

Traffic Engineering In Floodlight

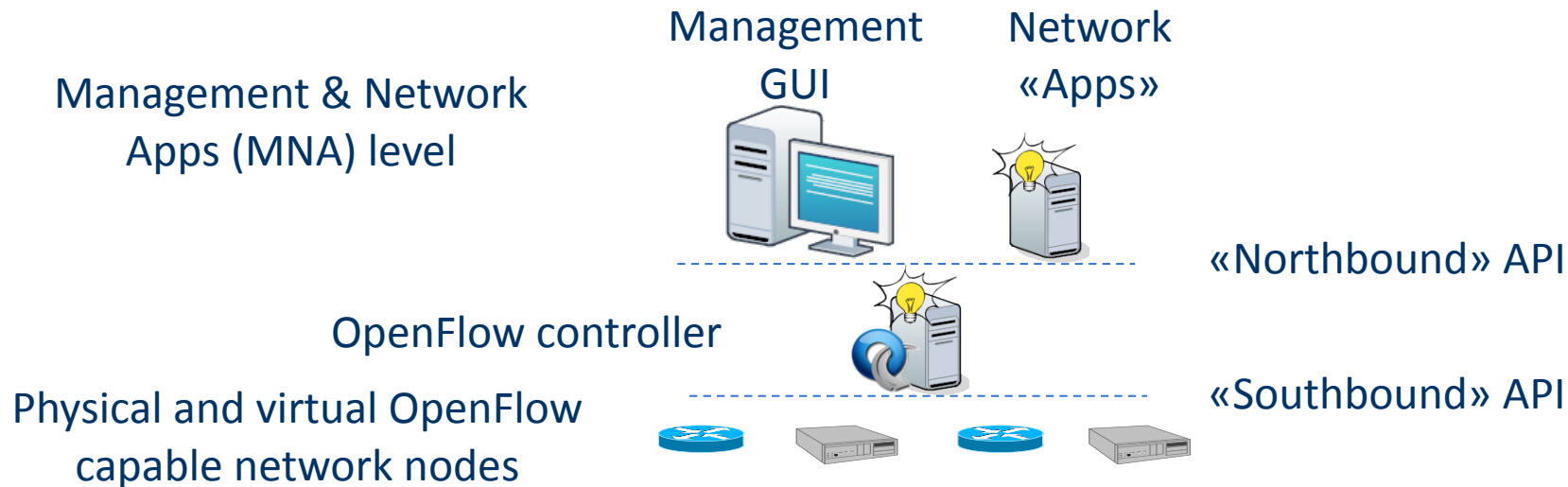
Candidato

Pier Luigi Ventre

Tutor

Stefano Domenico Salsano





- **Software Define Networking:**

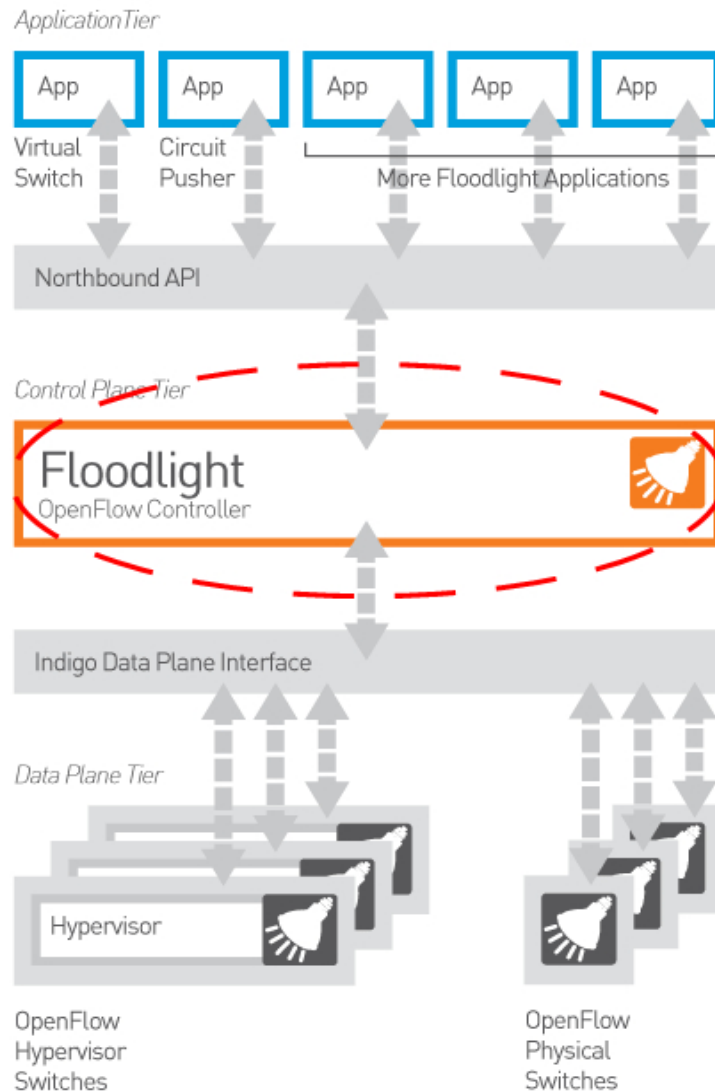
- Enables innovation in the infrastructure
- More “Apps” and Service

