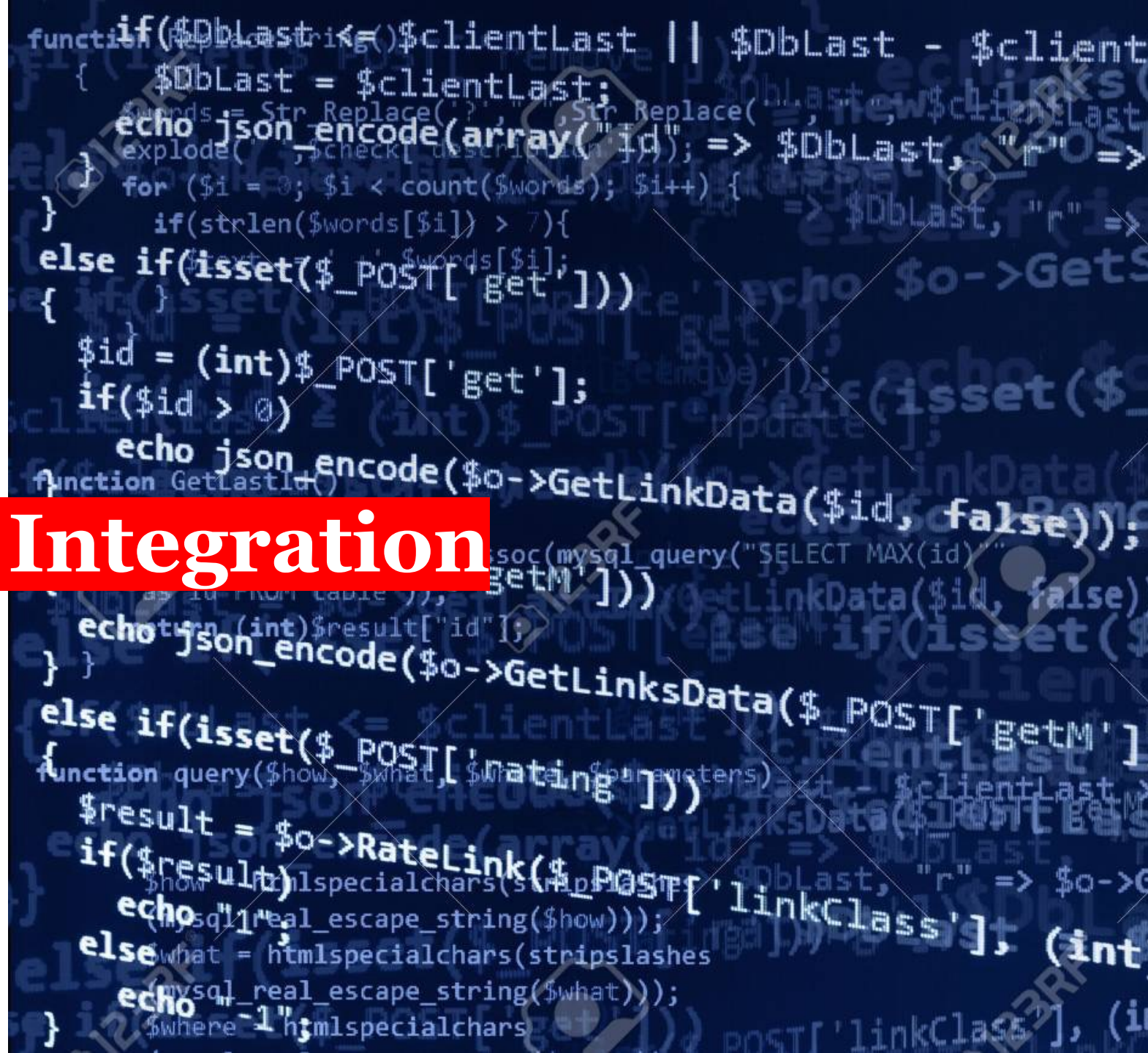
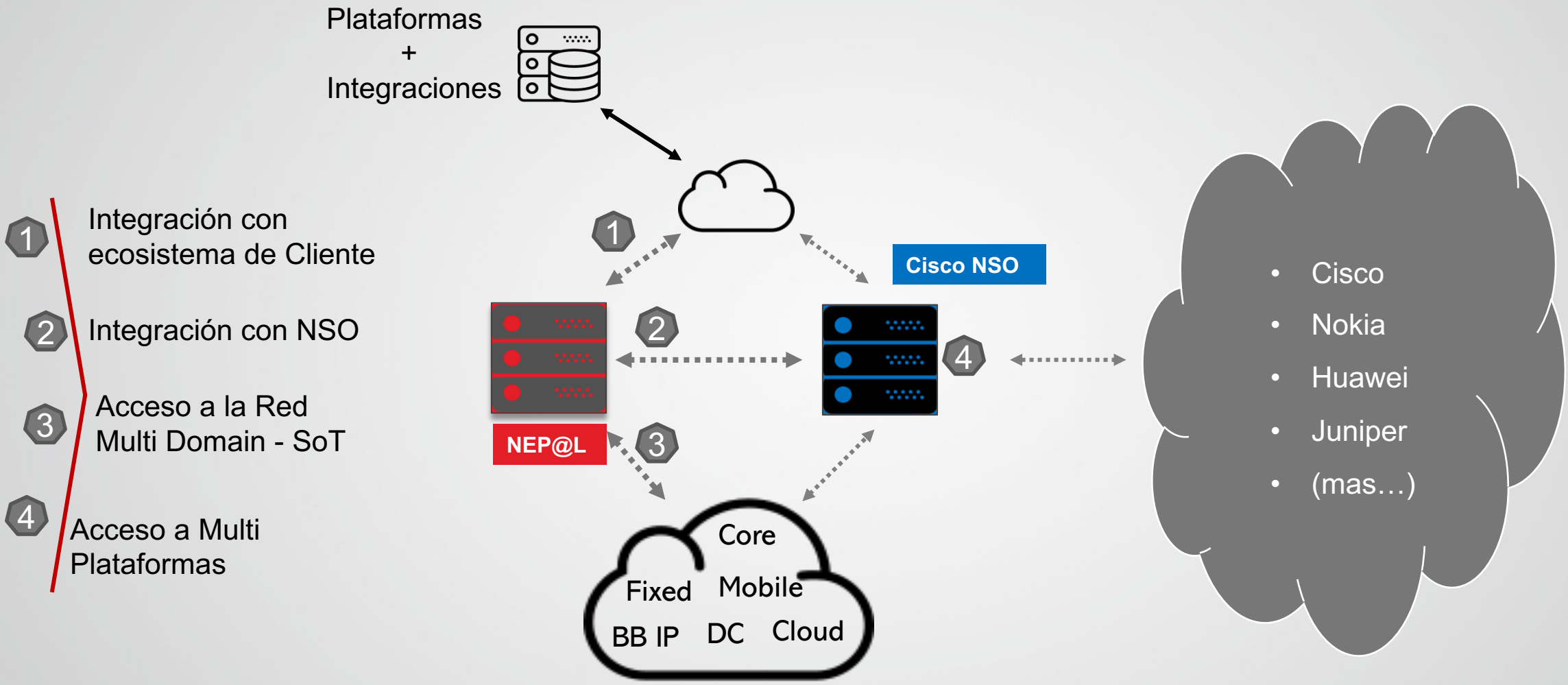
LOGICALIS

