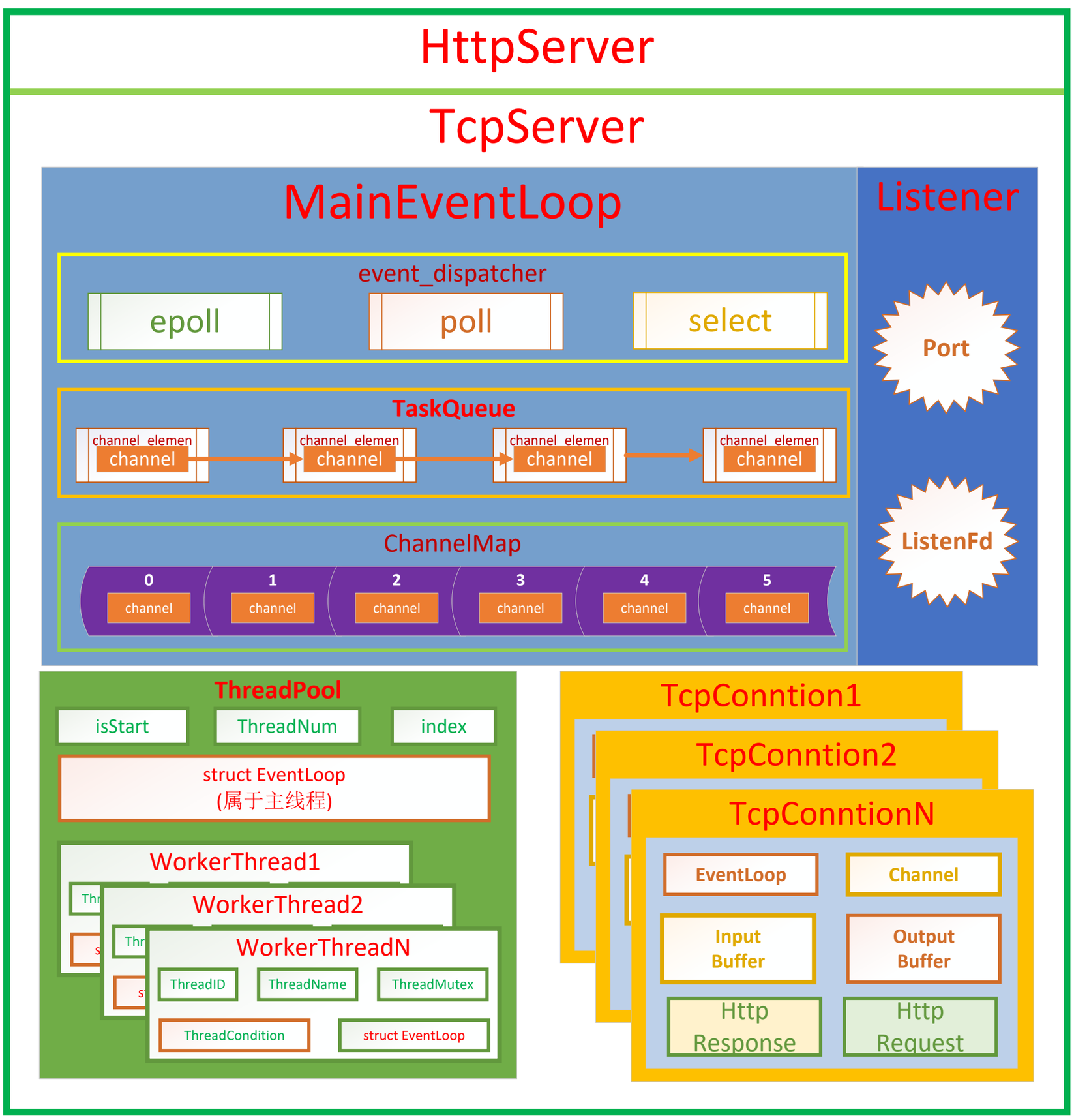
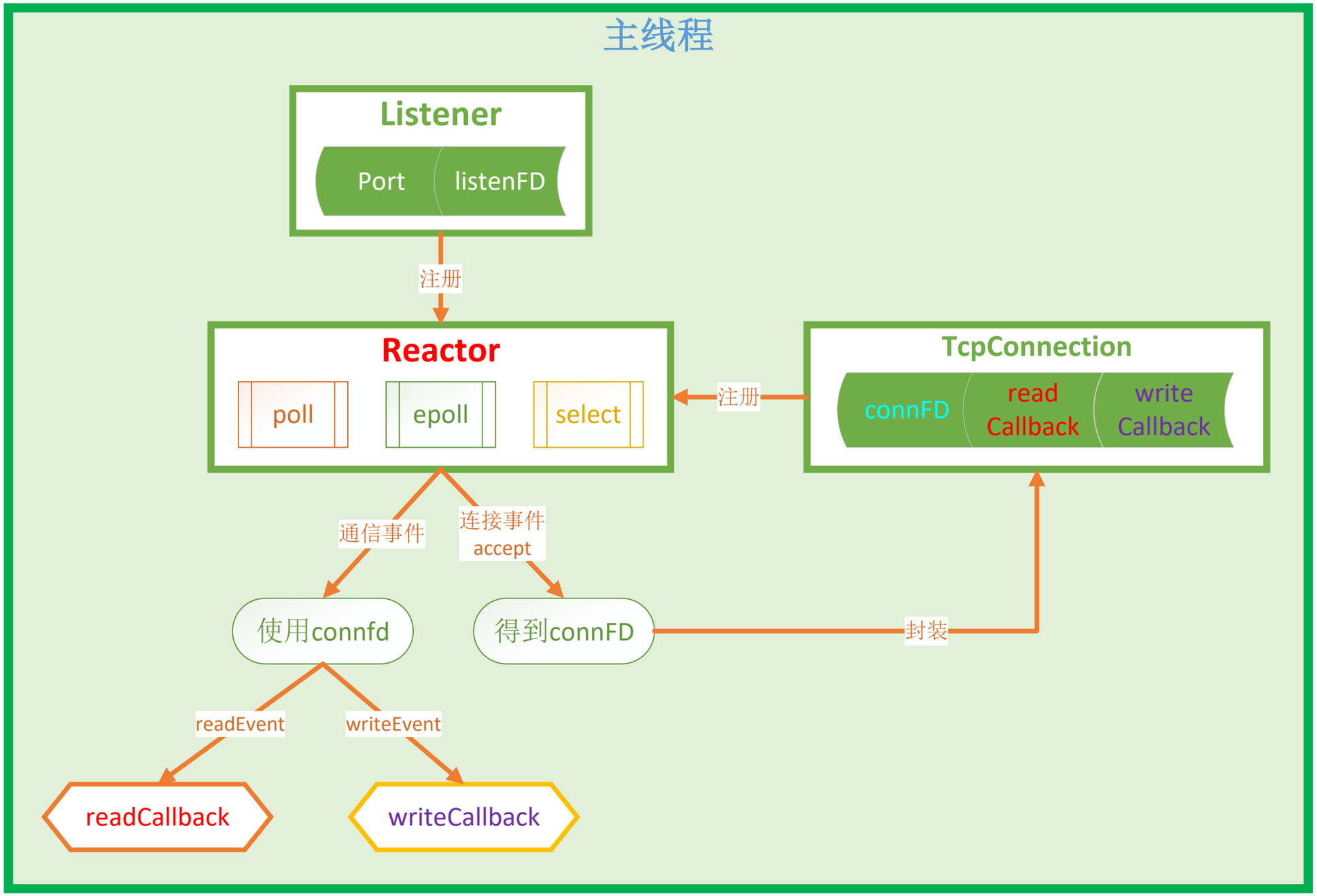
