
序列化和反序列化
其他设计

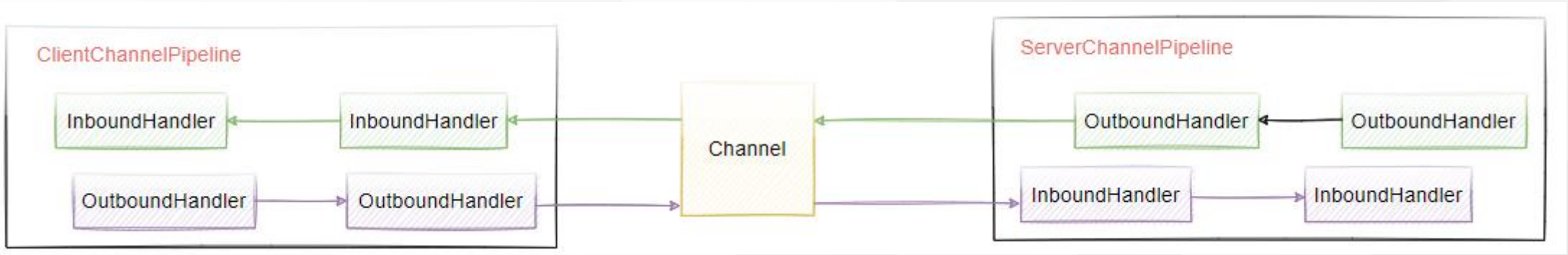

ServerBootStrap.bind(port)
NioServerSocketChannel
register
Client
Client
Client
CONNECT
bossNioEventGroup
Accept
Boss
NioEventLoop
NioSocketChannel
Worker
EventLoopGroup
EventLoop
Select事件循环
Read、Write I/O事件就绪
EventLoop
Select事件循环
EventLoop
ChannelPipeline
入站：ChannelInBoundHandlerAdapter
ChannelHandler
ChannelHandler
ChannelHandler
ChannelHandler
出站：ChannelOutBoundHandlerAdapter
网络通信层
事件调度器
服务编排层

