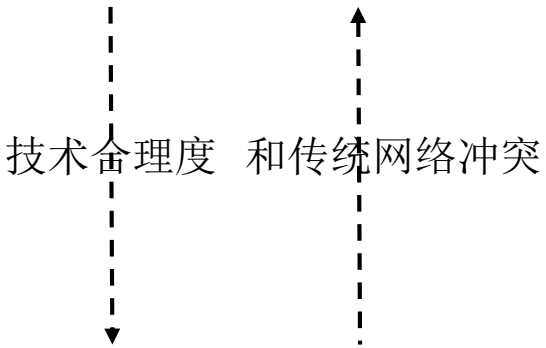
SDN实战特训营

ODL编程的三个切入点

SDN和传统网络： 发现SDN应用

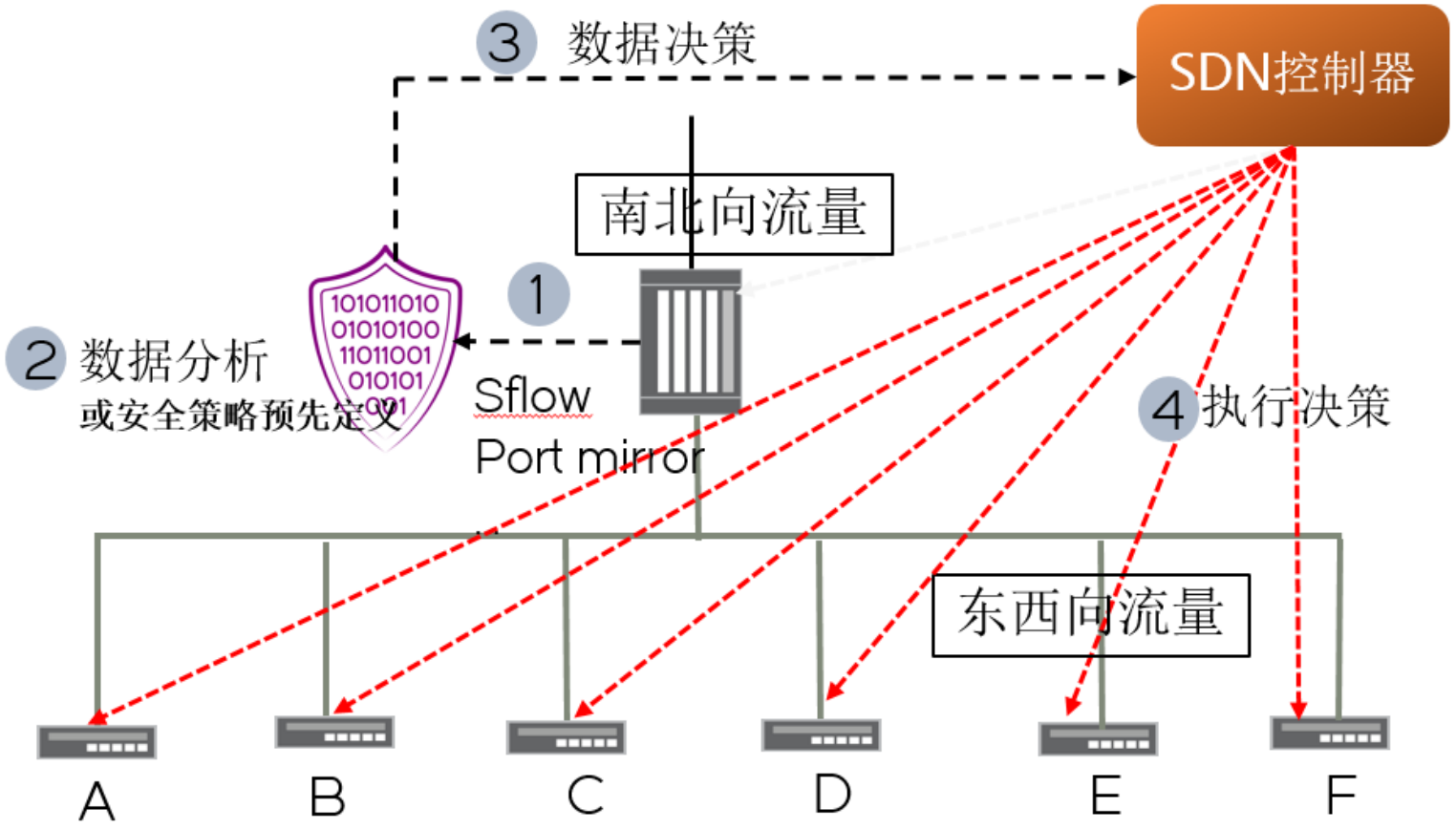
1. *他不行，你行*
2. *他也行，你更行 （譬如成本，灵活性，扩展性等等）*
3. *他也行，你也行，只是你没得玩*



