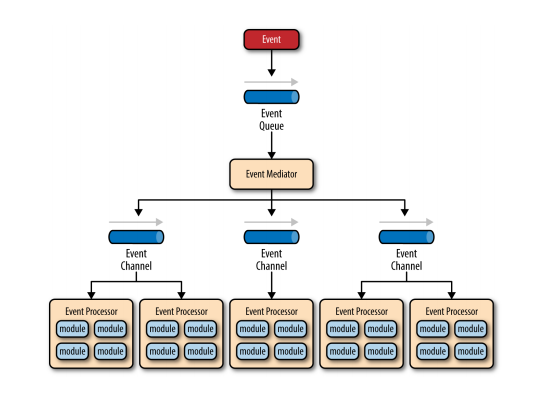
# 事件驱动

事件驱动有两种形式的结构：Mediator和Broken。

Mediator结构包含4个部分：事件队列，事件媒介，事件通道，事件处理器



# UE4动画系统

引擎中动画相关的资源：

1. 数据资源。由外部DCC工具制作的动画序列。
2. 由于有的时候动画序列是分散开来的，所以引擎中存在一种资源叫合成动画，将分散的动画序列合成起来。

## 动画混合

概念：同一个角色或骨骼上的两套动画之间平滑切换。

## 动画蒙太奇

蒙太奇：把各种效果混合在一起

# 框架分析

AppMain Tick函数每帧都会执行

# 技能系统

技能系统的难点：

1. 同一个角色有多个技能。
2. 场景中有多个角色在释放技能
3. 怪物也有技能。
4. 状态多变。攻击状态，空闲状态。受击状态。击飞。眩晕。
5. 技能单体，范围。
6. 带锁定，不带锁定。
7. 技能还可以被打断
8. 与服务器交互

什么是技能系统？

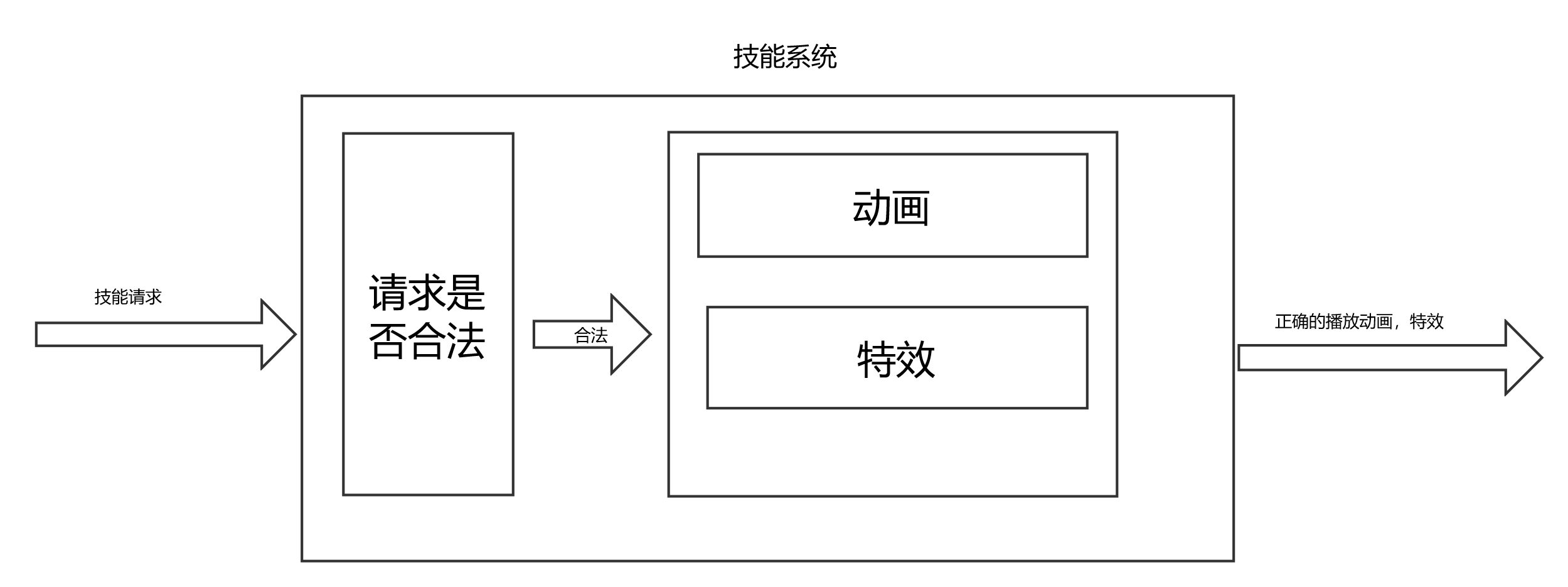
一个技能系统应该是一个对外封闭的黑箱。对于外界一个技能请求，做出正确的响应，正确的施放技能。



