

SDN controller: Intent-based Northbound Interface realization for extended applications

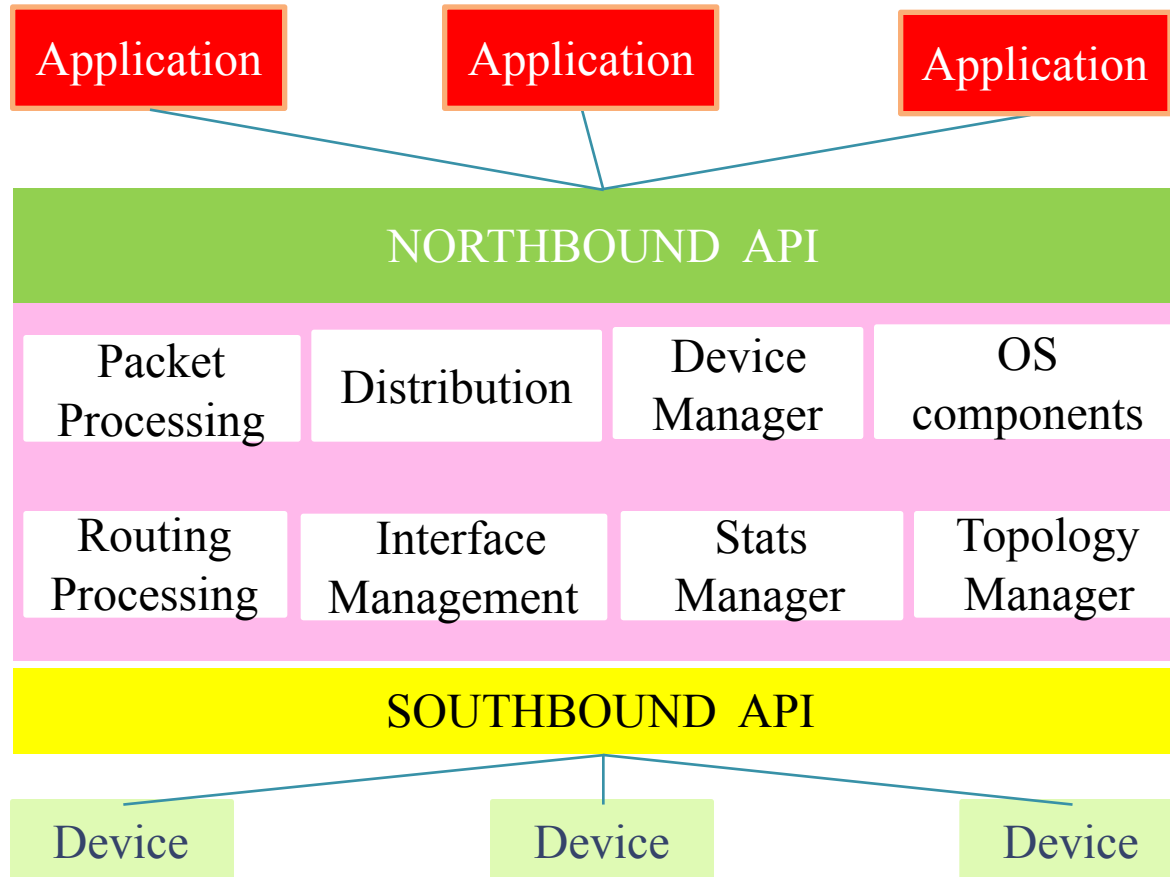
1. Introduction
2. SDN Controller
3. Intent-based Northbound Interface (NBI)
4. The Intent framework in ONOS controller
5. The proposed architecture for extended applications: requirement, Micro-service architecture, three-tier application architecture, domain-driven design
6. Prototype
7. Conclusion & Discussion

Minh Pham, Doan Hoang
University of Technology Sydney, Australia

Introduction

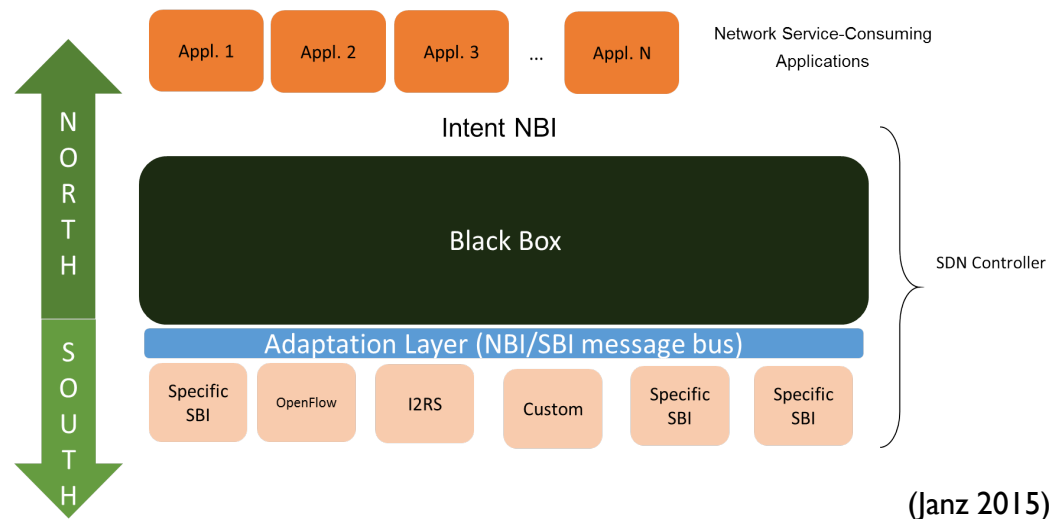
- The adoption of SDN introduces an emerging class of *extended network applications*
- Open Networking Foundation (ONF) principles of intent-based NBI *lack functionality to support the above extended applications*
- We *adopt the Intent-based NBI principles* and propose an architecture to realise extended network applications
- The proposed architecture is built on *Micro-service architecture, Domain driven design and three-tier application architecture*
- The prototype is to build the Dynamic Resource Application (DRM) using Open Network Operating System (ONOS) controller