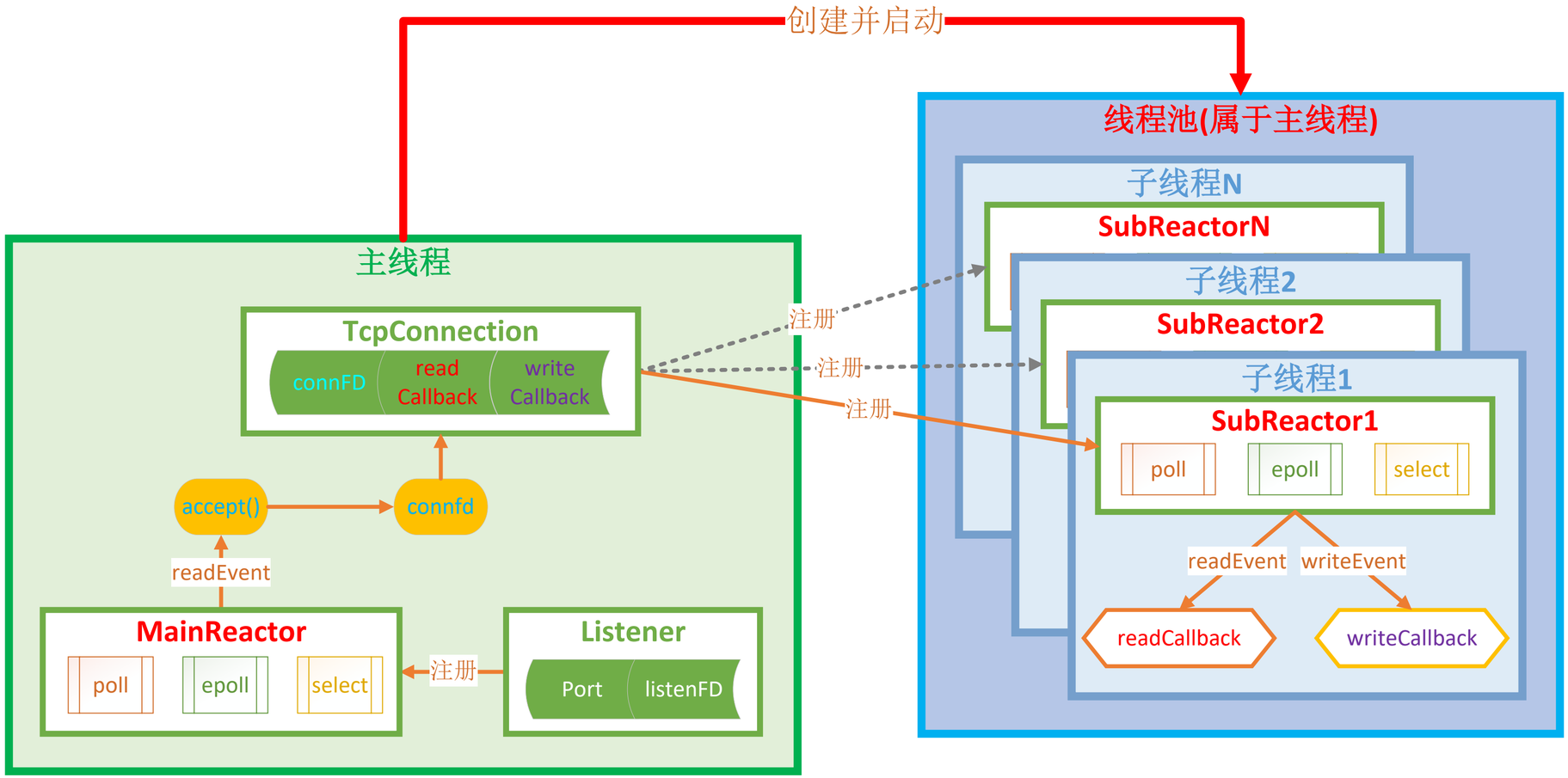
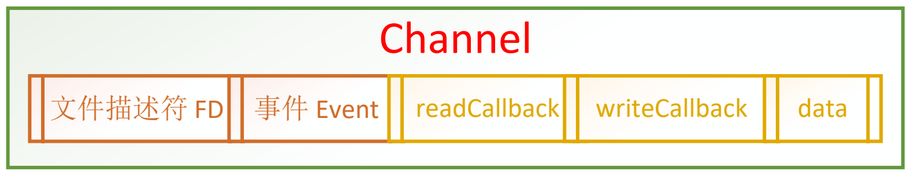