整个技能系统的难点在于 **正确** 二字

当技能数量变多，技能之间的交互规则变得复杂以后，技能系统会变的复杂很复杂

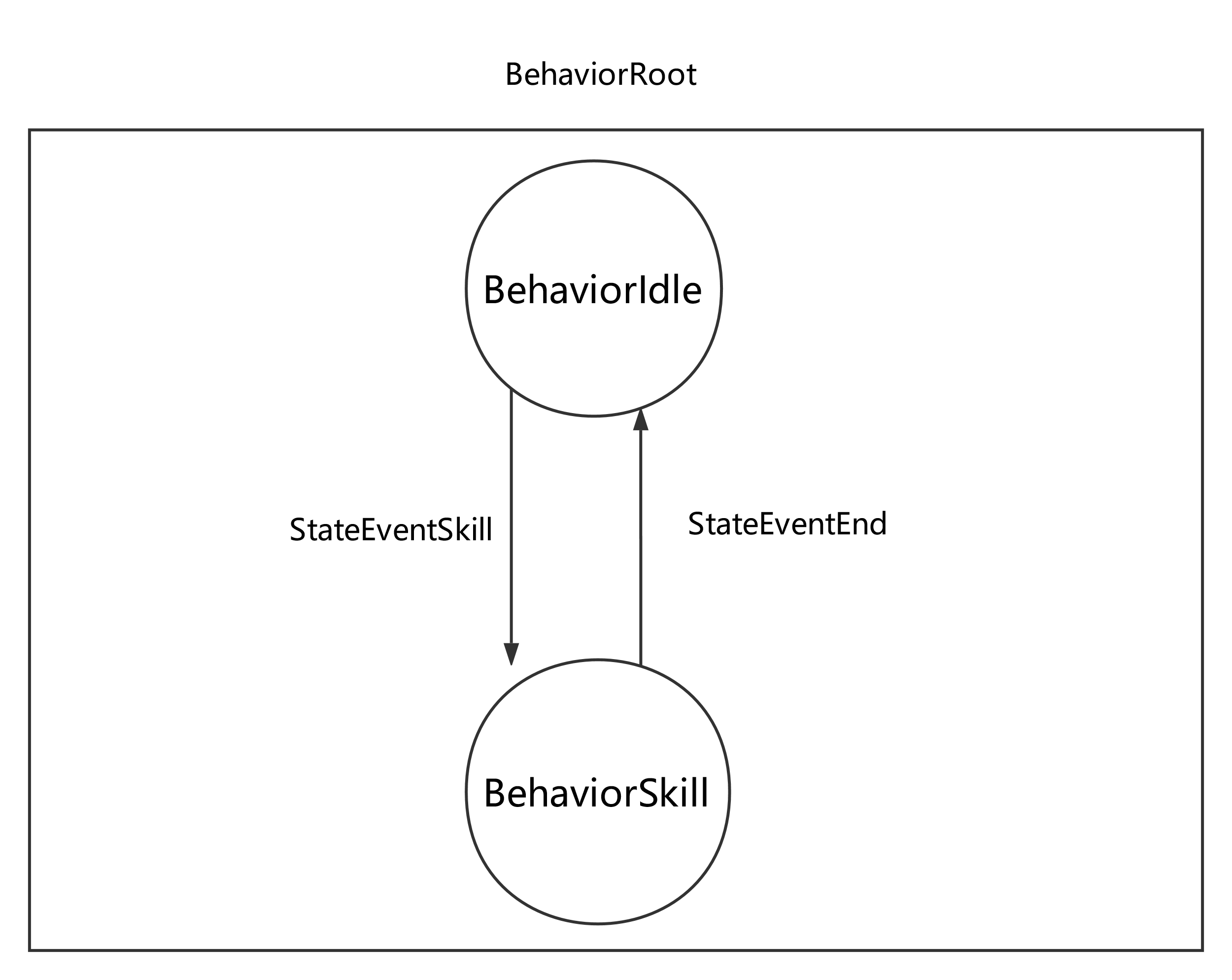