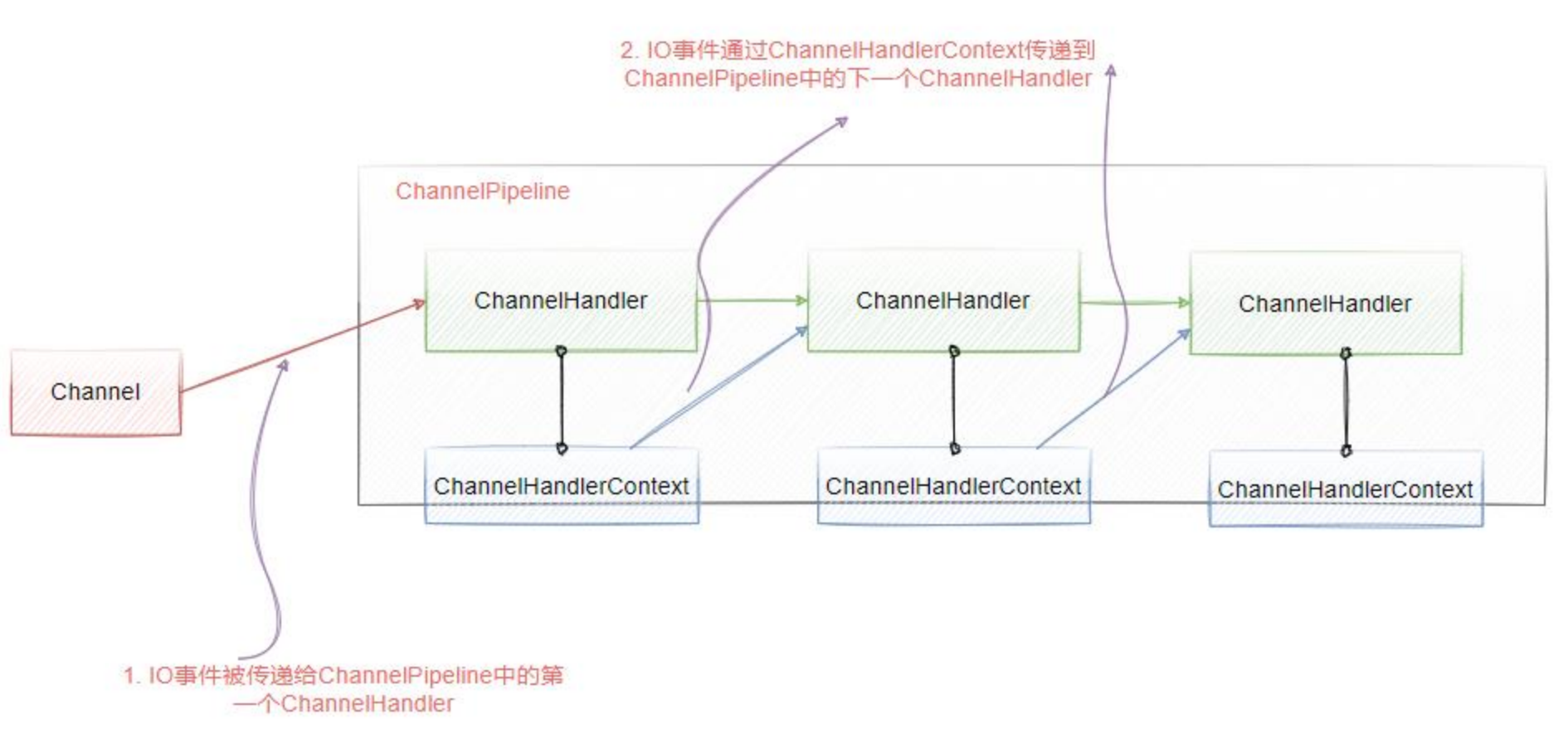

Channel
SocketChannel
AbstractChannel
KQueueSocketChannel
NioSocketChannel
EpollSocketChannel
AbstractNioMessageChannel
ServerChannel
EpollServerSocketChannel
NioServerSocketChannel
KQueueServerSocketChannel

