

DIStributed SDN COntrollers for rich and elastic services

IETF89

- **DISCO: DIStributed SDN COntrollers for rich and elastic services**
 - French National project (ANR INFRA 2013)
 - Industrial project
 - Starting date: 01/2014
 - Duration: 42 months
- **Partners:**
 - Thales Communications & Security
 - INRIA Sophia Antipolis
 - ENS Lyon
 - 6WIND

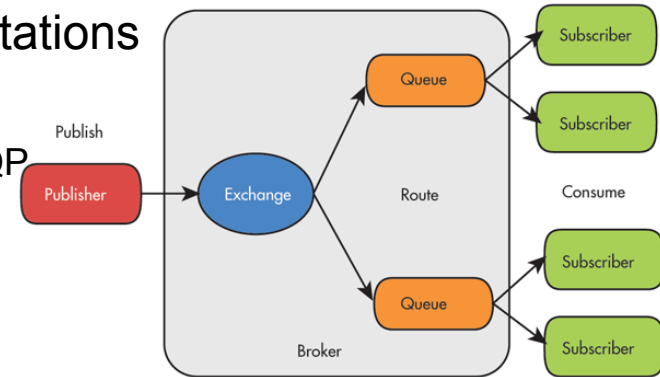
- **Resilient and scalable OpenFlow control plane**
 - How to resist to network failures?
 - How to avoid overloading the controller?
- **Rich and elastic network services**
 - When the network and all its components are virtual...
 - Where to place virtual appliances?
 - How/when to move virtual appliances?
 - What physical resources to allocate them?
 - How application can benefit from this versatility?

Communications for resilient distributed controllers

- **Resilient SDN architecture:**
 - no single point of failure
 - increase responsiveness and reliability
 - scalable
 - adaptable to heterogeneous/asymmetric deployments (e.g., SATCOM link)
- **Decompose the network in domains**
 - a mechanism enables to manage Master/Slave controllers in a domain

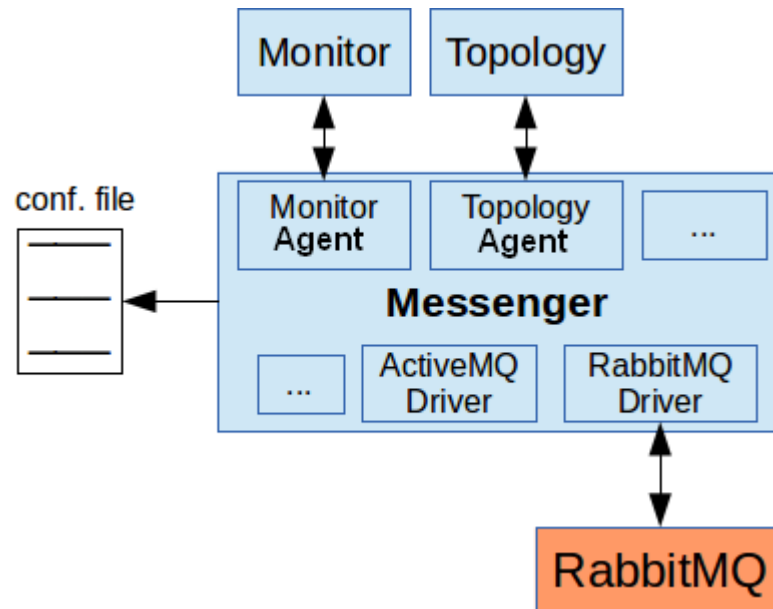
- **Distributed Broker Architecture: AMQP (Advanced Message Queuing Protocol – OASIS Standard)**

- With ActiveMQ, RabbitMQ or other implementations
- Complexity
 - The system uses the simplicity and efficiency of AMQP
- Interoperability
 - Every system can use the framework
- Resilience
 - Each subnet is independent and in case of failure, the remaining Controllers could handle the orphan switches
- Westbound communications ensured with AMQP protocol
 - underlying mechanisms are transparent and AMQP brokers ensures all other functionalities (heartbeats, failure handling, ...)

