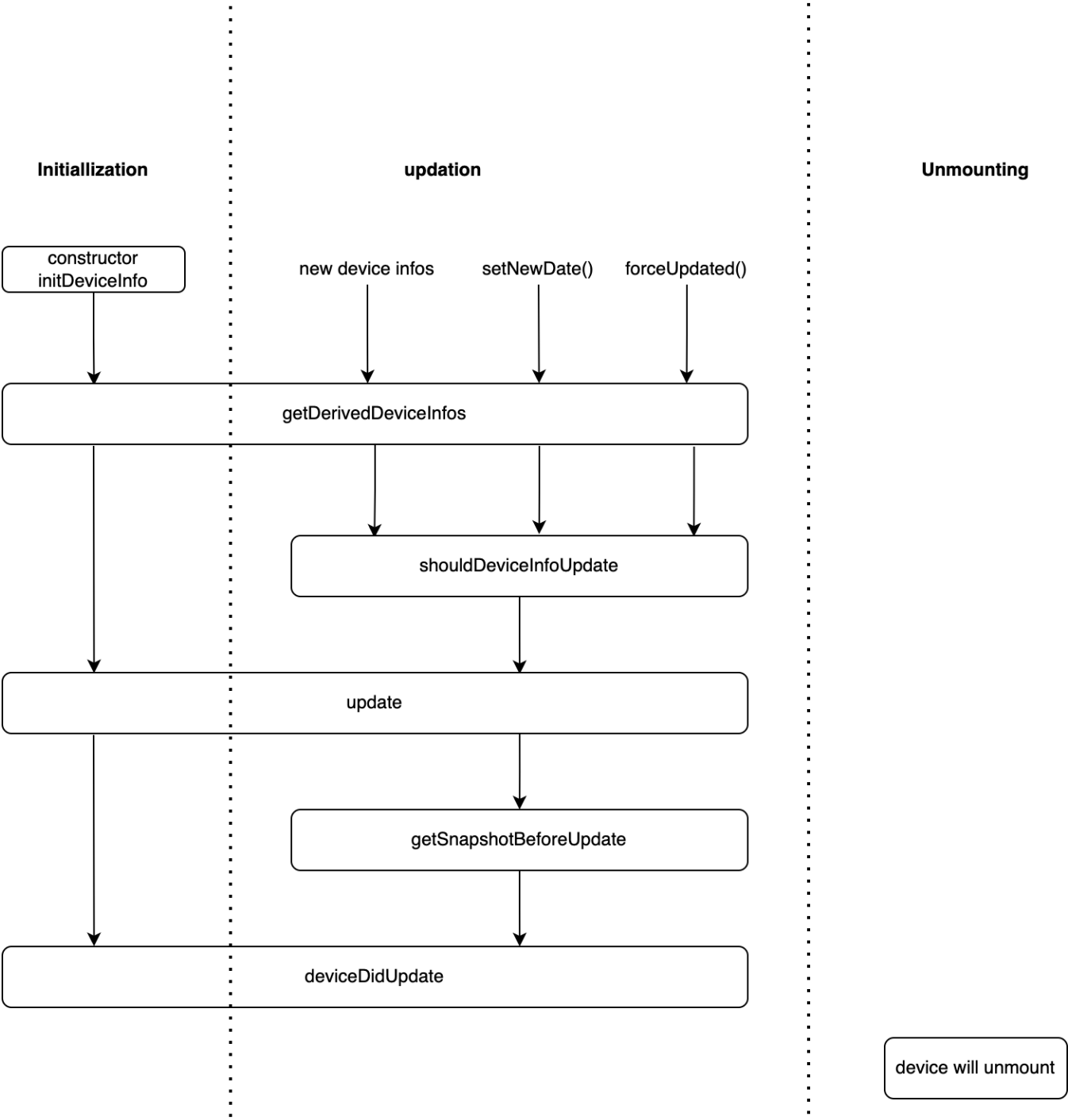


Tinker device状态 生命周期架构设计

tinker device 的生命周期前端架构图：



constructor()/initDeviceInfo(): 如果不初始化device或者不进行初始化动作则不需要实现initDeviceInfo。在收到device前，会调用initDeviceInfo。

static getDerivedDeviceInfos(nextTinkerDeviceInfo,prevCNSTDeviceInfo): 用于 修改新的device的上报的信息，在更新（update）device信息前，对device上报的信息做修正（或忽略）可以调用该方法。该方法为静态方法。提供两个参数nextTinkerDeviceInfo和prevCNSTDeviceInfo。

shouldDeviceInfoUpdate(): 此方法 可以用于比较device新旧数据，如果返回false 则不会执行update方法。此方法仅作为性能优化的方式而存在。不要企图依靠此方法来“阻止”device 的update，因为这可能会产生 bug。

update():该函数是device生命周期中唯一必须实现的方法，当update方法被调用时，会检查tinker上展示的device的SN与物理上的SN是否相匹配，并存储新的device信息。

getSnapshotBeforeUpdate():在更新tinker中device状态前调用，比如 想在tinker内存中记录一下之前的状态 以便与新状态做对比。

deviceDidUpdate(): 该函数会在与CNST/SH同步数据后立即调用。如果需要通过网络请求进行其他的操作，此处是实例化请求的好地方。这个方法也是比较适合添加订阅的地方。如果添加了订阅。请不要忘记在deviceWillUnmount()里取消订阅。

deviceWillUnmount(): 会在device卸载及销毁之前直接调用。在此方法中执行必要的清理操作，例如，清除 timer，取消网络请求或清除在deviceDidUpdate()中创建的订阅等。