SDN Controllers



Intent-based NBI

Intent-Based SDN



(Janz 2015)

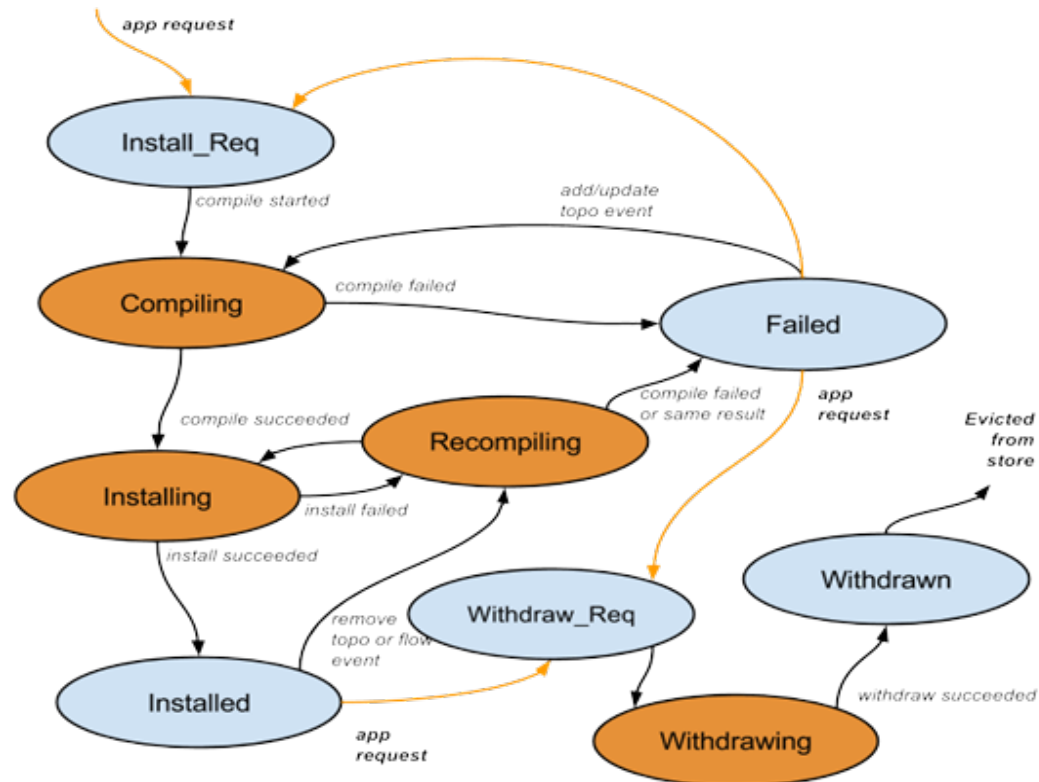
- ONF identified two types of NBI usages: prescribed and intent-based
- In intent-based NBI, users describe their request in normal conversation language, it is the WHAT question, not HOW question
- Each request has two main components: substance and constraint, substance contains the objects and constraint contains the conditions

Open Network Operating System (ONOS)