- 一般来说，技术更新和社会变革都来源于如上的动机，而拯救以及建设更美好世界的动机本身未必具有全然的客观性。
- 第一个动机相比较于其他的动机来说更具有合理性（ironically，也同时具有更大的伤害性）
- 延伸到SDN的应用来说，假如我们可以设计一个应用是在传统网络很难实现，用SDN的方法更有效更经济地解决问题，并且不伤害已经存在的现有网络功能，那就是一个比较好的SDN应用。
- 所以目前来说：发现好的SDN应用本质上是发现现有网络上存在的问题，然后去探讨用SDN方法解决的可能性。某种程度上说发现SDN应用比实现更关键，需要对于现有网络和存在问题的良好认识和判断。

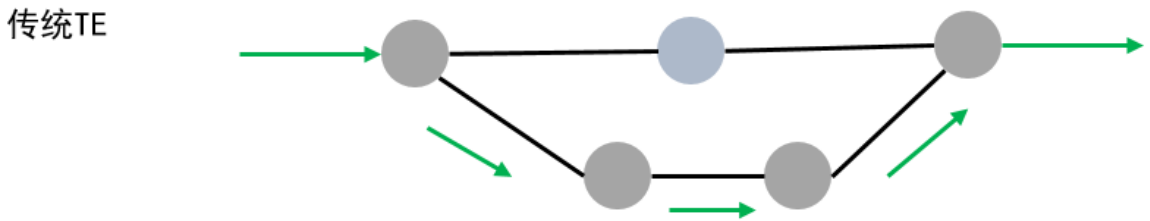
SDN的一些应用方向和需求 I

- 安全：从端点安全到分布式安全，覆盖各种网络类型和应用安全

