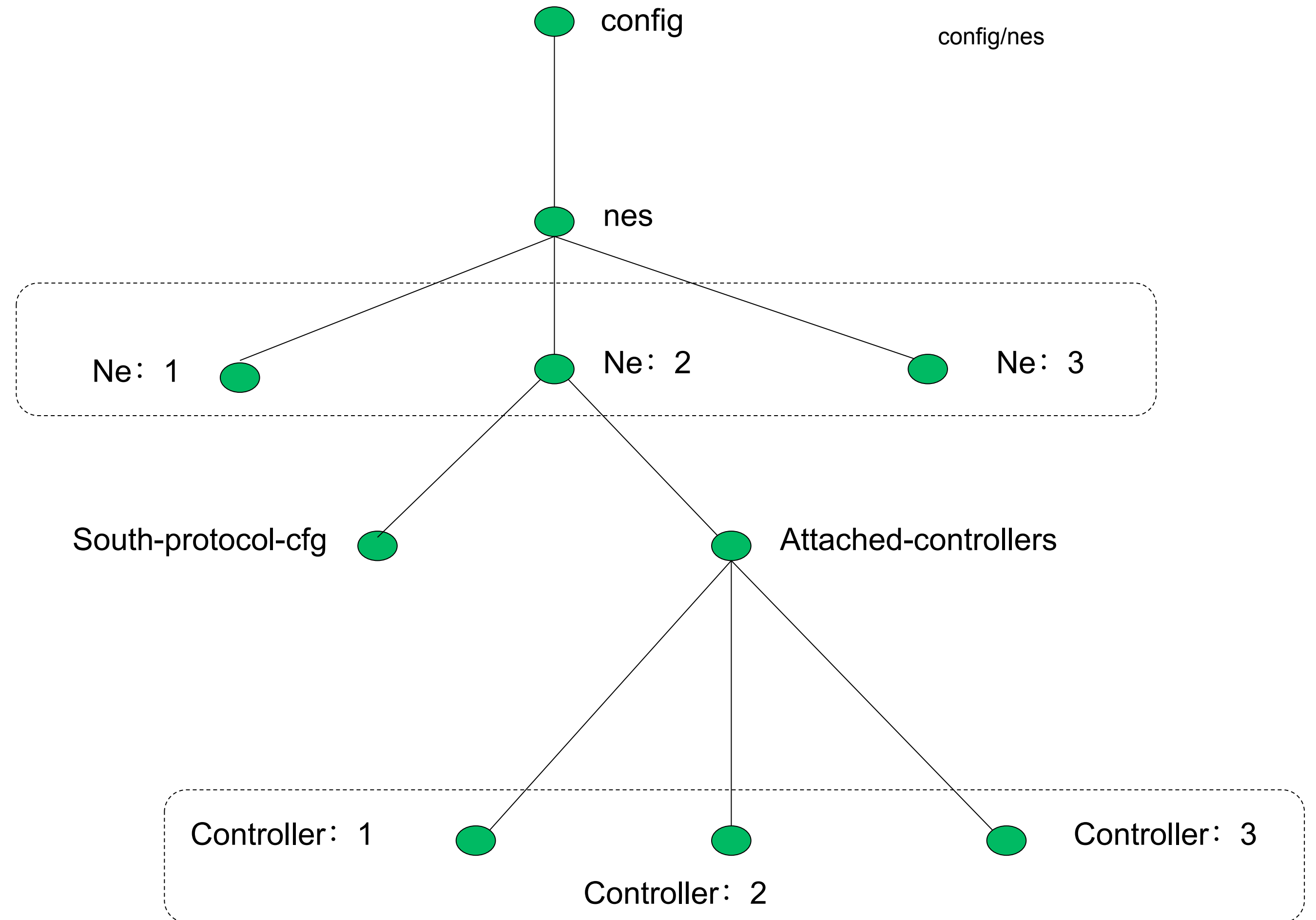


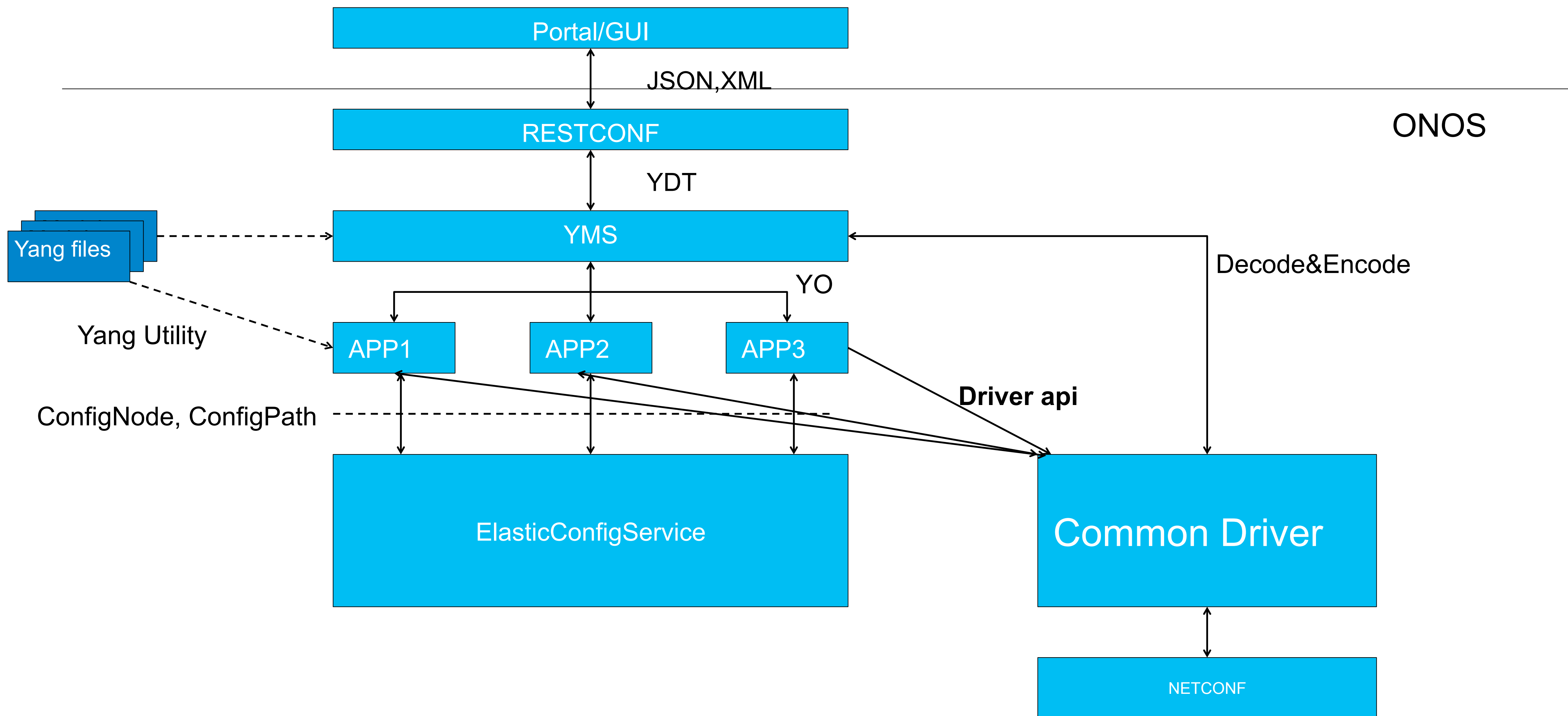
Dynamic Configuration System Diagram and Common Driver Idea

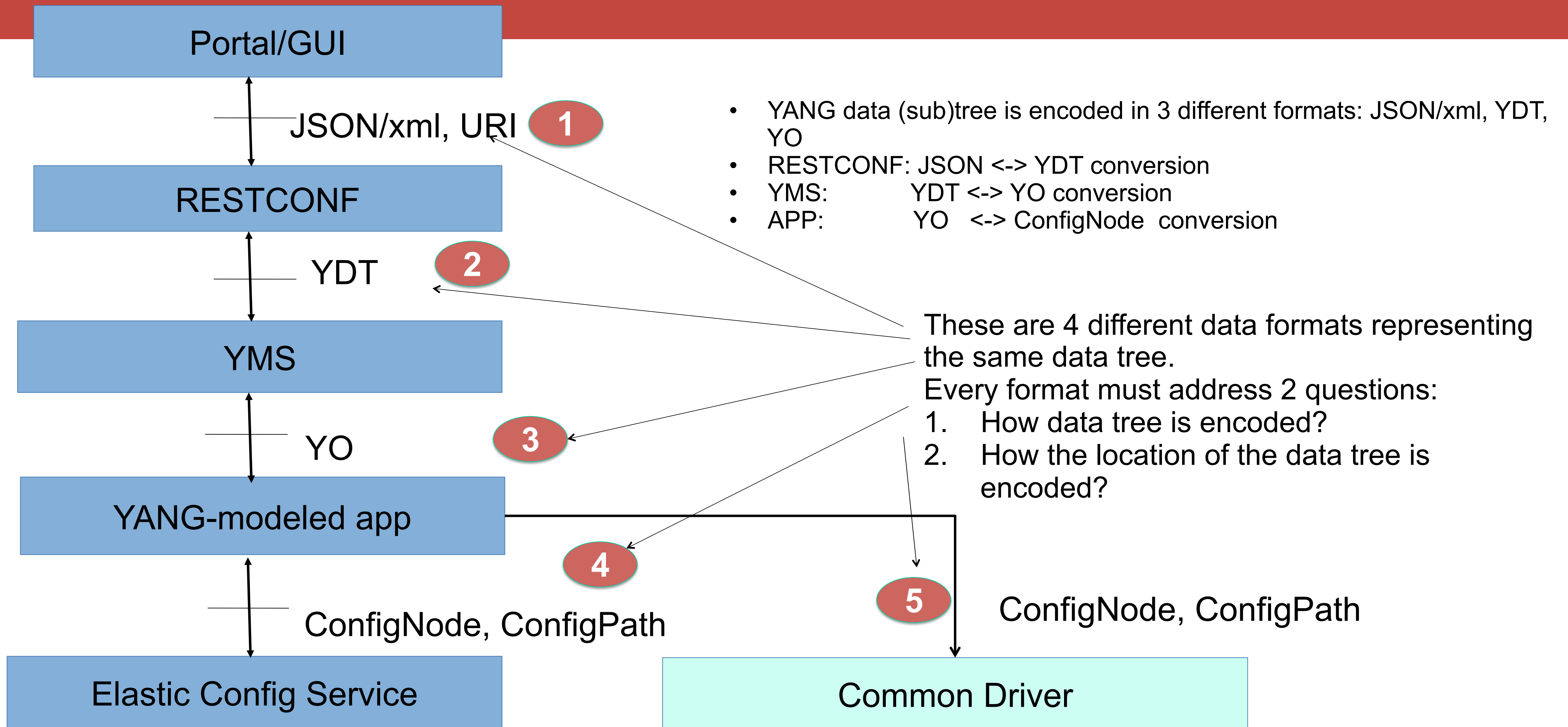