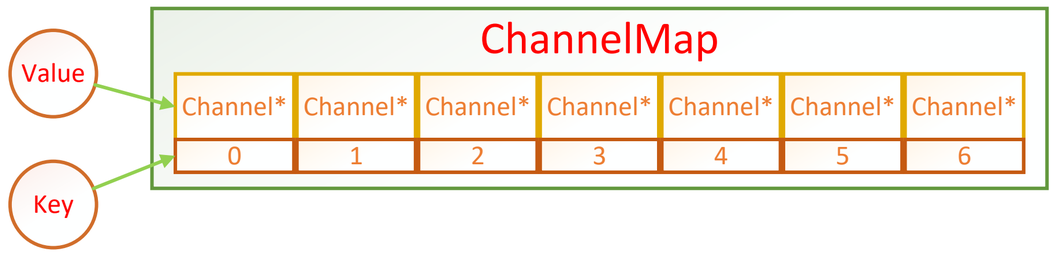
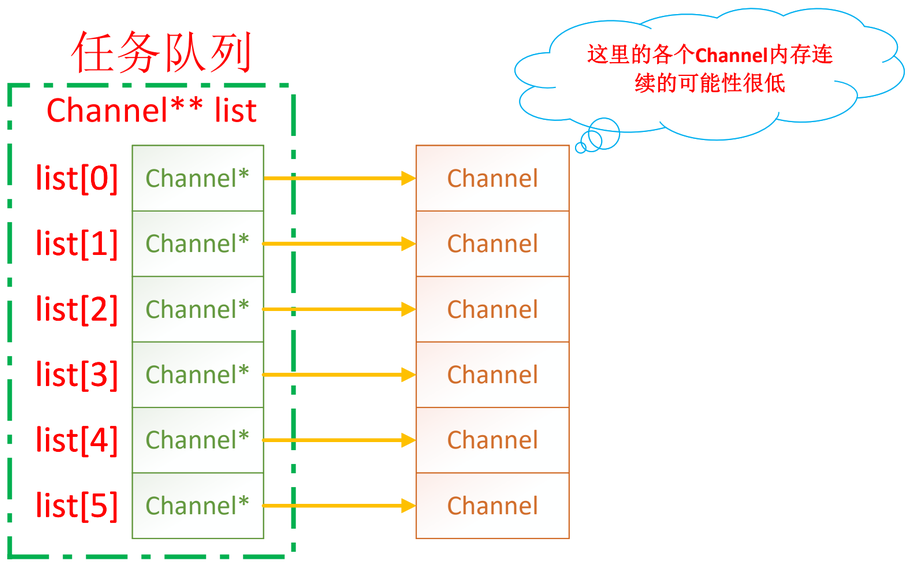