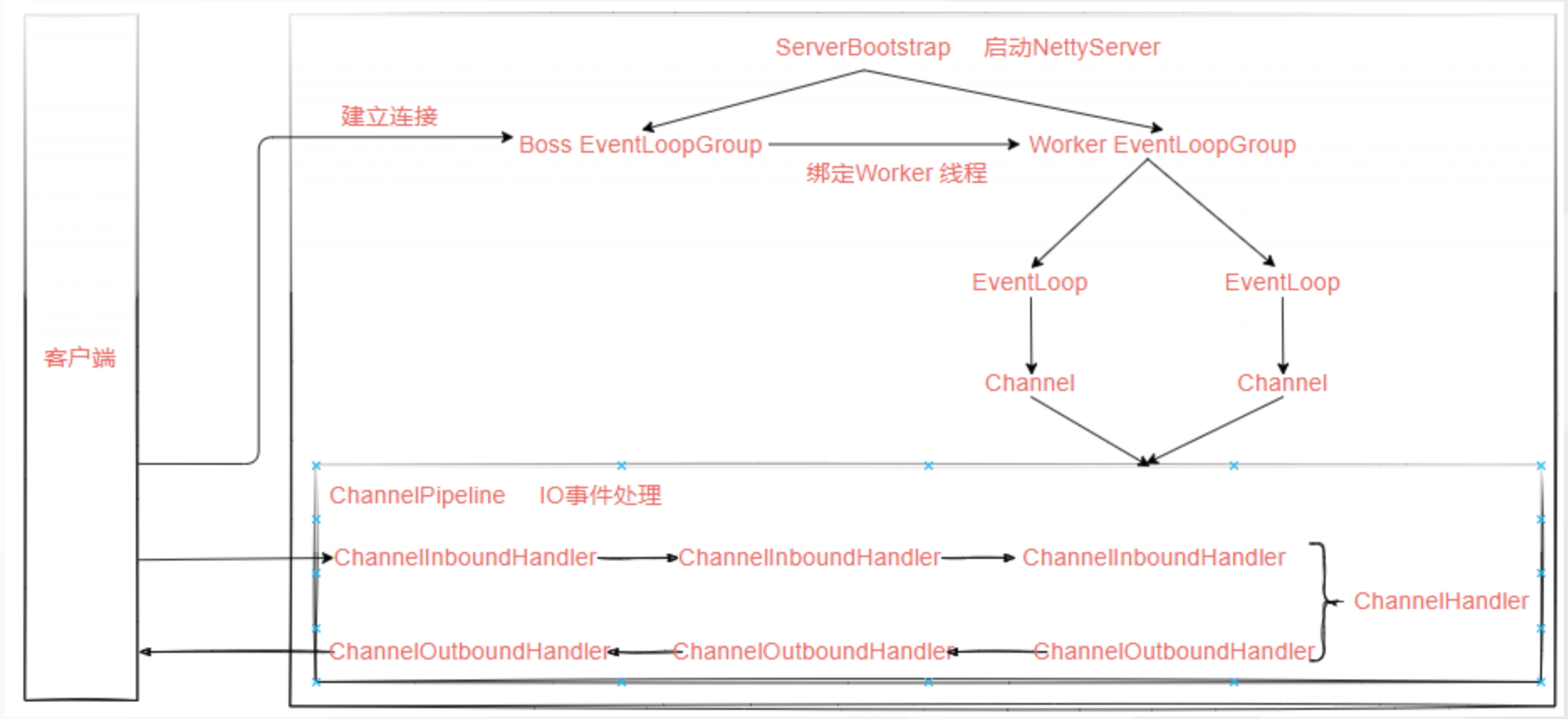

EventLoopGroup
EventLoop
EventLoop
EventLoop
EventLoop
EventLoop
EventLoop
Channel
Channel
Channel
建立新连接
绑定EventLoop
EventLoop处理网络
连接生命周期中的所
有I/O事件