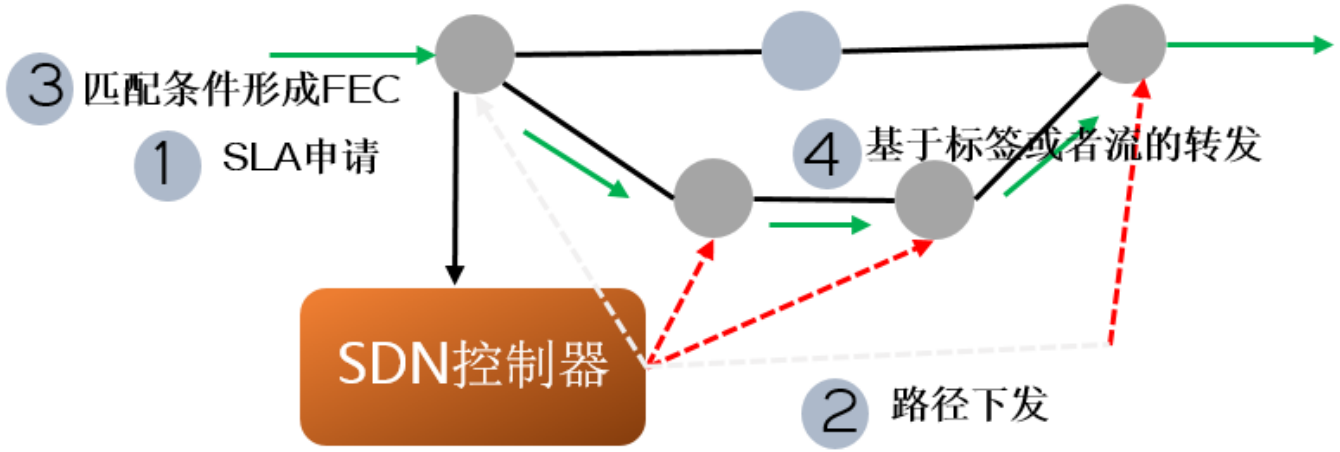


SDN的一些应用方向和需求 II

路径规划：流量工程选择合适于应用的路径和QoS保证

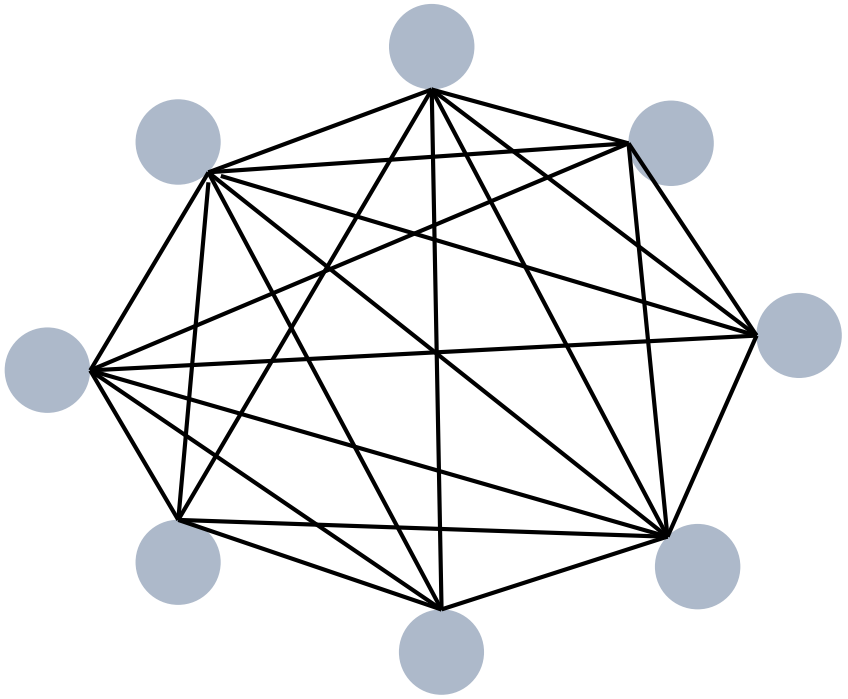


传统方式是各节点同步TE数据库，由头节点提出路径要求并计算路径，然后由沿路节点分配资源。
缺点：网络的TE计算因子和算法固定，功能增加需要全网升级。很难为某一应用选择路径。



SDN的一些应用方向和需求 III

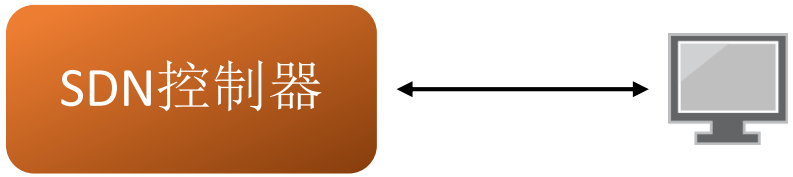
- 网络自动化运维



譬如要配置一个full-mesh的tunnel或者vpn，每增加一个node，需要增加2*(n-1)条单向链路配置，并且需要登陆所有的n节点进行配置，配置完成后也无法有效地进行正确性检查。