Intent framework

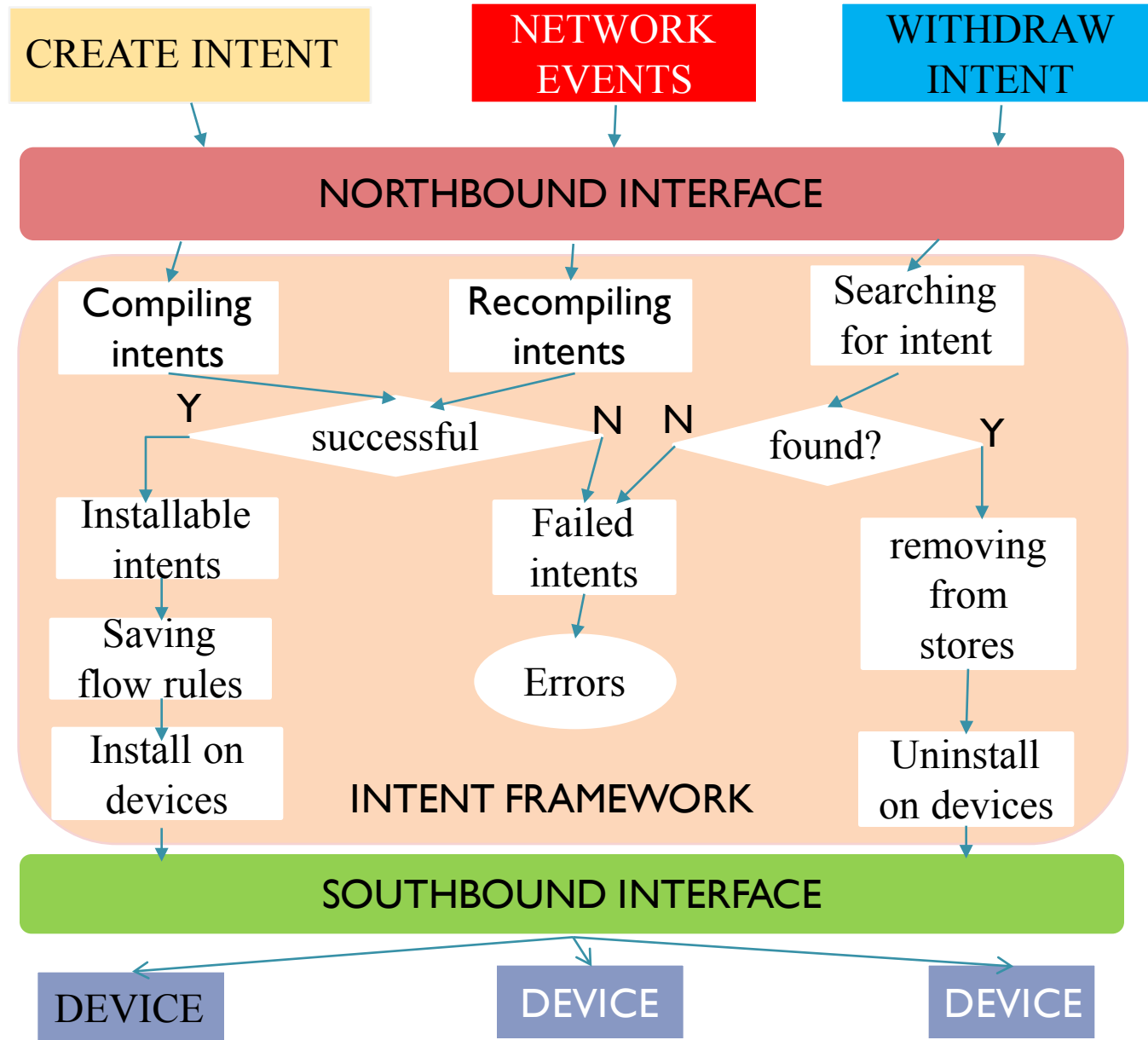
- ONOS provides application intent framework as the NBI to describe network connectivity as network policy. It is a sub-system of ONOS.
- Intents in the framework are organised into a hierarchy; developers will select the intents that are closest to their network models.
- Each intent has its compiler to compile the intent into flow rules; flow rules are installed as table entries in network devices based on the Openflow protocol.