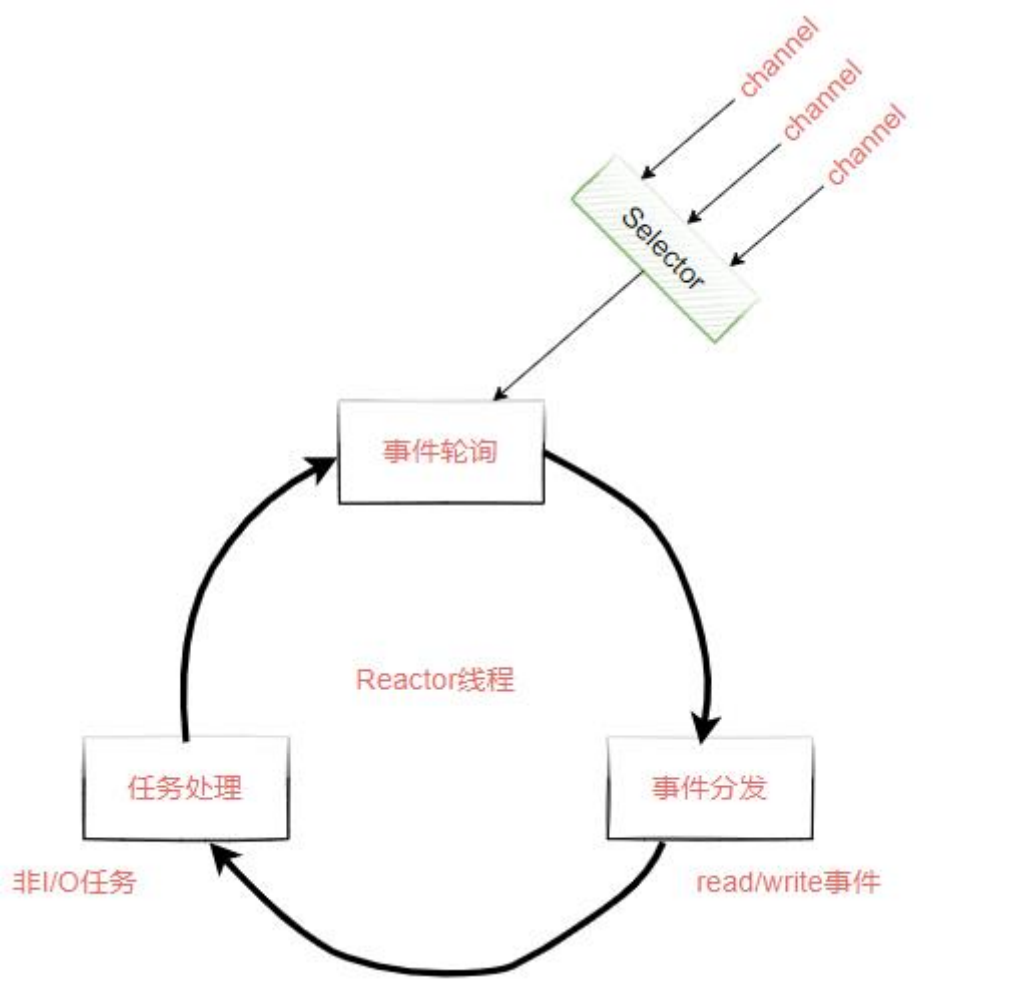

Executor
ExecutorService
ScheduledExecutorService
Iterable
EventExecutorGroup
AbstractEventExecutorGroup
EventLoopGroup
MultithreadEventExecutorGroup
EventLoop
MultithreadEventLoopGroup
NioEventLoop
EpollEventLoop
NioEventLoopGroup
KQueueEventLoop