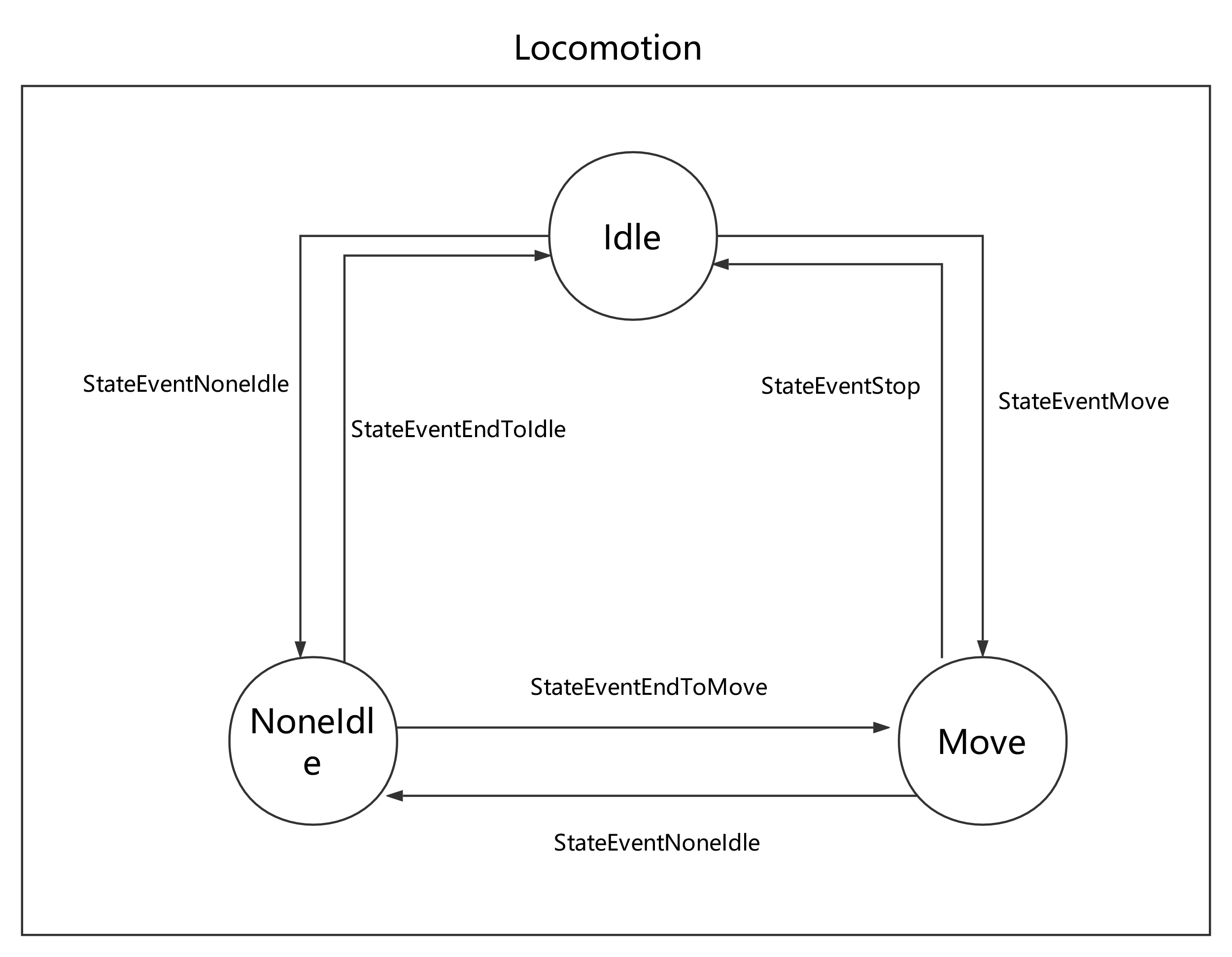
对于每一个技能请求，首先必须判断请求是否合法。合法以后再播放动画和特效。