ONOS Intent life cycle

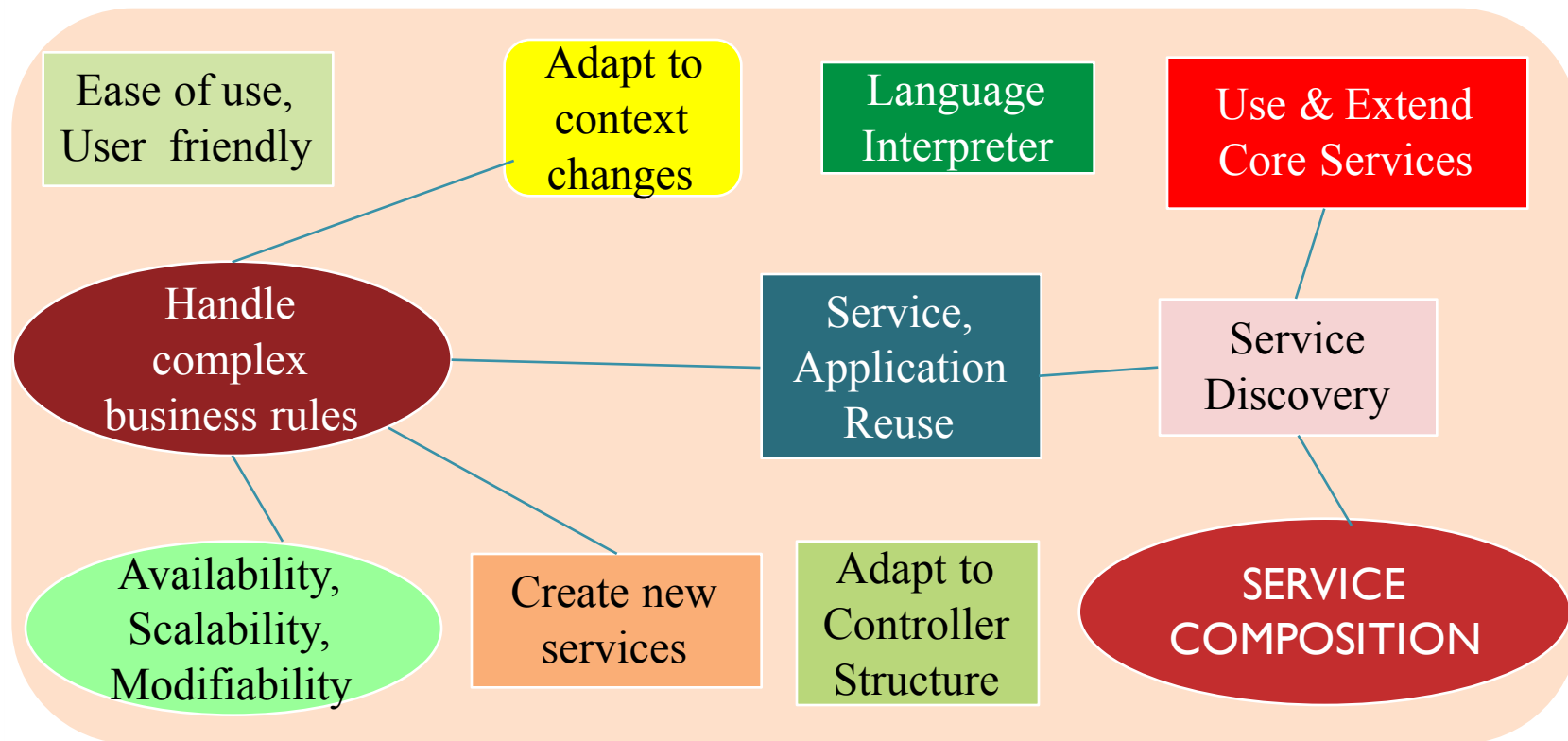


Intent life cycle
(ONOS, 2014)

ONOS Intent framework



The proposed architecture - Requirements

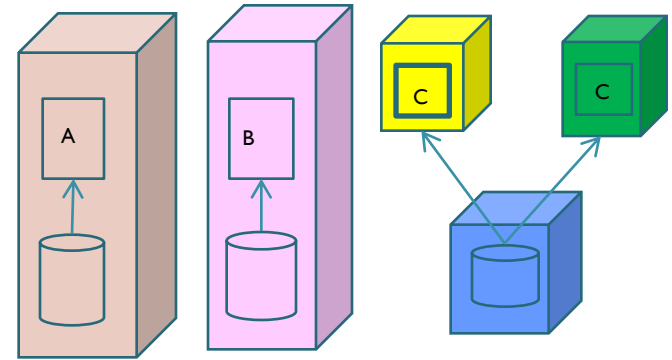


- Service composition is the main attribute to support intent's composability attribute
- Requiring creating of new services, reuse of existing services and composing them into applications
- Other requirements: user friendly, availability, scalability, modifiability

Microservice architecture (MSA)



Fine-grained Service Composition



Decentralised data management
(Fowler, 2013)

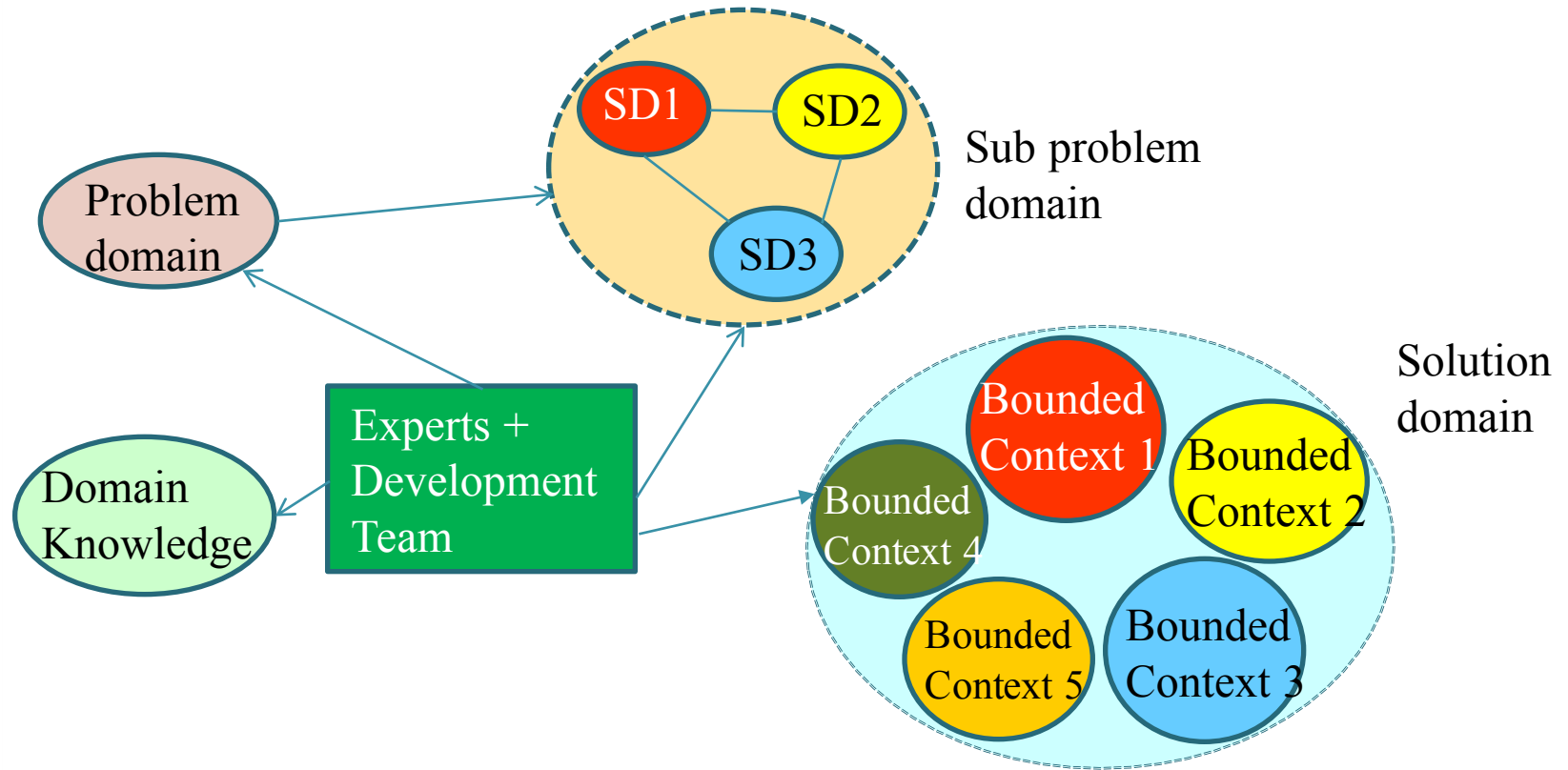
MSA design principles:

- Data decentralized
- Componentized application
- Application design robustness
- Process isolation

Patterns: Self registration, Service registry, Client service lookup
(Richardson, 2014)

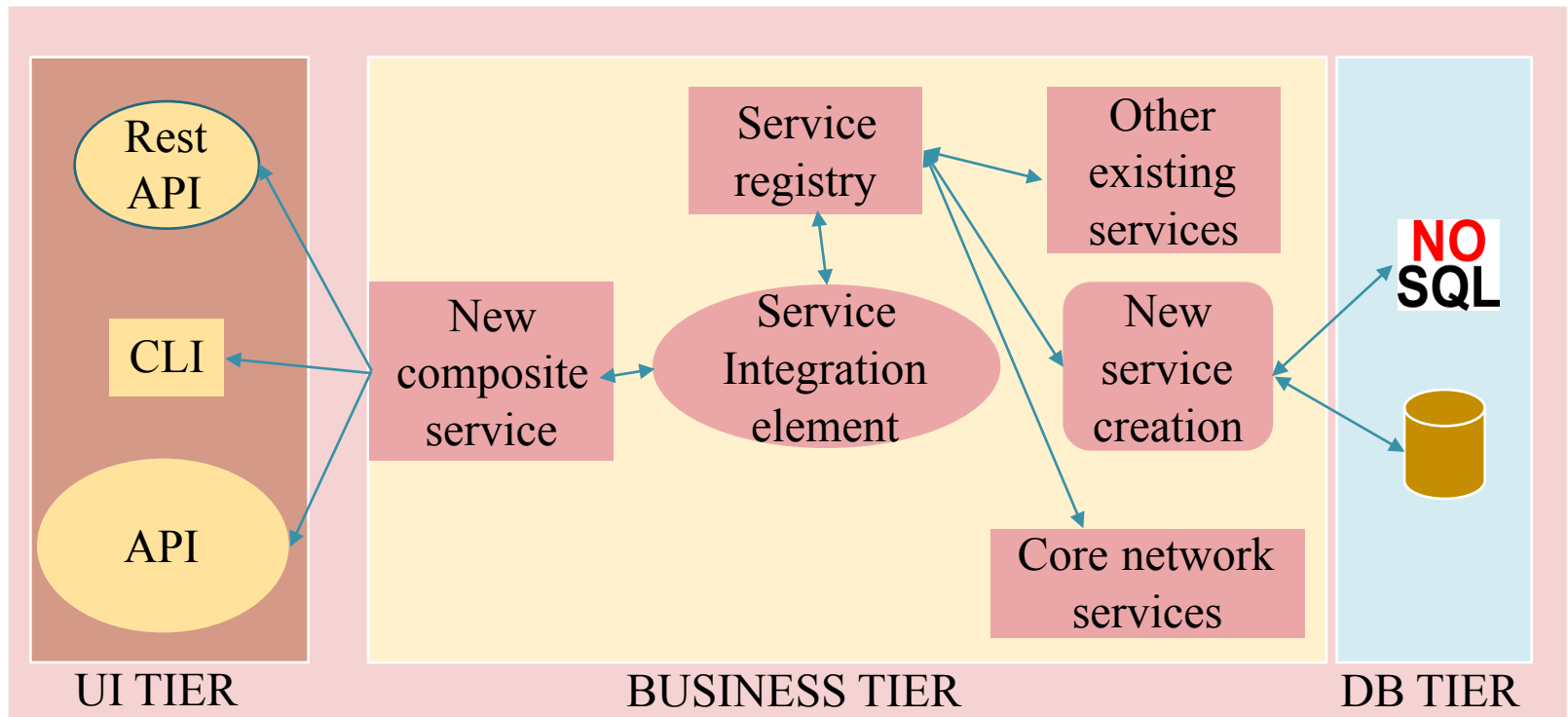
MSA is used to promote the service composition for the intent realisation