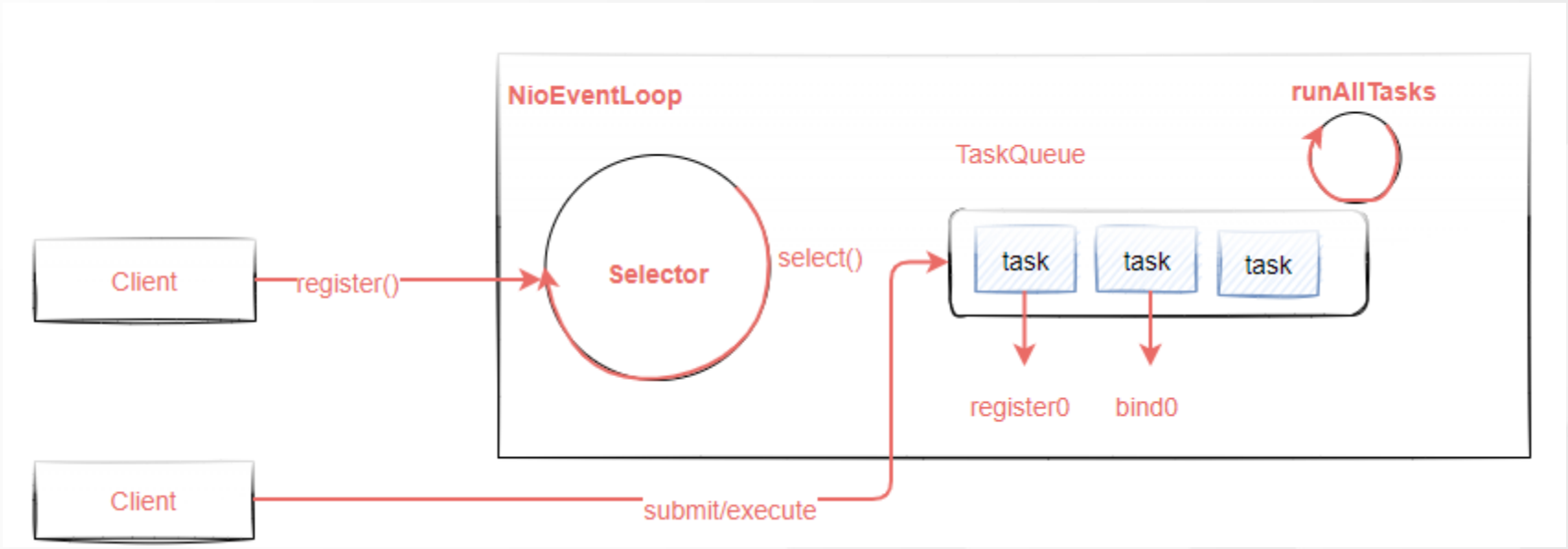

ChannelPipeline
Inbound Handler N
Outbound Handler 1
Inbound Handler N-1
Outbound Handler 2
ChannelHandlerContext.fireIN_EVT()
[method call]
ChannelHandlerContext.OUT_EVT()
[method call]
Inbound Handler 2
Outbound Handler M-1
Inbound Handler 1
Outbound Handler M
[Socket.read()]
[Socket.write()]
Netty Internal I/O Threads (Transport Implementation)