Patrick Liu/Henry Yu
Nov. 2016

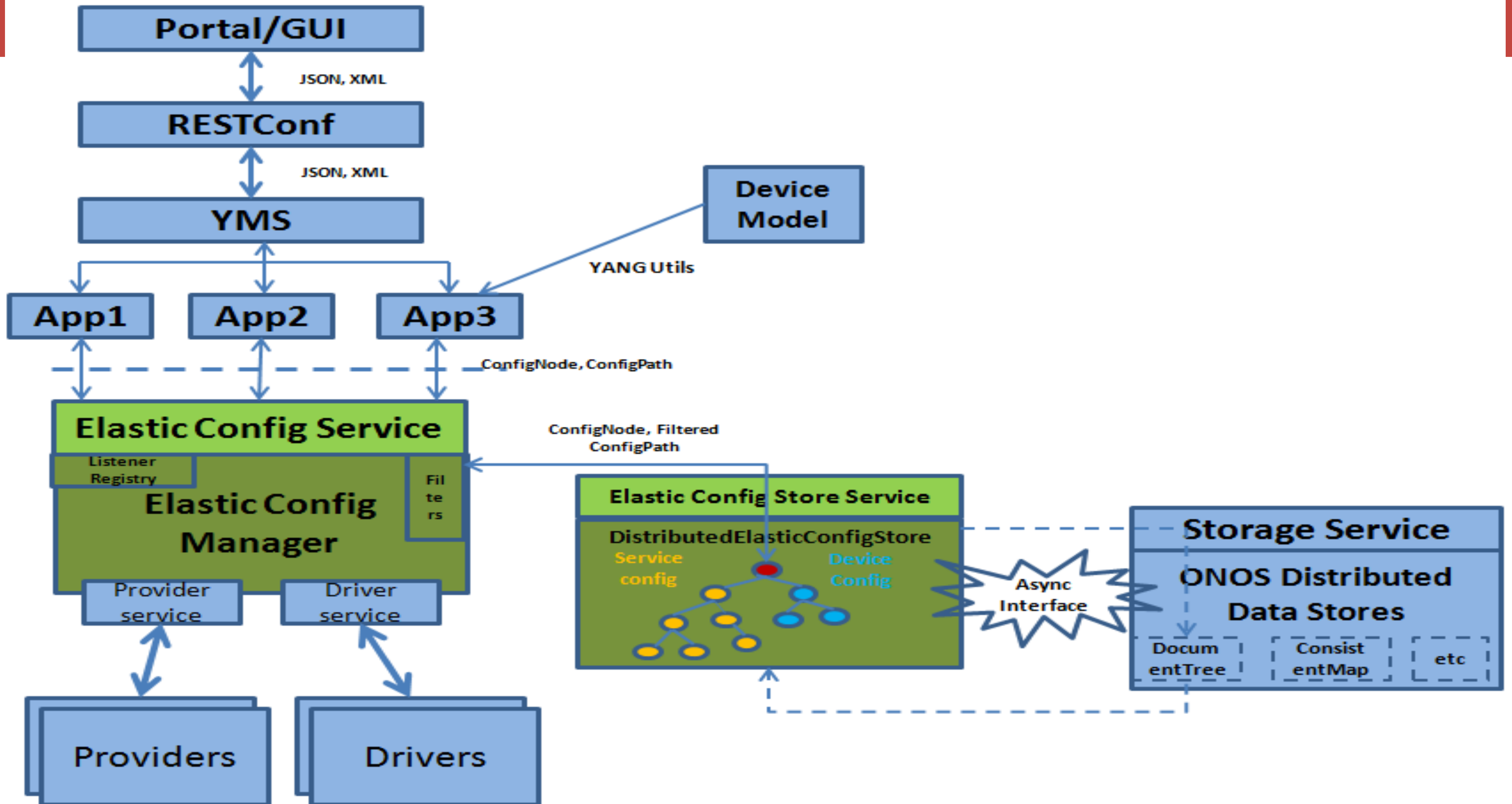
Example: YANG to YANG Data Tree

```
container nes {
*   list ne {
*     key "ne-id"
*     leaf ne-id {
*       type uuid;
*     }