HttpServer
TcpServer
MainEventLoop
event_dispatcher
epoll
poll
select
TaskQueue
channel_elemen
channel
channel_elemen
channel
channel_elemen
channel
channel_elemen
channel
ChannelMap
0
channel
1
channel
2
channel
3
channel
4
channel
5
channel
Listener
Port
ListenFd
ThreadPool
isStart
ThreadNum
index
struct EventLoop
(属于主线程)
WorkerThread1
WorkerThread2
WorkerThreadN
ThreadID
ThreadName
ThreadMutex
ThreadCondition
struct EventLoop
TcpConntion1
TcpConntion2
TcpConntionN
EventLoop
Channel
Input
Buffer
Output
Buffer
Http
Response
Http
Request

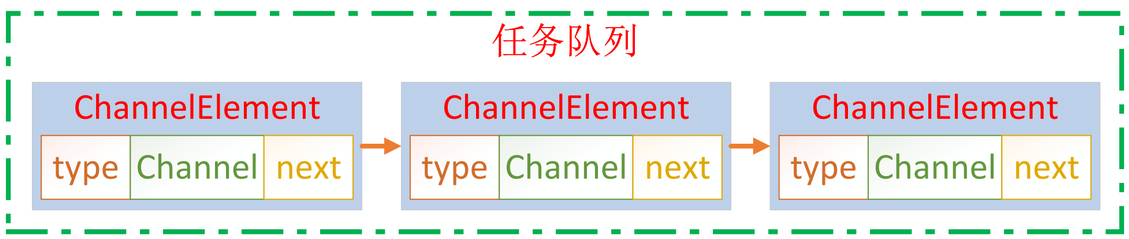

主线程
Listener
Port
listenFD
注册
Reactor
poll
epoll
select
注册
TcpConnection
connFD
read
Callback
write
Callback
通信事件
连接事件
accept
使用connfd
得到connFD
封装
readEvent
writeEvent
readCallback
writeCallback

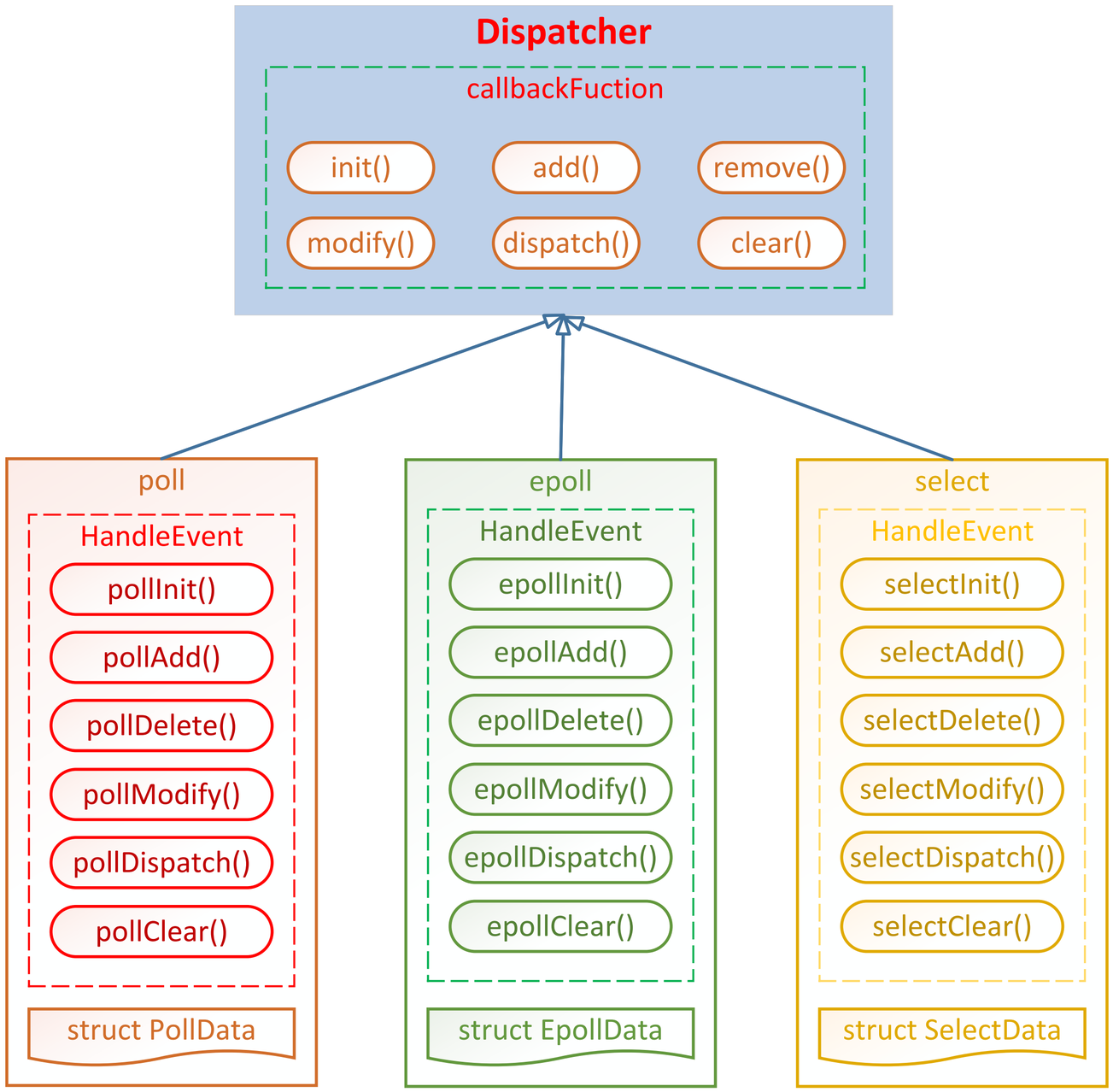

创建并启动
线程池(属于主线程)
子线程N
SubReactorN
子线程2
SubReactor2
子线程1
SubReactor1
poll
epoll
select
readEvent
writeEvent
readCallback
writeCallback
主线程
TcpConnection
connFD
read Callback
write Callback
accept()
connfd
readEvent
MainReactor
poll
epoll
select
注册
Listener
Port
listenFD
注册
注册
注册