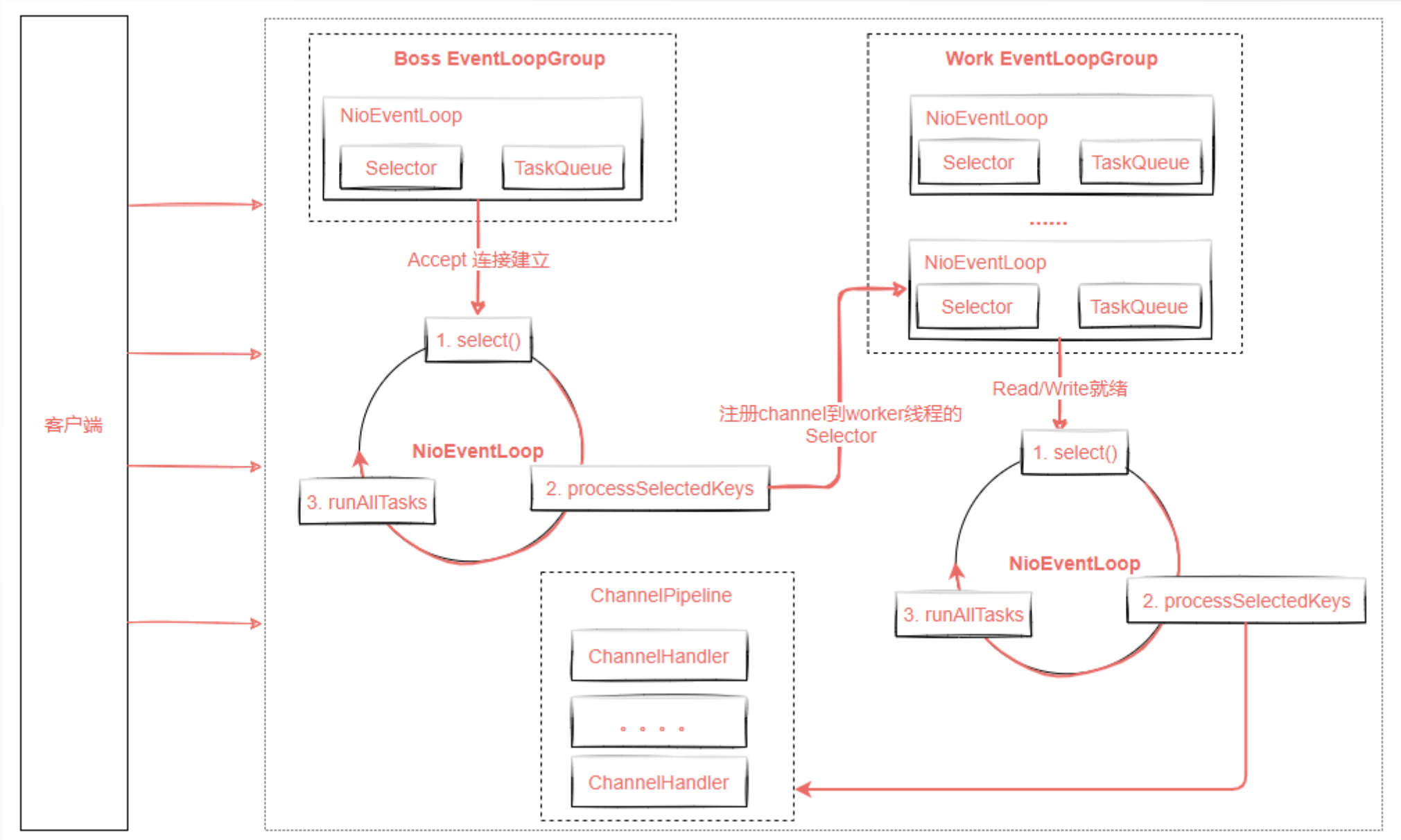

ClientChannelPipeline
InboundHandler
InboundHandler
OutboundHandler
OutboundHandler
Channel
ServerChannelPipeline
OutboundHandler
OutboundHandler
InboundHandler
InboundHandler

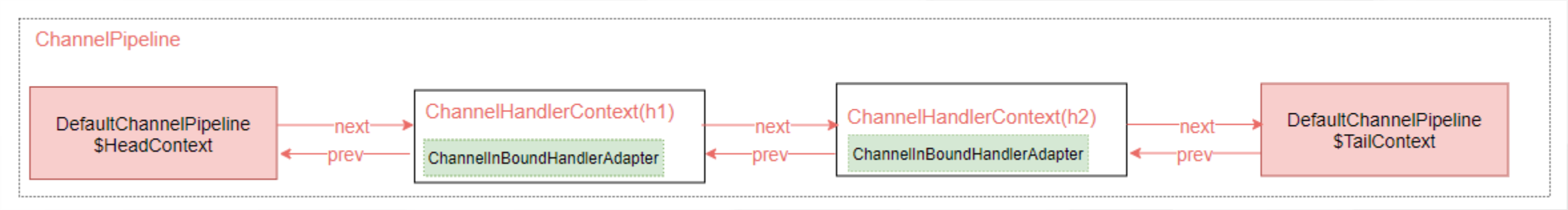

2. IO事件通过ChannelHandlerContext传递到
ChannelPipeline中的下一个ChannelHandler
ChannelPipeline
ChannelHandler
ChannelHandler
ChannelHandler
Channel
ChannelHandlerContext
ChannelHandlerContext
ChannelHandlerContext
1. IO事件被传递给ChannelPipeline中的第
一个ChannelHandler

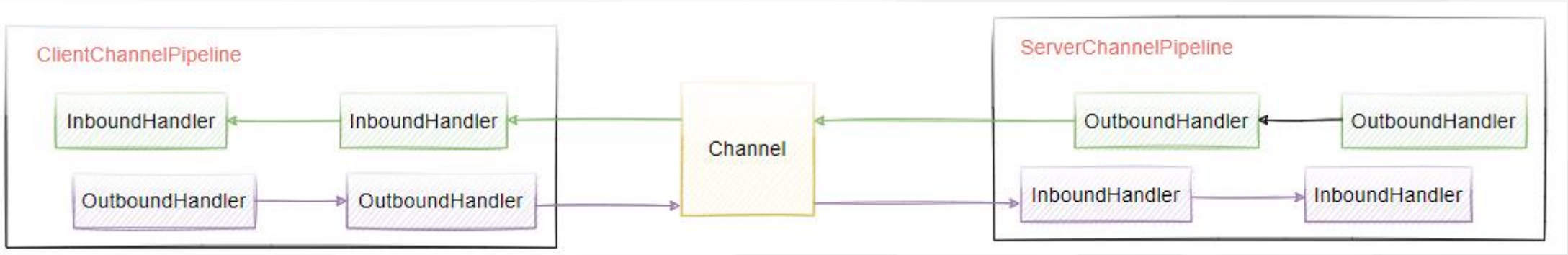

ServerBootstrap
启动NettyServer
建立连接
Boss EventLoopGroup
Worker EventLoopGroup
绑定Worker 线程
EventLoop
EventLoop
客户端
Channel
Channel
ChannelPipeline
IO事件处理
ChannelInboundHandler
ChannelInboundHandler
ChannelInboundHandler
ChannelHandler
ChannelOutboundHandler
ChannelOutboundHandler
ChannelOutboundHandler