Domain driven design



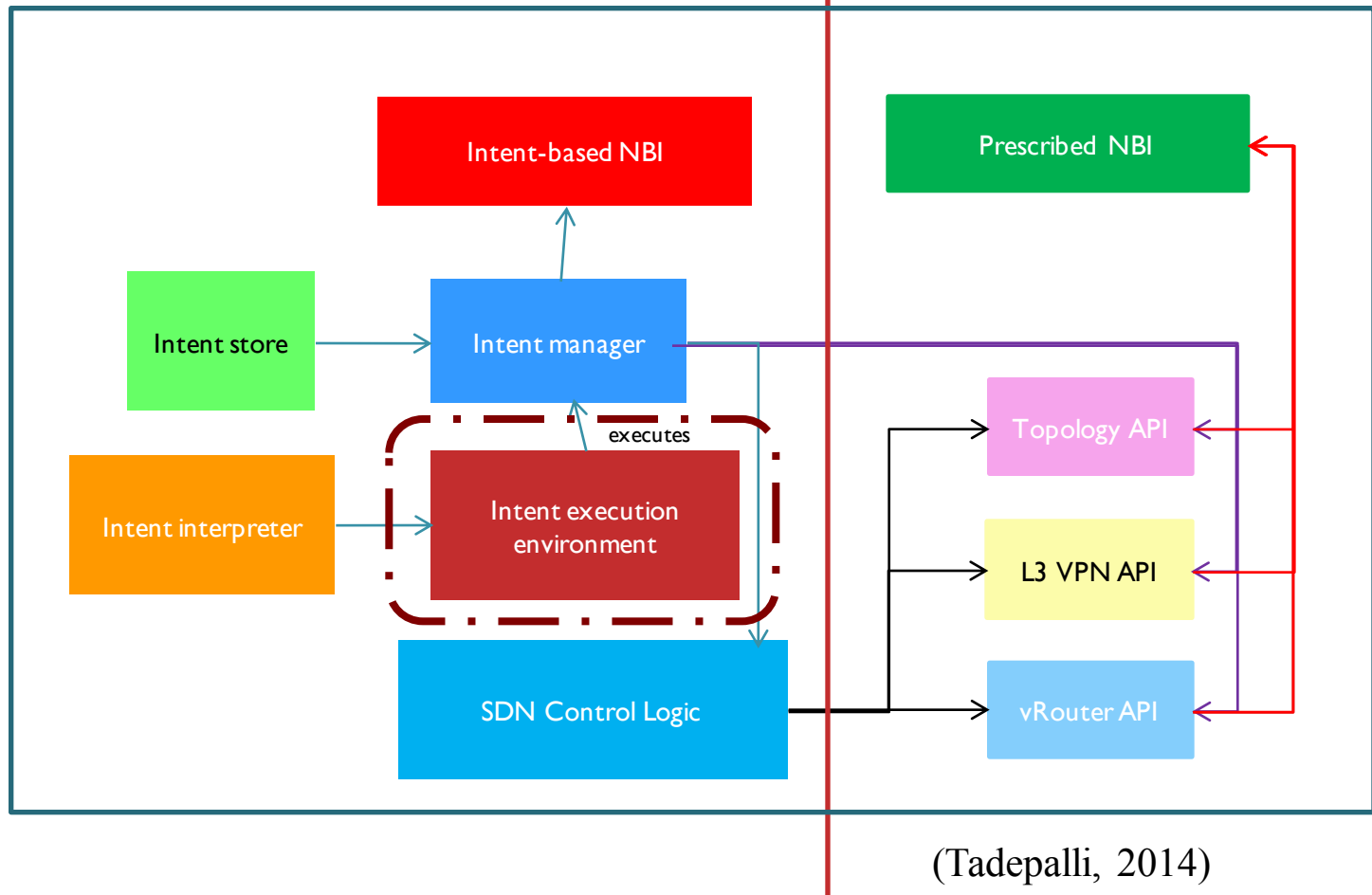
- Promoting modular design and divide-and-conquer solution approach
- Working well with the Micro-service architecture