Channel
文件描述符 FD
事件 Event
readCallback
writeCallback
data


ChannelMap
Value
Key
Channel*
Channel*
Channel*
Channel*
Channel*
Channel*
Channel*
0
1
2
3
4
5
6


任务队列
Channel** list
list[0]
list[1]
list[2]
list[3]
list[4]
list[5]
Channel*
Channel
这里的各个Channel内存连续的可能性很低


任务队列
ChannelElement
type
Channel
next
ChannelElement
type
Channel
next
ChannelElement
type
Channel
next


Dispatcher
callbackFuction
init()
add()
remove()
modify()
dispatch()
clear()
poll
HandleEvent
pollInit()
pollAdd()
pollDelete()
pollModify()
pollDispatch()
pollClear()
struct PollData
epoll
HandleEvent
epollInit()
epollAdd()
epollDelete()
epollModify()
epollDispatch()
epollClear()
struct EpollData
select
HandleEvent
selectInit()
selectAdd()
selectDelete()
selectModify()
selectDispatch()
selectClear()
struct SelectData

EventLoop

Dispatcher

epoll

poll

select

DispatcherData

TaskQueue

ChannelElement
channel*

ChannelElement
channel*

ChannelElement
channel*

ChannelMap

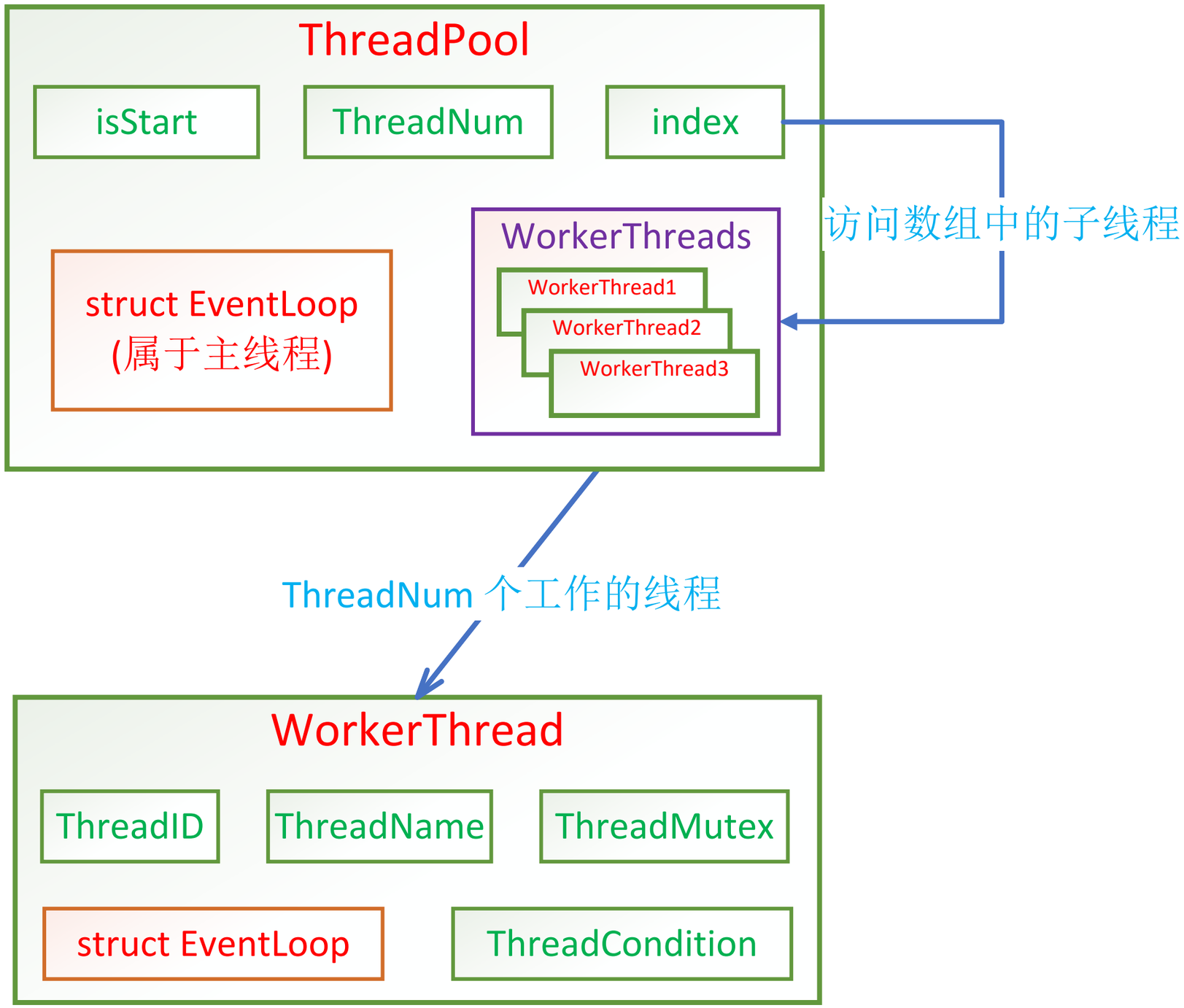
0	1	2	3	4
channel*	channel*	channel*	channel*	channel*