所以一个技能系统主要分为两个阶段。第一阶段对请求的合法性进行考察。第二阶段播放正确的动画和特效。

如何播放正确的动画和特效呢？ 状态机。当角色进入某一状态时，播放对应状态的特效和动画。

目前系统中的状态机的实现是一个并行状态机。其中一个是运动状态机，Locomotion。另外一个是技能状态机BehaviorRoot



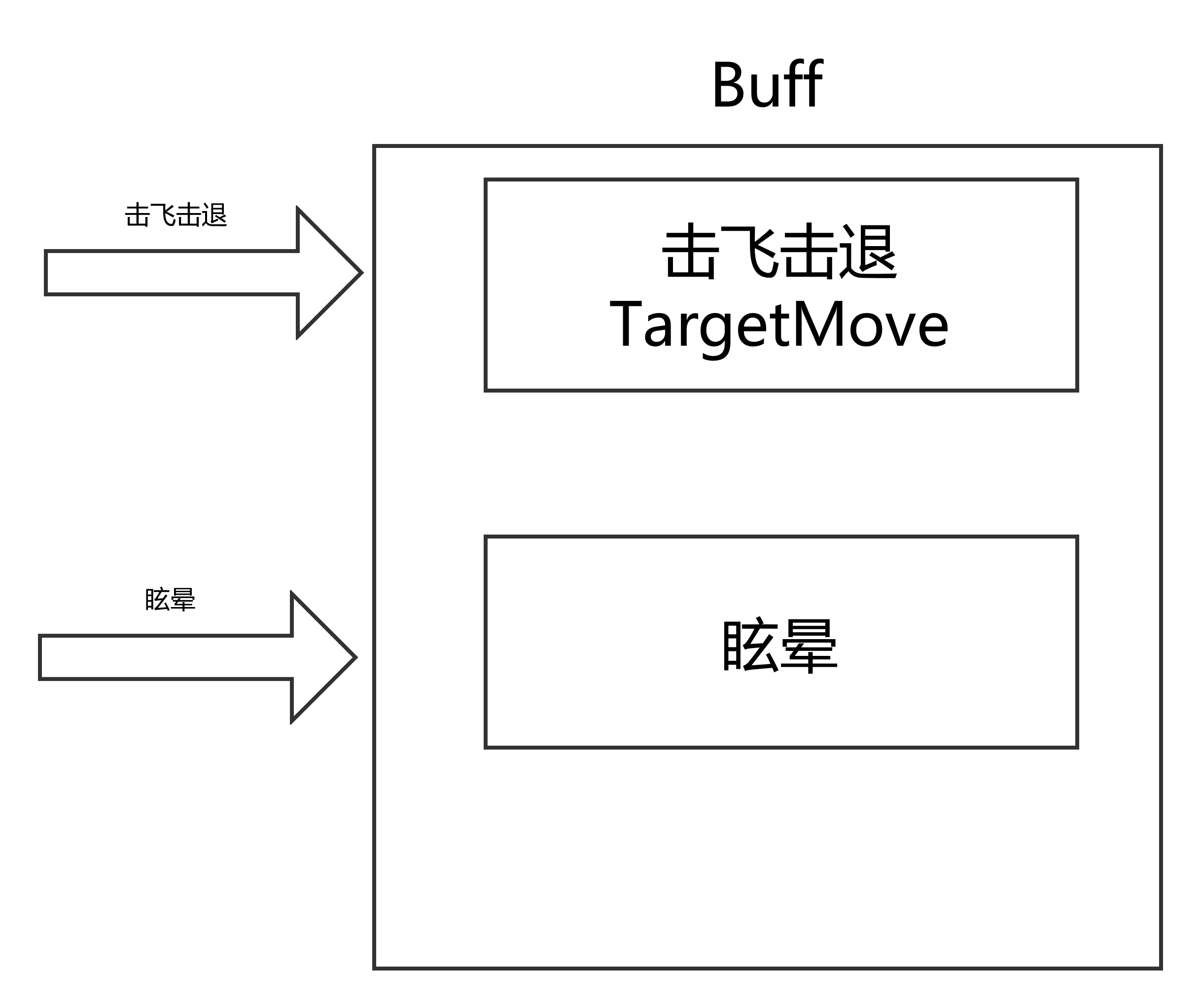
对于每一个状态会有一个Entry函数和一个Exit函数。

一个好的技能系统应该有尽可能少的入口。我们这个技能系统在这一点上设计的非常好。只有唯一的一个入口。当玩家发动技能时，我们把技能ID传进来。然后由技能系统来判断技能是否合法，以及施放。

之前的状态机是用来处理技能动作的。然而一个技能除了动作以外，还可以施加各种Buff。因此一个技能系统可以划分为如下结构。



一个Buff系统可以处理多种Buff请求，如击飞击倒眩晕等。



# UE4物理材质

# UE4基本材质概念

# UE4 物理碰撞