Three-tier application architecture



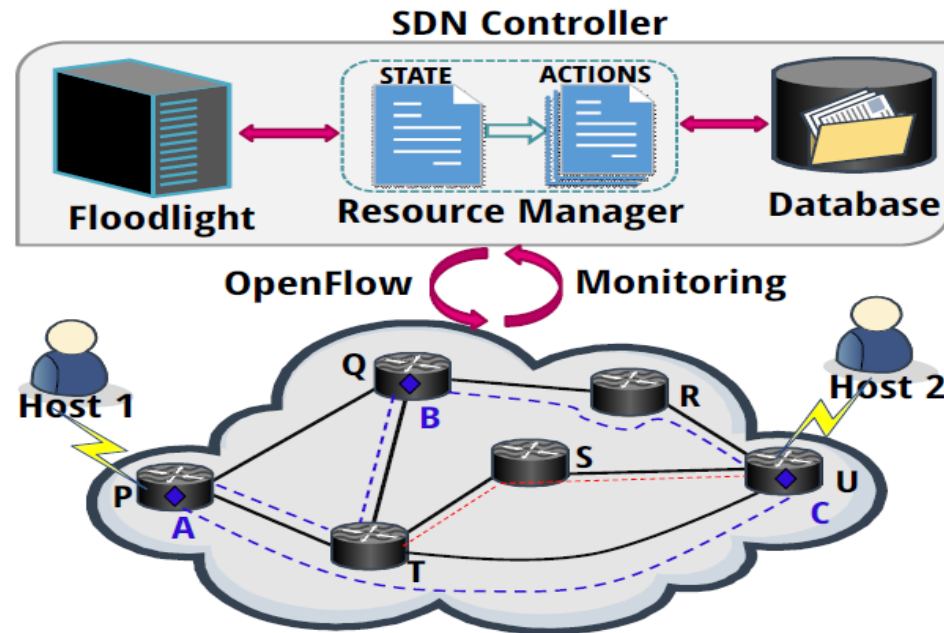
- Three-tier application architecture satisfies the requirements and serves well in the development of commercial applications
- Database tier persists states
- Business tier handles business logic via creation of new services, reuse existing services and service composition
- Presentation tier to interact with users: CLI, REST API, API

Virtualizer



- Service composition and intent realization in the proposed architecture are parts of the intent execution environment of the virtualizer of NBI

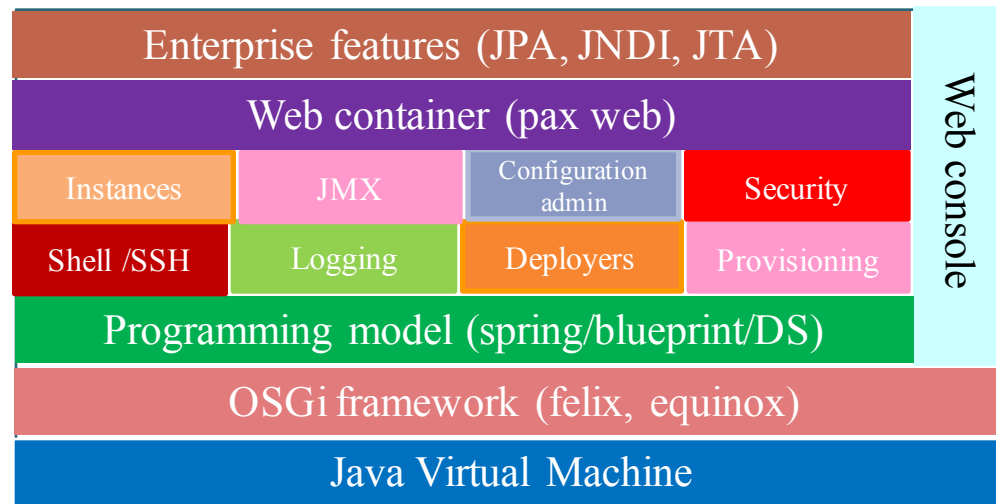
Dynamic Resource Management (DRM) Application



(Mijumbi, 2014)