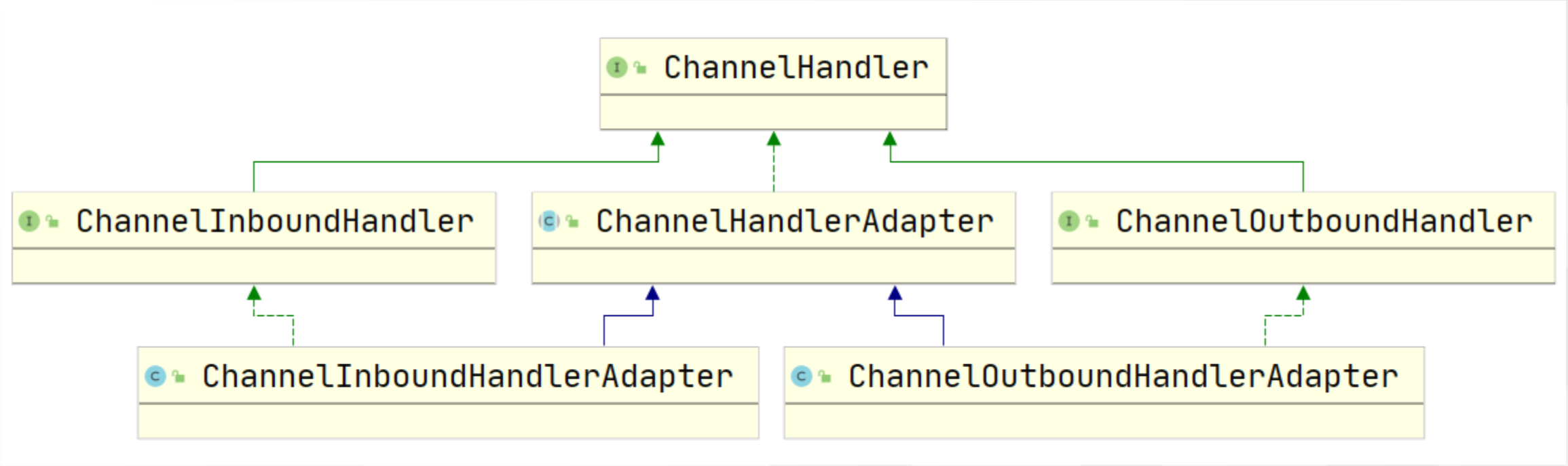

channel
channel
channel
Selector
事件轮询
Reactor线程
任务处理
事件分发
非I/O任务
read/write事件


NioEventLoop
runAllTasks
TaskQueue
Client
register()
Selector
select()
task
task
task
register0
bind0
Client
submit/execute


客户端
Boss EventLoopGroup
NioEventLoop
Selector
TaskQueue
Accept 连接建立
1. select()
NioEventLoop
3. runAllTasks
2. processSelectedKeys
注册channel到worker线程的
Selector
Work EventLoopGroup
NioEventLoop
Selector
TaskQueue
.....
NioEventLoop
Selector
TaskQueue
Read/Write就绪
1. select()
NioEventLoop
3. runAllTasks
2. processSelectedKeys
ChannelPipeline
ChannelHandler
. . . .
ChannelHandler


ChannelPipeline
DefaultChannelPipeline
$HeadContext
next
prev
ChannelHandlerContext(h1)
ChannelInBoundHandlerAdapter
next
prev
ChannelHandlerContext(h2)
ChannelInBoundHandlerAdapter
next
prev
DefaultChannelPipeline
$TailContext


ClientChannelPipeline
InboundHandler
InboundHandler
OutboundHandler
OutboundHandler
Channel
ServerChannelPipeline
OutboundHandler
OutboundHandler
InboundHandler
InboundHandler


ChannelHandler
ChannelInboundHandler
ChannelHandlerAdapter
ChannelOutboundHandler
ChannelInboundHandlerAdapter
ChannelOutboundHandlerAdapter

咕泡教育
—gupaoedu—

你的建议，我们用心聆听

咕泡教育
—gupaoedu—

谢谢观赏

GUPAO EDU

Mic

我们的愿景

推动每一次人才升级

我们的使命

让每个人的职业生涯不留遗憾