- **OpenFlow**

- Decouples control plane from data plane;

- Introduces Controller, a new network element;
- Open standard protocol;
- Enables researchers to run experimental protocols;
- “Full-duplex” link;
- Network intelligence

Environment And Goals



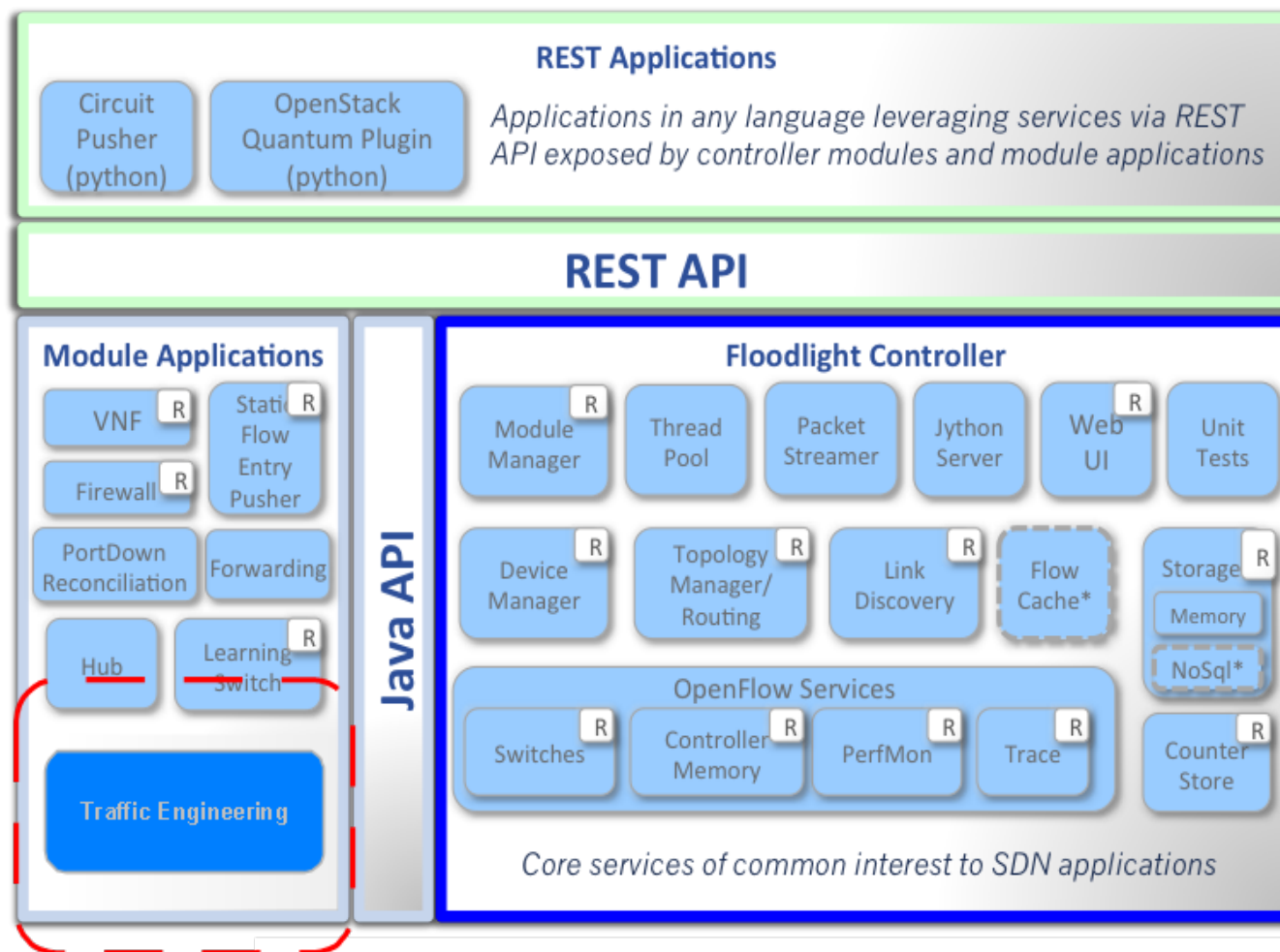
- **Floodlight is a SDN Controller:**

- Open (Apache-licensed)
- Full support OpenFlow protocol
- Open community, Easy to Use, Tested and Supported
- Java-based