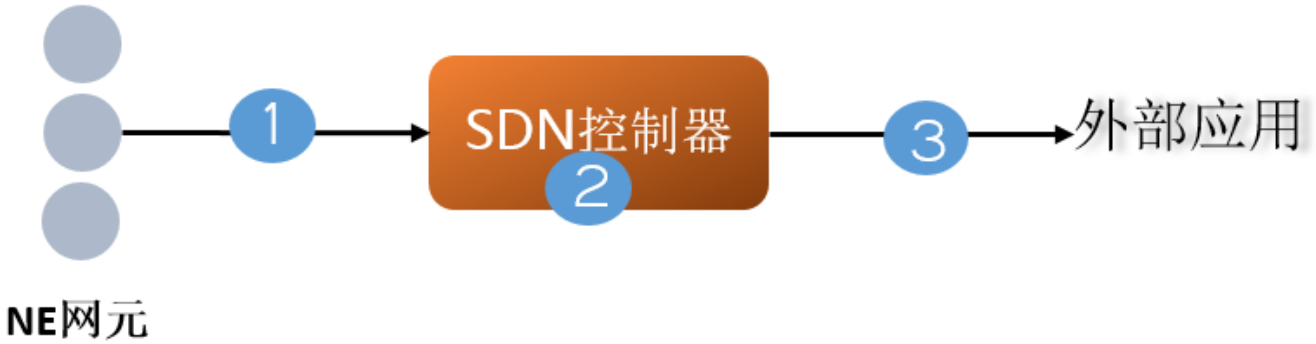
用自动化的方式可以降低人工成本以及错误的发生可能。

- **获得网络拓扑**
- **获得节点信息**
- **自动建立互联tunnel**



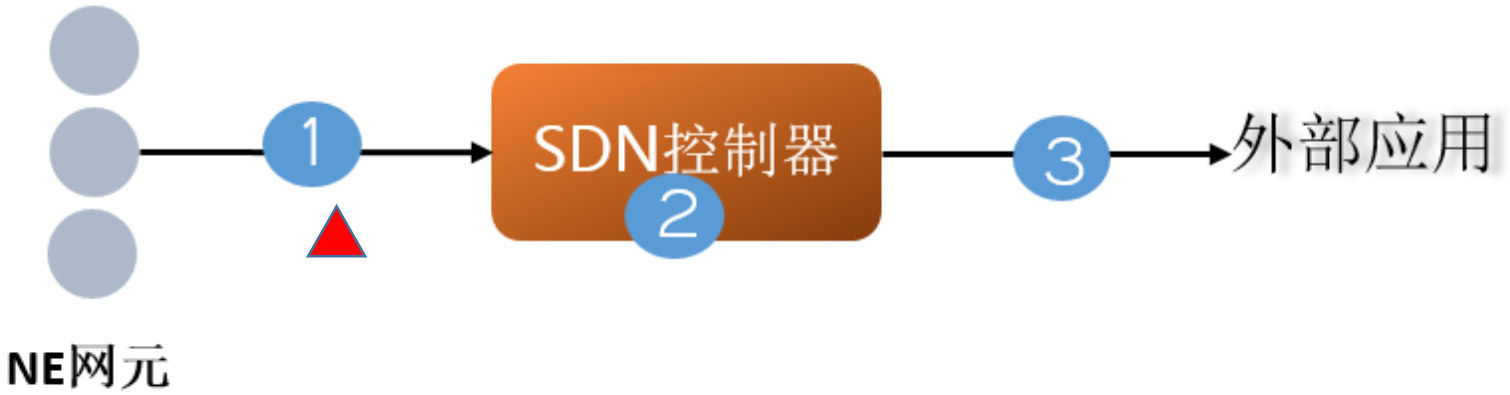
实现SDN应用的三个编程位置

1. *面向网元的协议编程*