*     container south-protocol-cfg {
*     }
*     list attached-controller {
*       key "controller-address"
*       leaf role {
*         type string;
*       }
*       leaf controller-name {
*         type string;
*       }
*       leaf controller-address {
*         type ipv4-address;
*       }
*       leaf controller-public-address {
*         type ipv4-address;
*       }
*       leaf snc-connect-address {
*         type ipv4-address;
*       }
*       uses controller-node;
*     }
*     leaf name {
*       type string;
*     }
*     leaf operate-status {
*       type operate-status;
*     }
*   }
}
```

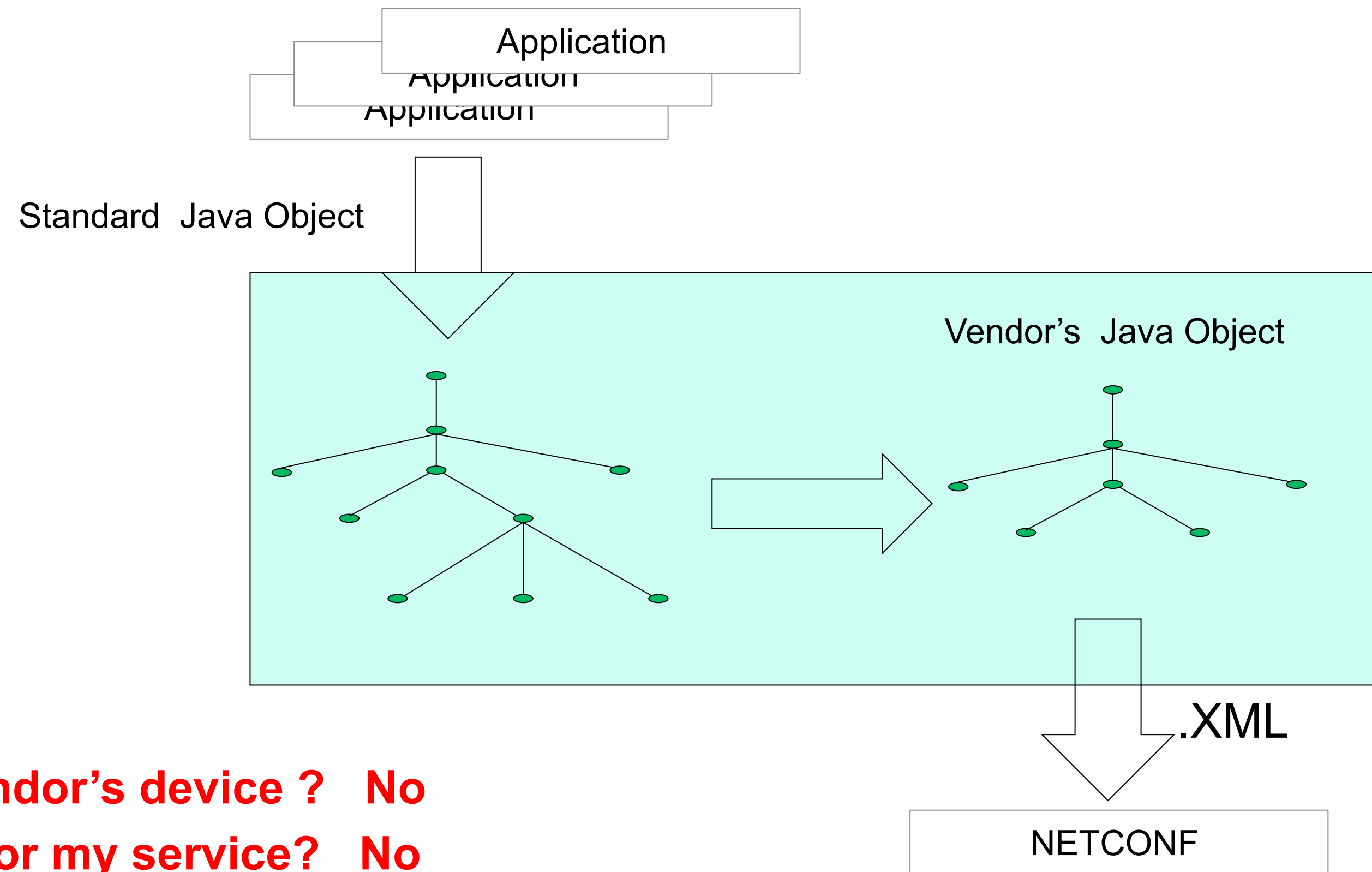








Common driver does the model translation from standard to vendor specific without code changes!