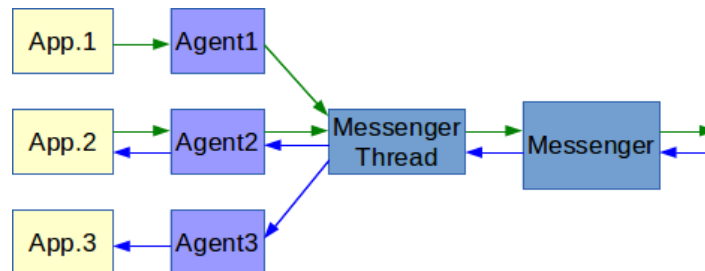


■ Overview of Messenger

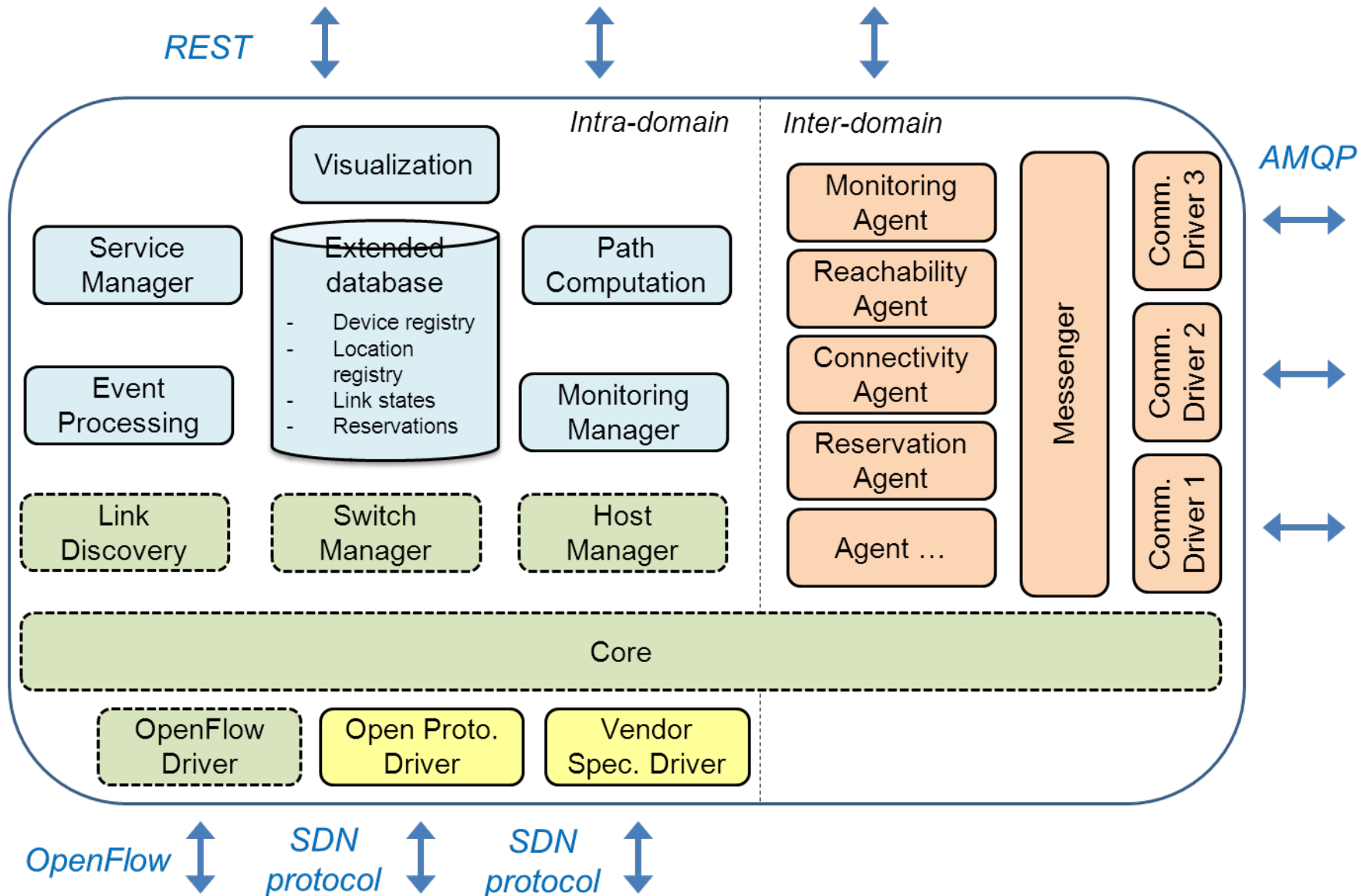
- An application to make the link between the different controller's modules
 - e.g., Link Discovery, Device Manager, ...
- Applications are agnostic of the system thanks to “agents”
 - Agents use the applications REST APIs to retrieve and push data from/to applications
- Extension of the system with “drivers”.
 - Messenger uses high level methods (e.g., sendMessage, establishConnection)
 - The driver translates them to AMQP commands



- **AMQP driver must implement:**
 - pair (neighbor controller ID)
 - setup inter-domain control channel with a neighbor controller
 - unpair (neighbor controller ID)
 - terminates inter-domain control channel
 - subscribe (topic)/unsubscribe topic (topic)
 - add/delete a topic in/from the interest of the node
 - send (topic, message)/receive(topic, message)
 - send/receive a message on a specific topic



Disco architecture



Rich and elastic network services

- **SDN allows to virtualize network appliances**
 - (e.g., load balancers, ciphers, DPI, firewall)
 - **Virtual network appliances allows to place appliances where and when they are needed**
 - (e.g., flash crowd, energy reduction when not necessary)
 - **Mobile appliances implies agile resource (re)allocation**
 - (e.g., a load balancer and a cipher do not have the same processing needs)
- => How to (re)allocate resources?**

- **Multiple devices**

- Decide where to place each appliance
 - Based on traffic and policies

- **Multicore devices**

- Provide a mechanism to increase and decrease the number of cores allocated to a virtual appliance
 - Dynamically and without service interruption

- **API**

- Define an interface for virtual appliances to reserve and release resources

DIStributed SDN COntrollers for rich and elastic services

Damien Saucez (on behalf of the consoritium)