2. *网络控制器内模块编程*
3. *面向网络控制器已有能力的编程*

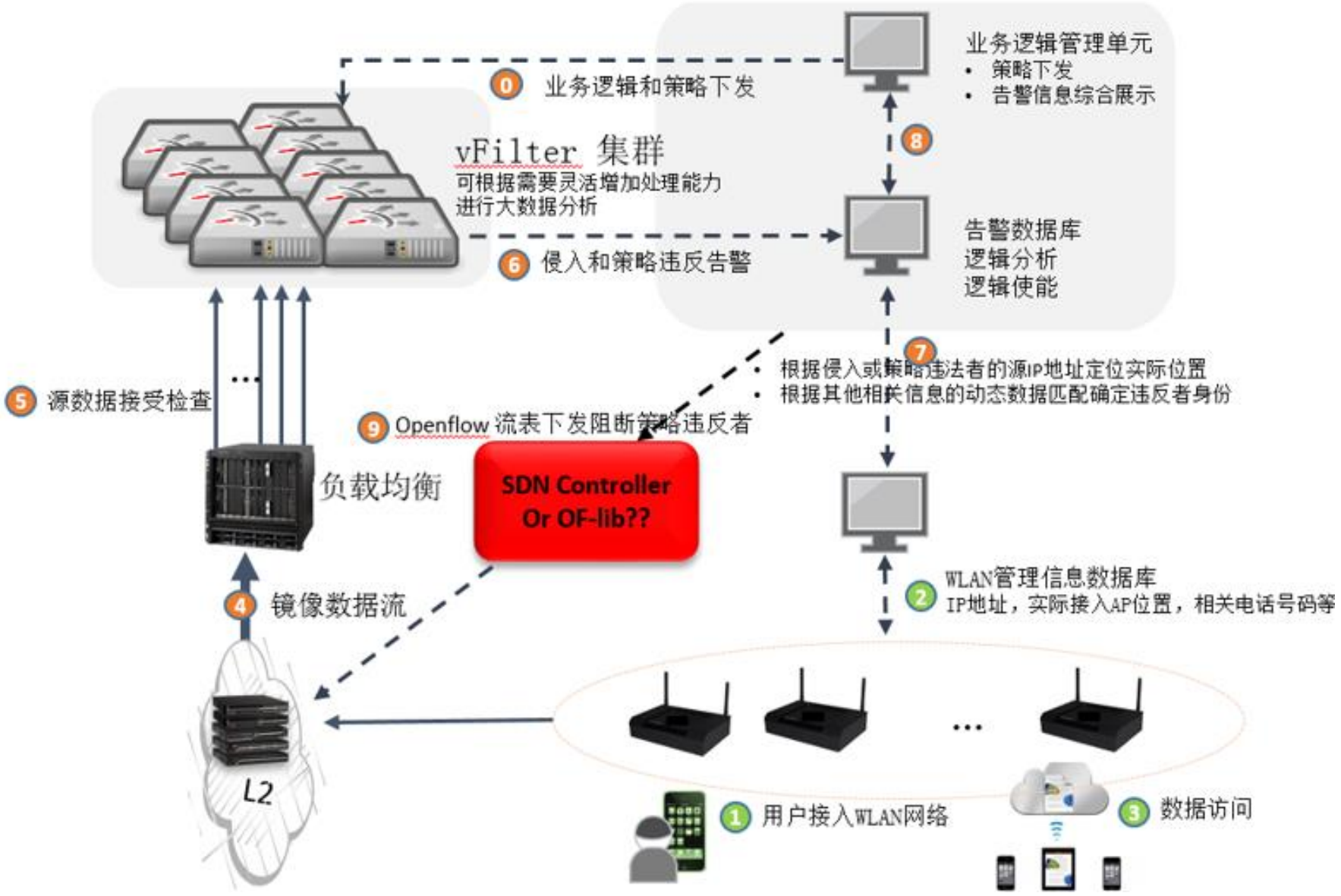


第一个编程切入点：网元级别编程

- 网络复杂度不高
- 网元类型和管理协议单一
- 拓扑固定
- 服务固定
- 服务抽象再向上提供服务的必要性不大
- 面向特定的网元管理协议的编程或许就可以解决问题