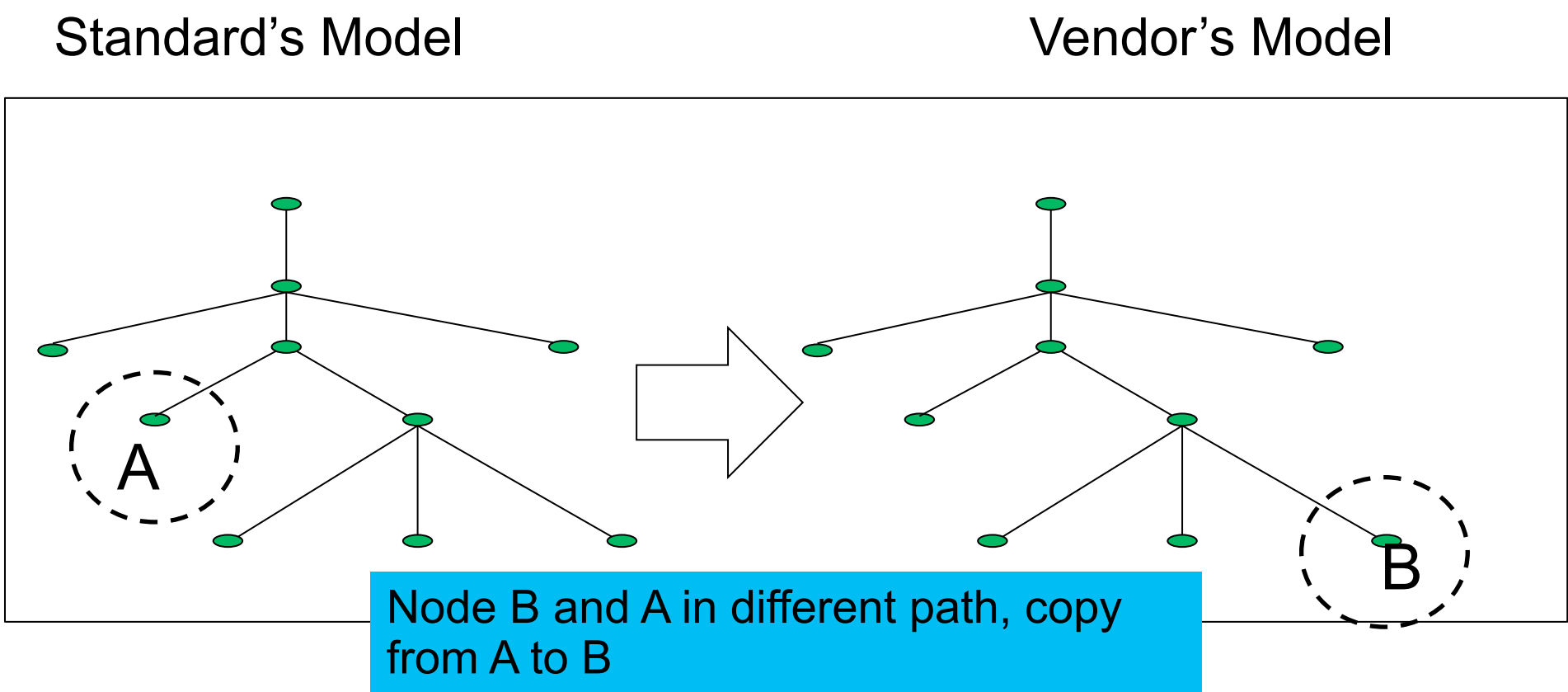
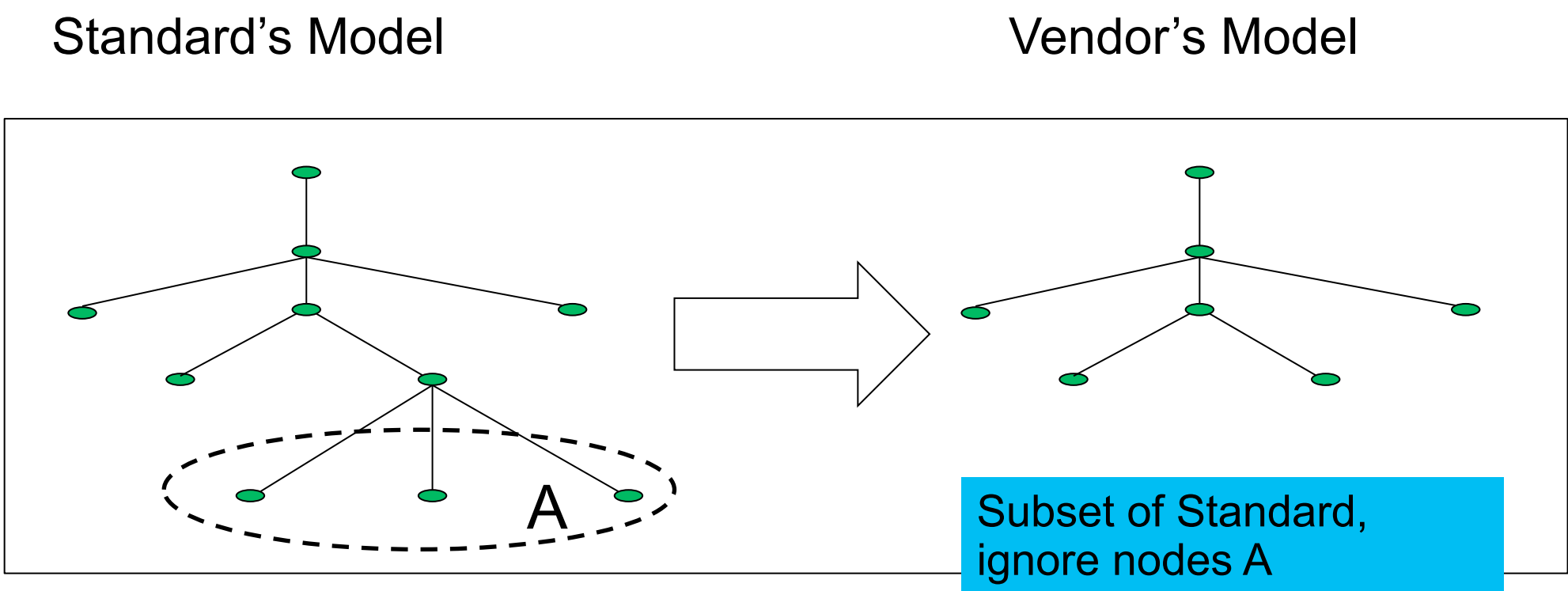
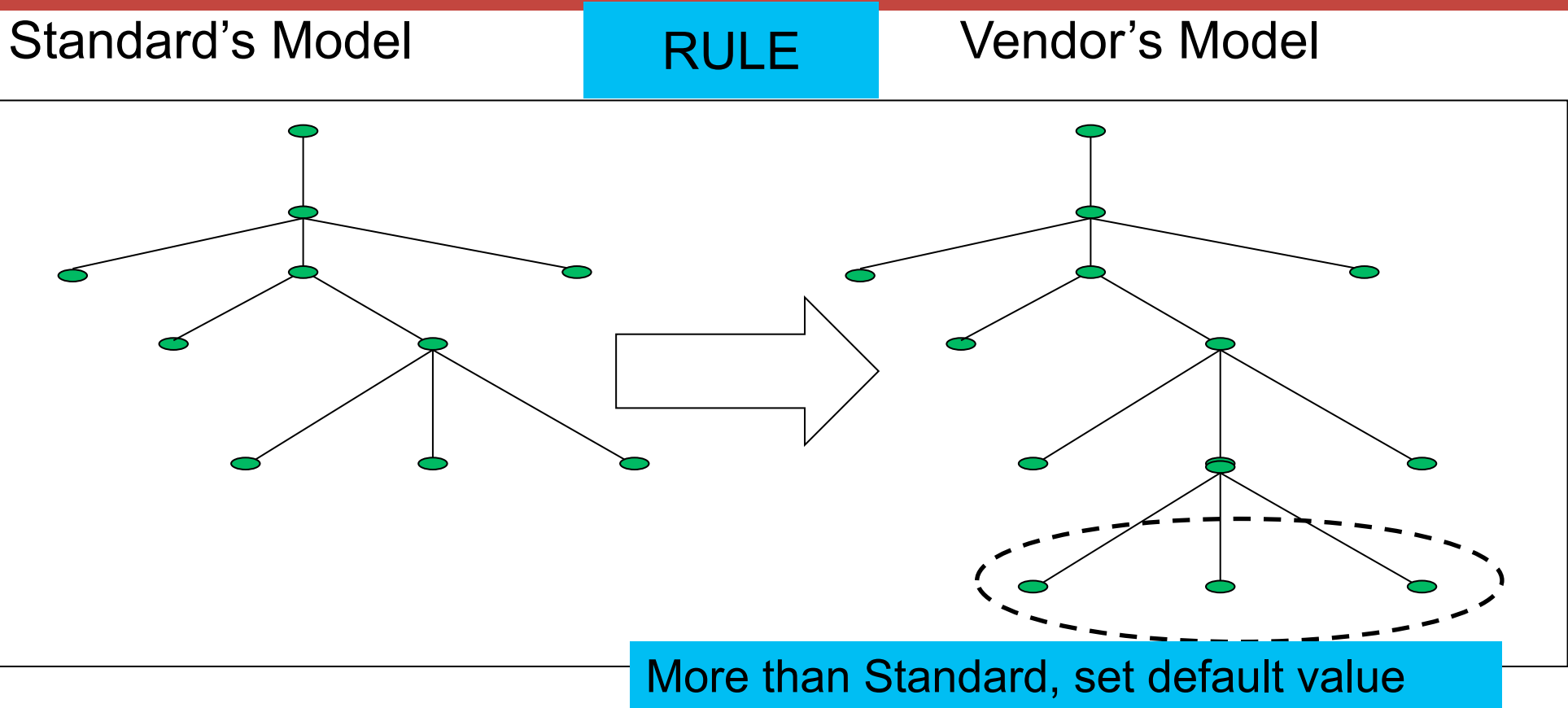
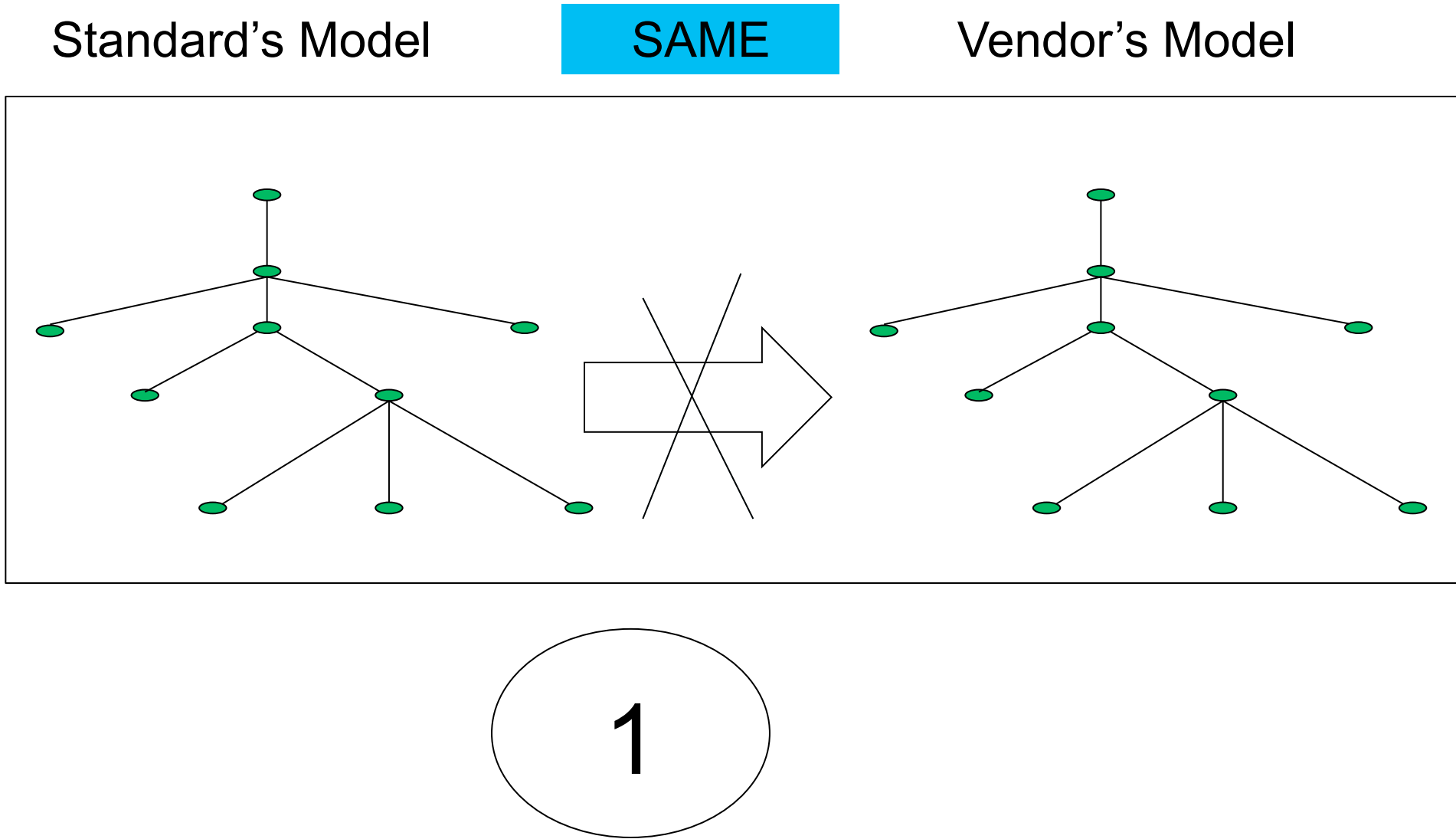


Do I need write a driver for each vendor's device ? No

Do I need define behavior in driver for my service? No

Do I need update driver code when Vendor's device model is changed? No

Translation in Common Driver



Thank You