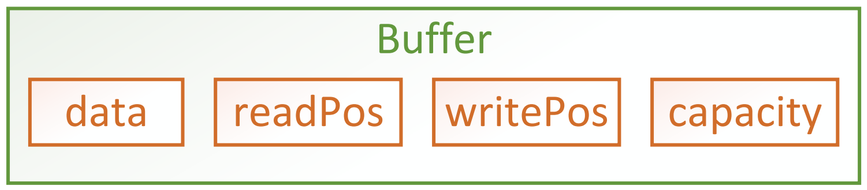
ThreadID

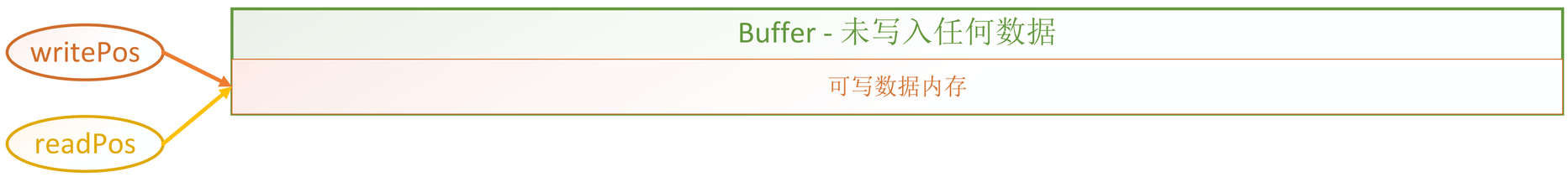
ThreadName

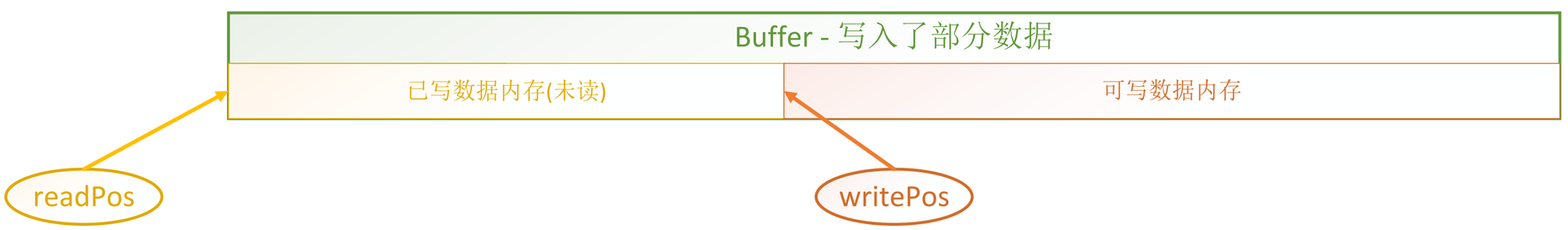
ThreadMutex

ThreadCondition

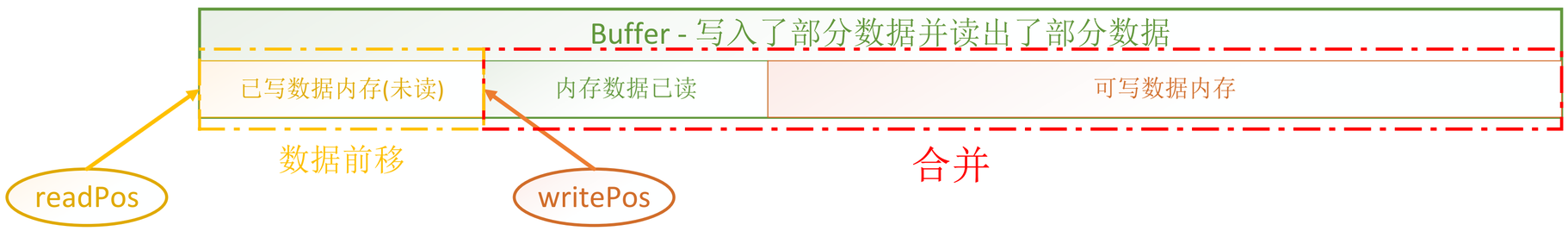

ThreadPool
isStart
ThreadNum
index
WorkerThreads
WorkerThread1
WorkerThread2
WorkerThread3
struct EventLoop
(属于主线程)
访问数组中的子线程
ThreadNum 个工作的线程
WorkerThread
ThreadID
ThreadName
ThreadMutex
struct EventLoop
ThreadCondition