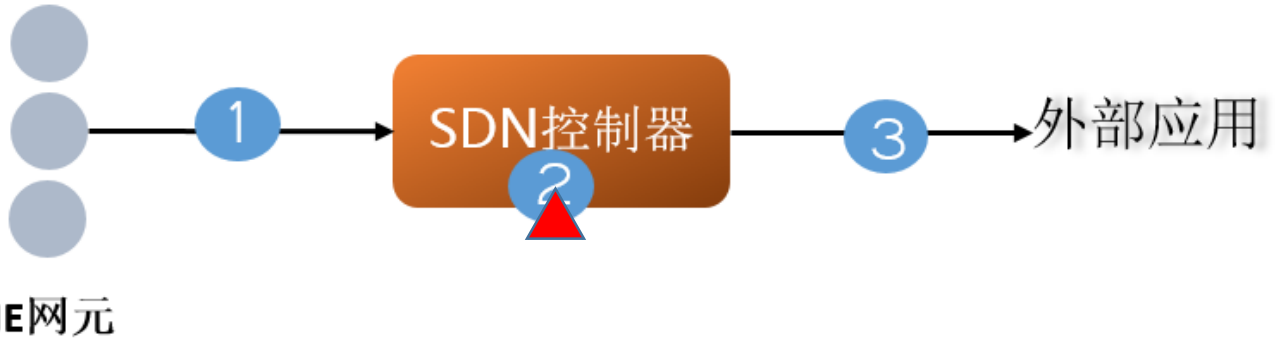


SDN：某个安全厂商的集成方案



第二个编程切入点： 控制器内编程

- 网络复杂
- 网元众多，类型各异
- 对于网络管理的冗余性和高可用性要求高
- 服务需要再次抽象
- 实现服务模块的可重用性

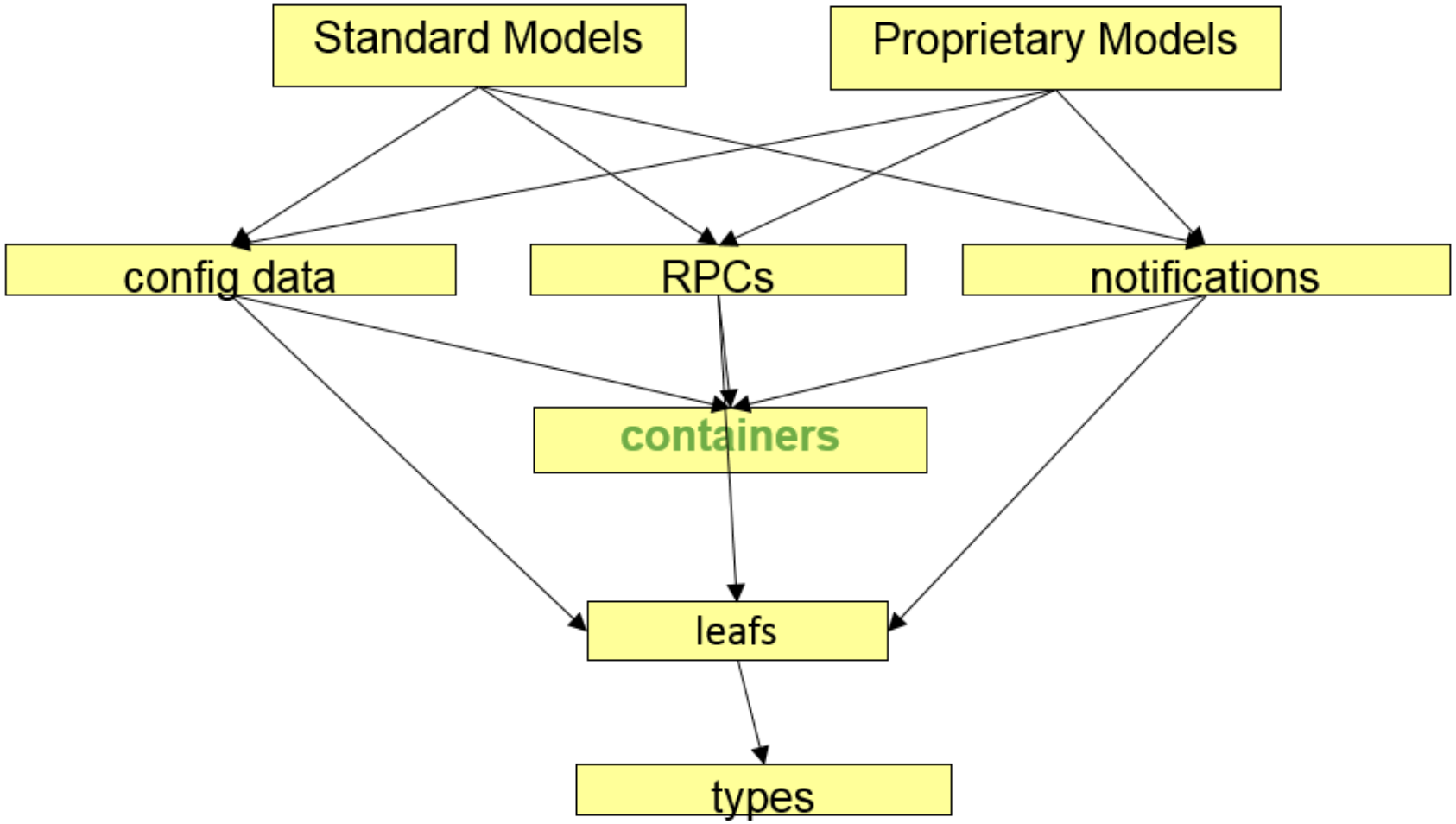


数据建模

	SNMP	NETCONF	SOAP	REST
Standard	IETF	IETF	W3C	-
Resources	OIDs	Paths		URLs
Data models	Defined in MIBs	YANG Core Models		
Data Modeling Language	SMI	YANG	(WSDL, not data)	Undefined, (WSDL), WADL, text…
Management Operations	SNMP	NETCONF	In the XML Schema, not standardized	HTTP operations
Encoding	BER	XML	XML	XML, JSON,…
Transport Stack	UDP	SSH TCP	SSL HTTP TCP	SSL HTTP TCP

"RESTConf"