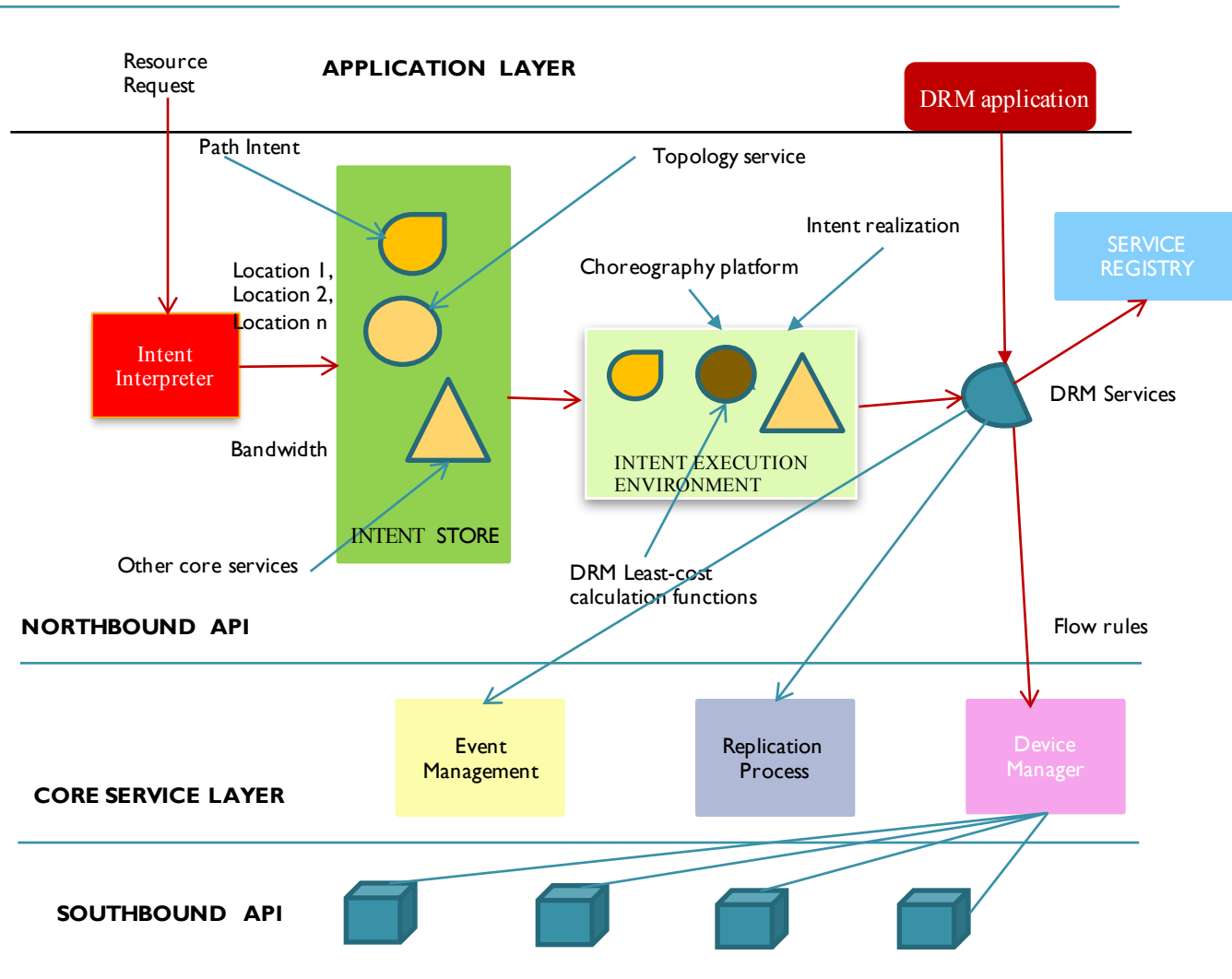
- DRM proposed a solution for network virtualisation in an efficient resource management
- In virtual network creation, the least cost path is selected based on the least ratio of usage resource over available resource for switches and links

Apache Karaf container architecture



(Apache Karaf 2015)

The visualisation of the NBI Intent-based realisation



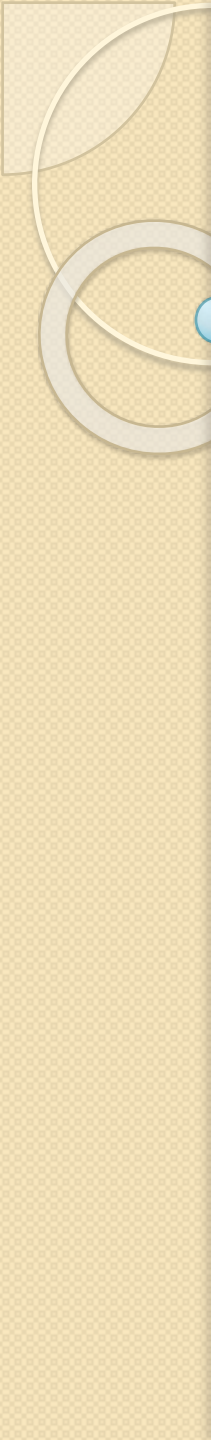
Prototype results

Table 1: Test results when running DRM on ONOS

<i>Test case 1 details</i>	Source: switch 09, Destination: switch 04, Bandwidth: 50
<i>Resource setup</i>	ONOS returns two paths between 09–04 Path 1: 09-08-04: average usage / availability ratio: 102/600, Path 2: 09-29-04: average usage / availability ratio: 102/270
<i>Expected result</i>	Path 1 with the least average ratio
<i>Actual result</i>	An intent was created for Path 1
<i>Test case 2 details</i>	Source: switch 00, Destination: switch 31, Bandwidth: 50
<i>Resource setup</i>	ONOS returns two paths between 00–31 Path 1: 00-04-31: average usage / availability ratio: 102/200 Path 2: 00-02-31: average usage / availability ratio: 102/600
<i>Expected result</i>	Path 2 with the least average ratio
<i>Actual result</i>	An intent was created for Path 2
<i>Test case 3 details</i>	Source: switch 04, Destination: switch 12, Bandwidth: 50
<i>Resource setup</i>	ONOS returns one path between 04–12 Path 1: 04-29-15-12, all links are ≥ 100
<i>Expected result</i>	Path 1 should be returned
<i>Actual result</i>	An intent was created for Path 1

Conclusion & Discussion

- The intent-based NBI is needed to develop extended, business-like network application
- Service composition, the creation of new services and reuse existing services are the main requirements
- The proposed architecture is based on micro-service principles, three-tier application architecture and domain driven design
- The architecture avoids the ad-hoc and self-explored way and promotes a systematic approach so developers can concentrate on the business requirement



QUESTIONS

THANK YOU