- **Aim of this project :**

- Add a new module for Traffic Engineering

Quick Look To Floodlight



- Modular Architecture
- Controller Modules and Application Modules

A Little Example

```
public class MyModule implements IOFMessageListener,
    IFloodlightModule {
    protected IFloodlightProviderService floodlightProvider;
    ...
    @Override
    public Collection<Class<? extends IFloodlightService>>
        getModuleDependencies() {
        // Return Module's dependencies
    }
    @Override
    public void init( FloodlightModuleContext context) throws
        FloodlightModuleException {
        // Initialize the module
    }
    @Override
    public void startUp( FloodlightModuleContext context) {
        // Register Event Listener
    }
    ...
    - Callback Function
    - Rest Api
    - Function
    ...
```

- The interfaces to implement
- The service provider
- Module's dependencies
- Initialization
- Phase of startup
- Module's Function

TE Module

- ☐ Rest API;
- ☐ Use Topology Service;
- ☐ CSPF;
- ☐ Reservation Function;
- ☐ Static flow and traffic aware flow;
- ☐ Flow management;
- ☐ Auto-Route;
- ☐ Auto-Bandwidth flow;
- ☐ Resiliency;

TE Module

- ☐ Rest API;
- ☐ Use Topology Service;
- ☐ CSPF;
- ☐ Reservation Function;
- ☐ Static flow and traffic aware flow;
- ☐ Flow management;
- ☐ Auto-Route;
- ☐ Auto-Bandwidth flow;
- ☐ Resiliency;

DREAMER

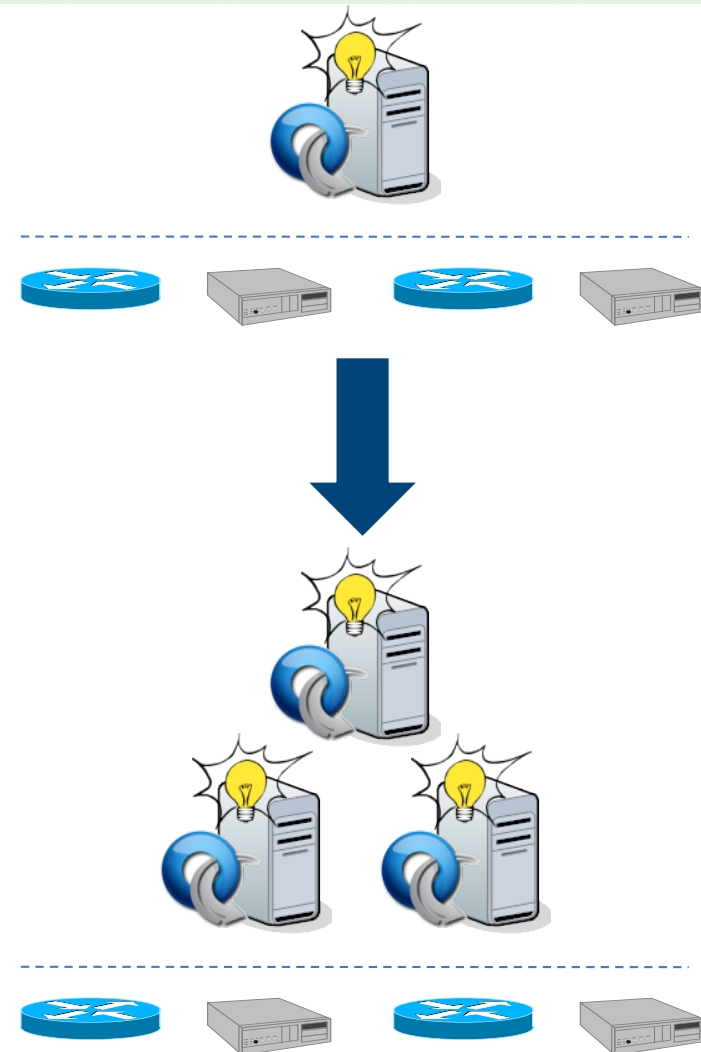
*Distributed REsilient sdn Architecture
MEeting carrier grade Requirements*