数据建模： YANG 模型

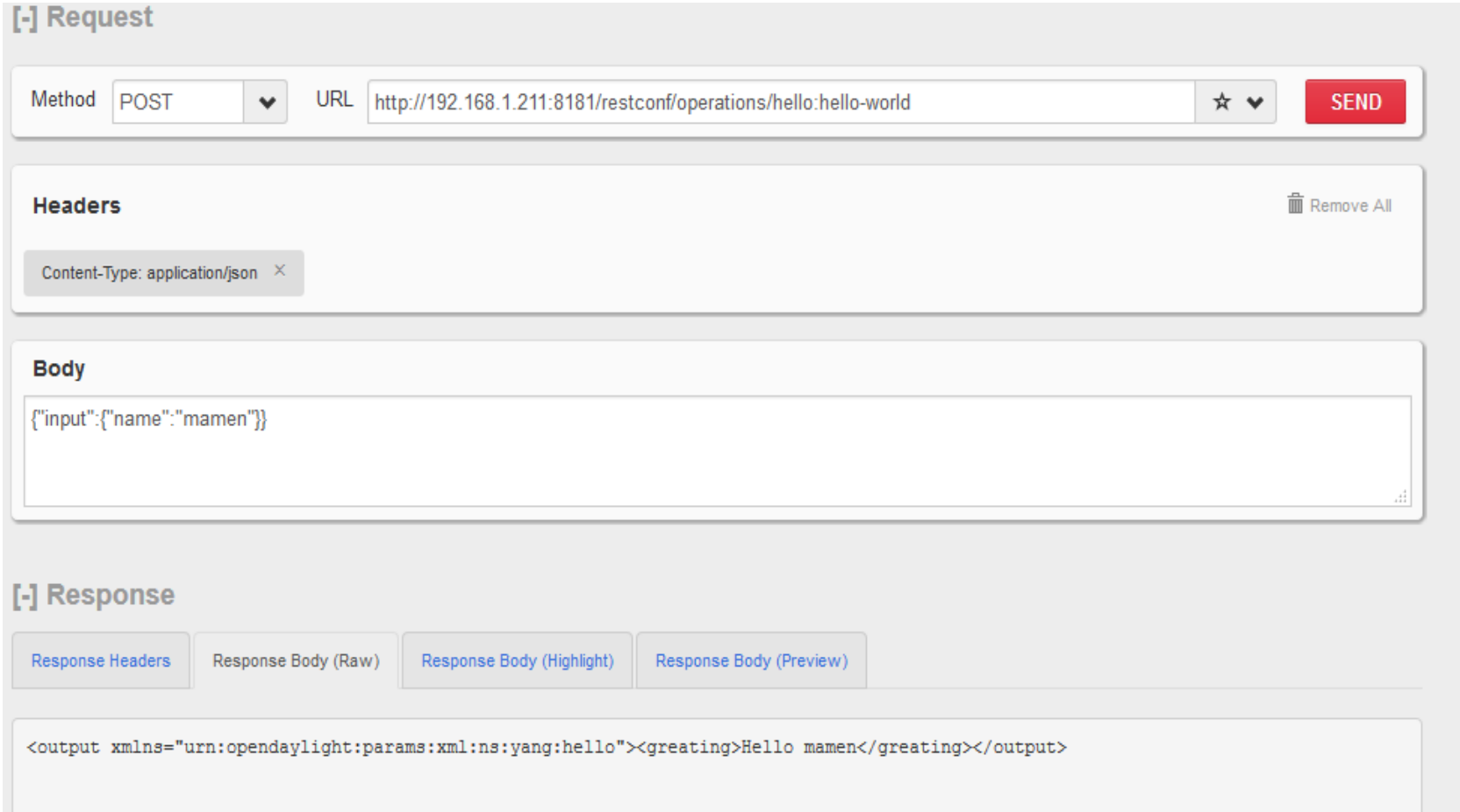

Standard Models
Proprietary Models
config data
RPCs
notifications
containers
leafs
types

数据建模： YANG 模型例子

```
module hello {
    yang-version 1;
    namespace "urn:opendaylight:params:xml:ns:yang:hello";
    prefix "hello";

    revision "2015-01-05" {
        description "Initial revision of hello model";
    }
    rpc hello-world {
        input {
            leaf name {
                type string;
            }
        }
        output {
            leaf greating {
                type string;
            }
        }
    }
}
```

```
[root@localhost impl]# ls -l
total 8
-rw-r--r-- 1 root root 1351 Aug 31 02:46 HelloProvider.java
-rw-r--r-- 1 root root  974 Aug 31 02:36 HelloWorldImpl.java
```



第三个编程切入点：面向网络控制器已有能力的编程

- SDN controller/ODL已经提供了现有丰富的北向接口
- 直接使用SDN控制器提供的北向接口编程，而不是重新发明轮子
- 结合高层应用逻辑，调用SDN控制器API来实现组合逻辑