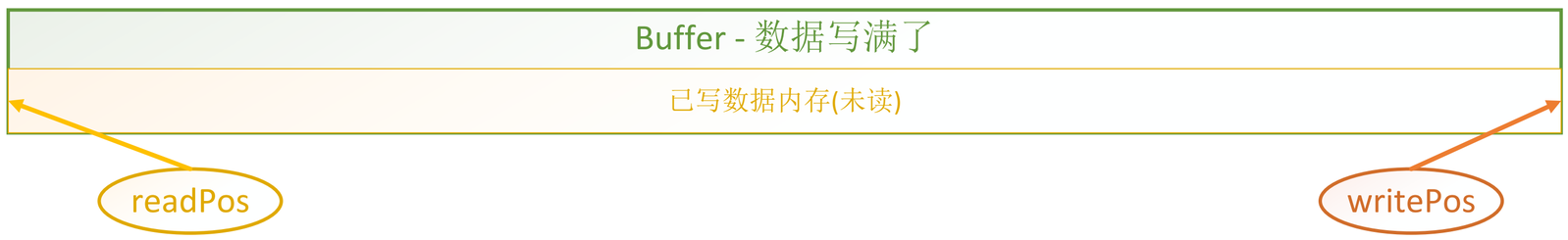

Buffer
data
readPos
writePos
capacity

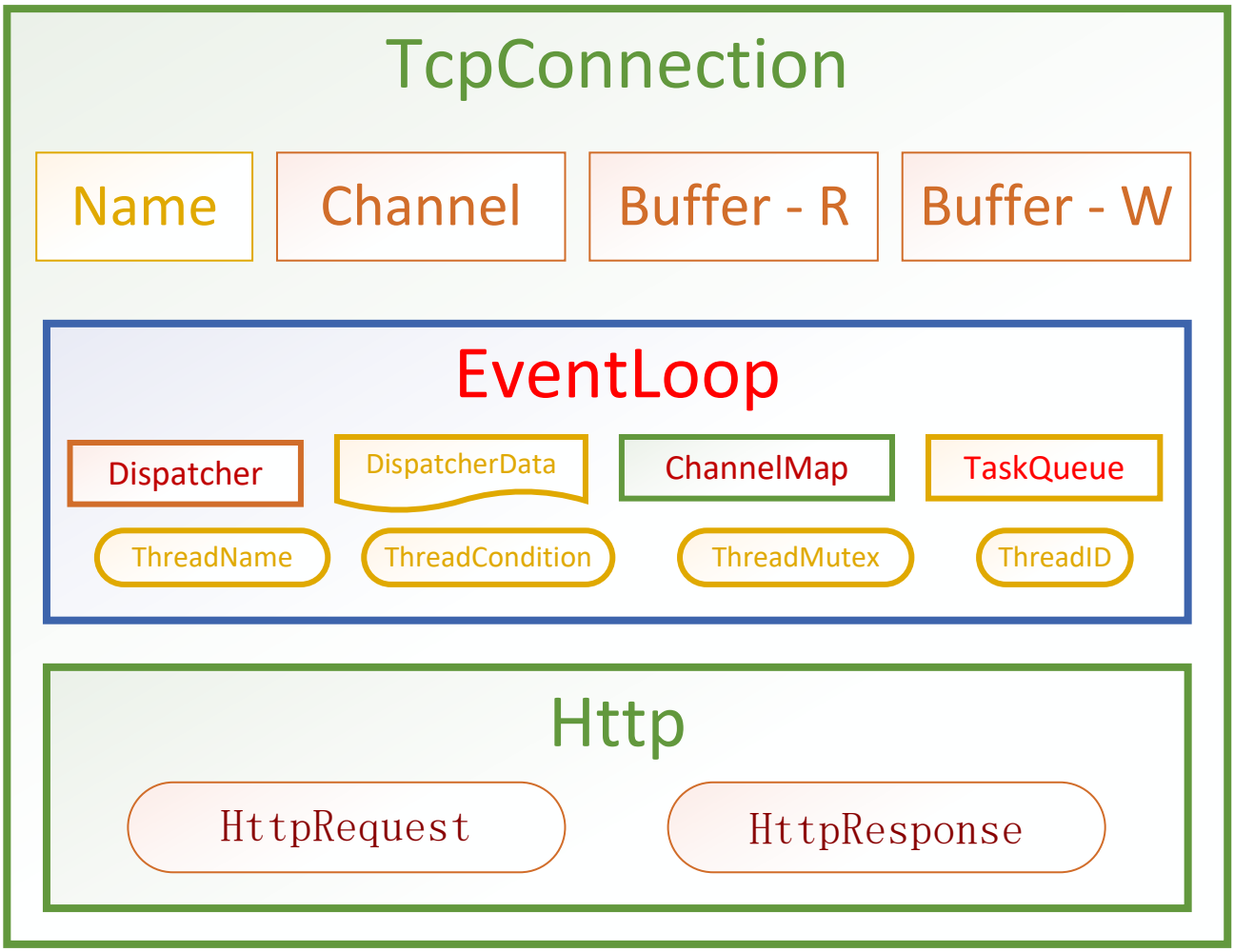

writePos
readPos
Buffer - 未写入任何数据
可写数据内存

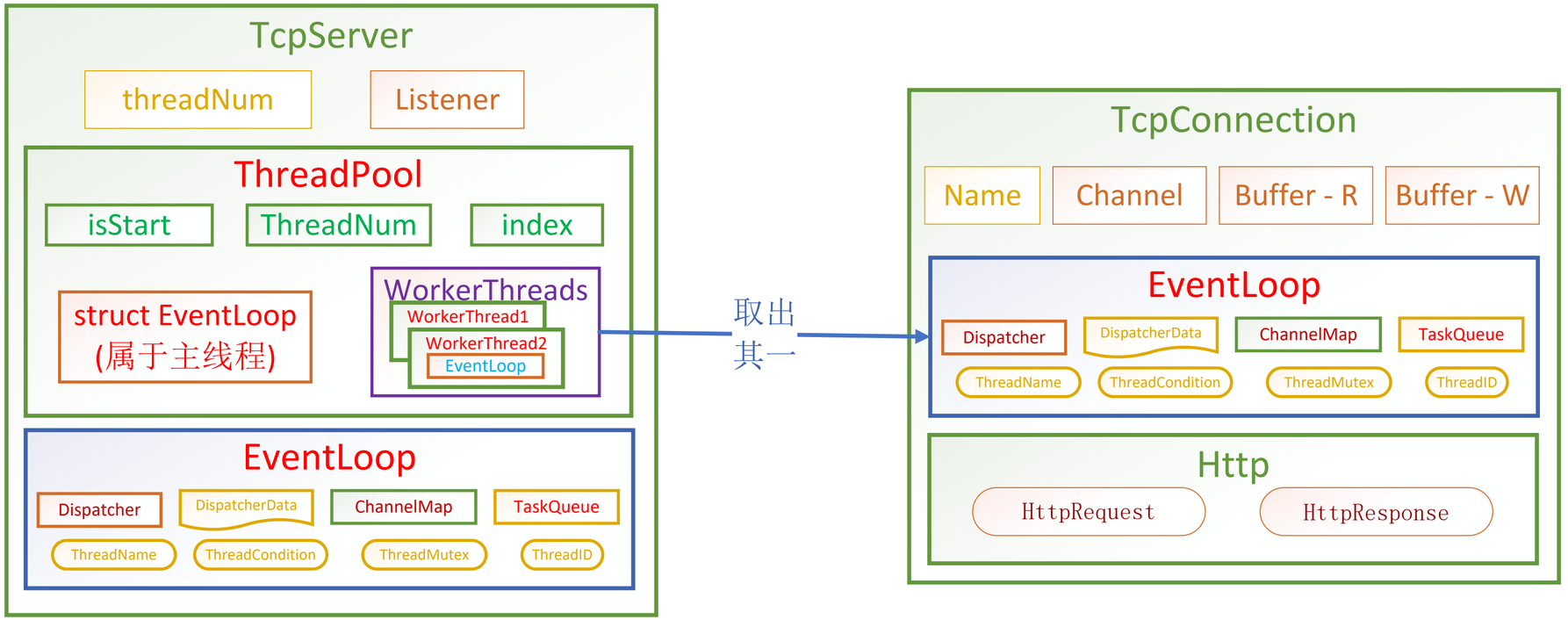

Buffer - 写入了部分数据
已写数据内存(未读)
可写数据内存
readPos
writePos

Buffer - 写入了部分数据并读出了部分数据
内存数据已读
已写数据内存(未读)
可写数据内存
readPos
writePos


Buffer - 写入了部分数据并读出了部分数据
已写数据内存(未读)
内存数据已读
可写数据内存
数据前移
合并
readPos
writePos


Buffer - 数据写满了
已写数据内存(未读)
readPos
writePos


TcpConnection
Name
Channel
Buffer - R
Buffer - W
EventLoop
Dispatcher
DispatcherData
ChannelMap
TaskQueue
ThreadName
ThreadCondition
ThreadMutex
ThreadID
Http
HttpRequest
HttpResponse


TcpServer
threadNum
Listener
ThreadPool
isStart
ThreadNum
index
struct EventLoop
(属于主线程)
WorkerThreads
WorkerThread1
WorkerThread2
EventLoop
EventLoop
Dispatcher
DispatcherData
ChannelMap
TaskQueue
ThreadName
ThreadCondition
ThreadMutex
ThreadID
取出