- Partners:

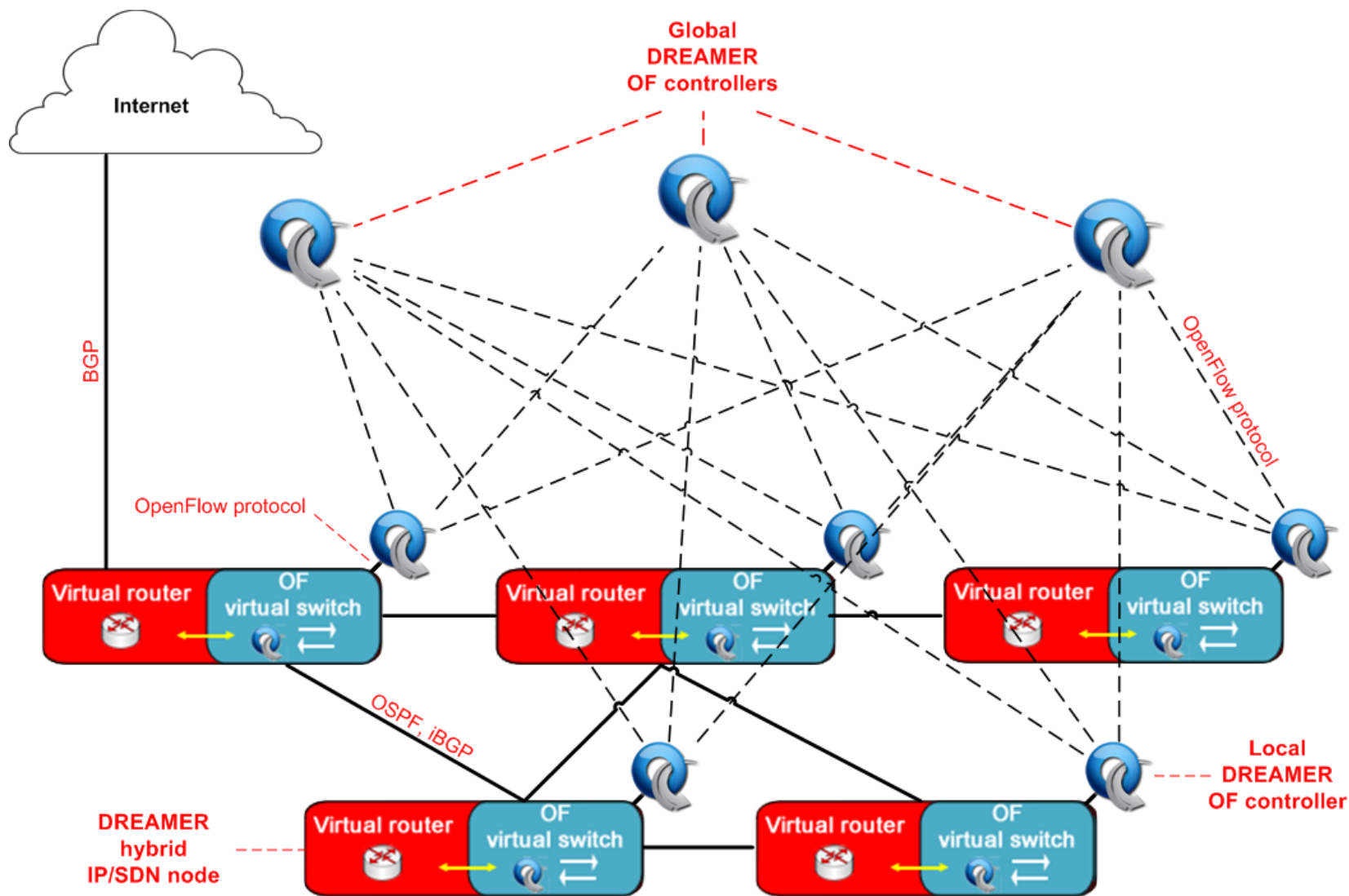


- Carrier Grade Requirements  Availability $\geq 99,999 \%$

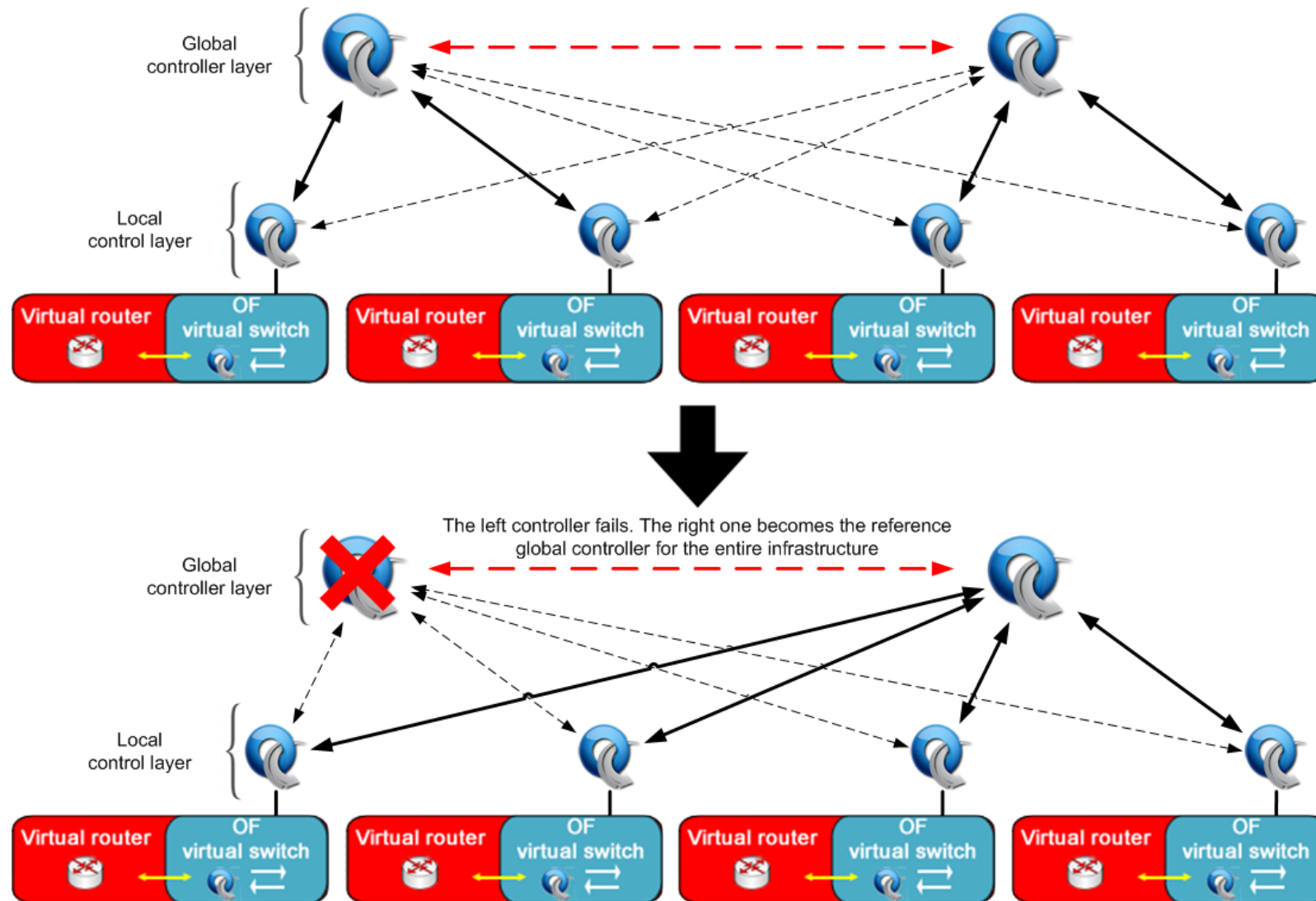
- **SND's Issue:**
 - Centralized architecture;
- **Two level of Resilience:**
 - Data plane
 - Control plane
- **Solution:**
 - Distributed architecture;
 - Promptly restore
 - Consistency



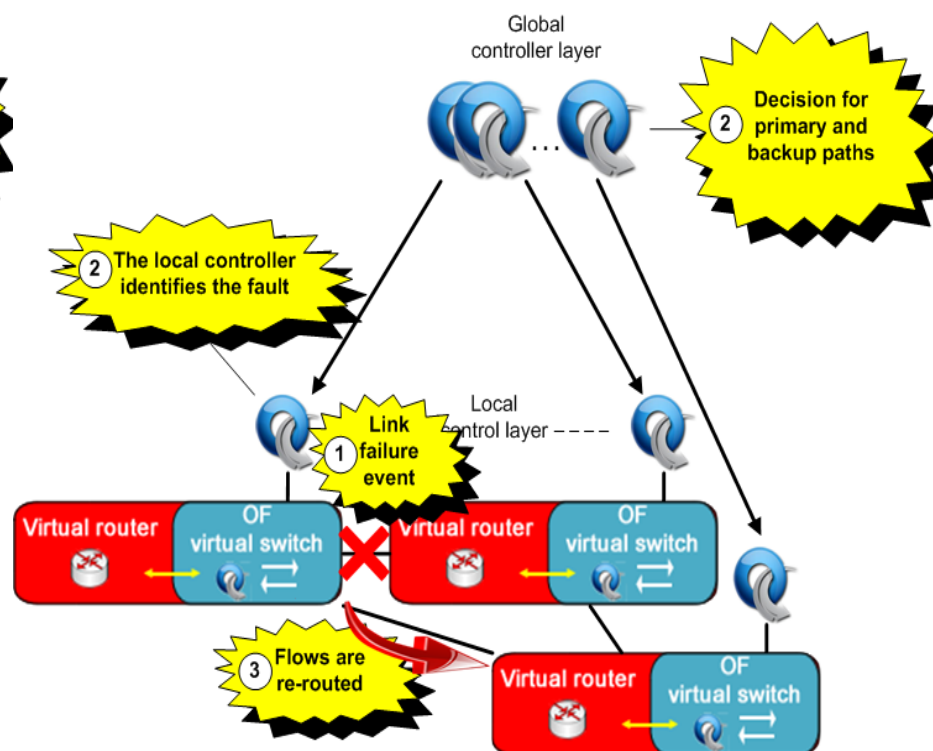
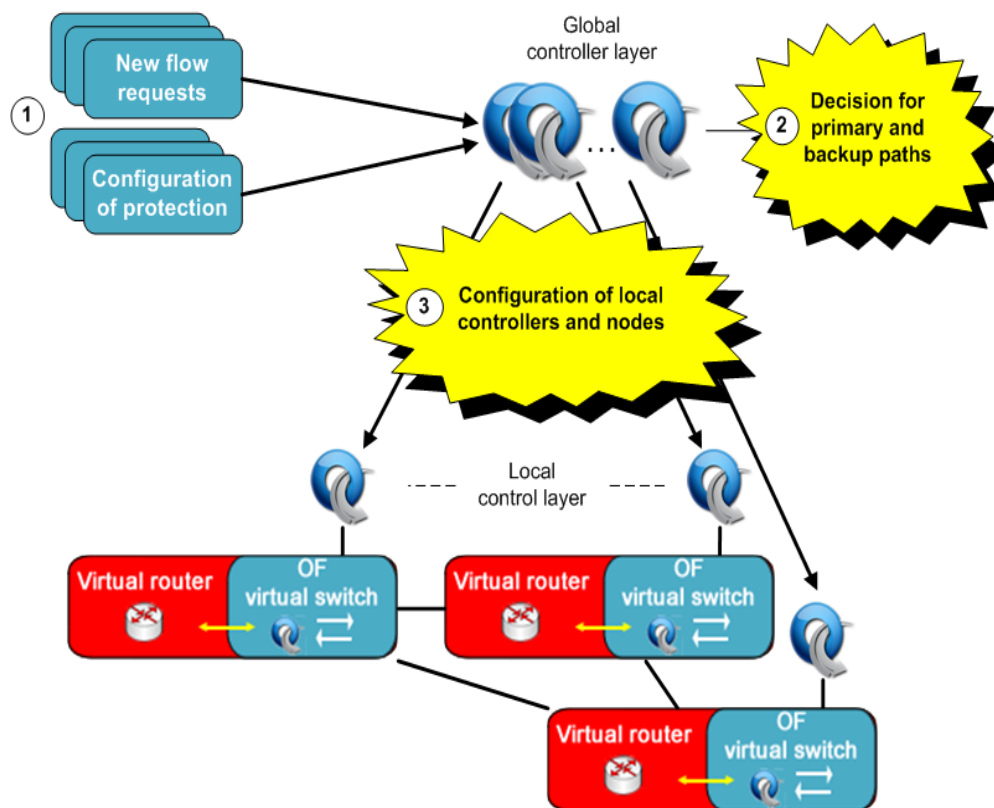
DREAMER – Conceptual Architecture



DREAMER – Control Plane Resilience



DREAMER – Data Plan Resilience



- Questions ?