其一
TcpConnection
Name
Channel
Buffer - R
Buffer - W
EventLoop
Dispatcher
DispatcherData
ChannelMap
TaskQueue
ThreadName
ThreadCondition
ThreadMutex
ThreadID
Http
HttpRequest
HttpResponse

服务器初始化和启动

HttpServer | TcpServer | EventLoop | Listener | ThreadPool | Channel | Dispatcher

httpServerInit()
httpServerRun()
tcpServerInit()
tcpServerRun()
创建事件循环(主线程)
eventLoopInit()
绑定+设置监听
listenerInit()
创建线程池
threadPoolInit()
启动线程池
threadPoolRun()
1 - 封装监听的套接字
channelInit()
3
eventLoopRun()
(主线程)
1. 检测有无就绪的事件, 并处理
2 - 添加任务
到任务队列
eventLoopAddTask()
2. 继续处理任务队列中的任务
事件循环是否属
于当前线程
eventLoopDisposChannel()
eventLoopModify()
修改
任务类型
删除
eventLoopRemove()
添加
eventLoopAdd()
taskWakeup()
remove()
add()
modify()
唤醒，解除阻塞
注册
注册
程序退出
(主线程)
已结束
是否已退出
事件循环
未结束
dispatch()
已就绪的
事件类型
ReadEvent
WriteEvent
eventActive()
事件类型
WriteEvent
writeCallBack()
ReadEvent
readCallBack()
acceptConntion()
和客户端建立新连接

阶段