NEP@L
Cisco NSO Integration



Edgardo Scrimaglia
NEP@L mentor

NEP@L and NSO architecture



LOGICALIS
Business and technology working as one

Modeling the integration – YANG & YAML

```
namespace "http://example.com/mgmt-access-template";
  prefix mgmt-access-template;
    import ietf-inet-types {
  prefix inet;
  }
  .
  .
  import mgmt-access-acl {
    prefix acl;
  }
  description
    "Bla bla...";
  revision 2016-01-01 {
    description
      "Initial revision.";
  }
  augment /ncs:services {
    list mgmt-access-template {
      description "This is an RFS skeleton service";
      key name;

      leaf name {
        tailf:info "Unique service id";
        type string;
      }
      uses ncs:service-data;
      ncs:servicepoint mgmt-access-template-servicepoint;

      leaf snmp-acl {
        mandatory true;
          type leafref {
          path "/ncs:services/acl:mgmt-access-acl/acl:name";
        }
      }
    }
    .
    .
}
```

NSO

From the abstraction

to an instance

```
Template:
  names:
    - name: nepal-template
      snmp_acl: nepal-acl2
      vty_acl: nepal-acl1
      snmp_community:
        - community1
        - community2
ACL:
  names:
    - name: nepal-acl1
      hosts:
        - 10.10.10.10
        - 10.10.10.20
      net:
        - ip: 172.16.1.0
          mask: 0.0.0.255
        - ip: 172.16.2.0
          mask: 0.0.0.255
    - name: nepal-acl2
      hosts:
        - 10.10.10.10
        - 10.10.10.20
        - 10.10.10.30
      net:
        - ip: 172.16.1.0
          mask: 0.0.0.255
        - ip: 172.16.3.0
          mask: 0.0.0.255
```

NEP@L

Coding the integration – Python on NSO

```python
import requests

# credentials
API_USER = 'admin'
API_PASS = 'admin'

# nso server address
API_BASE = 'http://nso:8080'

# api headers
API_HEAD = {
  'Accept': 'application/vnd.yang.data+xml'
}

api_session = requests.Session()
api_session.auth = (API_USER, API_PASS)

# create nso device from csr1.xml
api_endpoint = f'{API_BASE}/api/running/devices/device/csr1'
with open('nso_template_new_device.xml') as xml:
   api_response = api_session.put(api_endpoint, headers=API_HEAD, data=xml)
print(f'-> PUT: {api_endpoint}')
print(f' -> RESPONSE: {api_response.status_code}')
.
.
.
```

Procedural/OOP

- **What to do**
- **How to do**

LOGICALIS
Business and technology working as one

Coding the integration – Ansible and Python on NEPEL

```
# Create ACL SERVICE on NSO.
# By Ed Scrimaglia
---
- name:  "*** Config Service ACL on NSO ***"
    hosts: localhost
    gather_facts: no

vars_files:
    - ../Service_Model/Service_Model.yaml

tasks:
    - name:  include var file
      include_vars:
            file: nso_vars.yaml
            name: nso

    - name: Config ACLs
      nso_config:
            url: "{{nso.nso_url}}"
            username: "{{nso.username}}"
            password: "{{nso.password}}"
             data:
                 tailf-ncs:services/mgmt-access-acl:
                       - name: "{{item.name}}"
                         host:
                             "{{item.hosts}}"
                         net:
                            "{{item.net}}"
                         __state: "{{state}}"
        when:  state == "present"
        loop:
            "{{var_service_model.ACL.names}}"
        register:  salida
```

Declarative

- **What to do**

LOGICALIS
Business and technology working as one

NEP@L gives CI/CD to the integration

Pipeline NSO_SERVICE_ACL

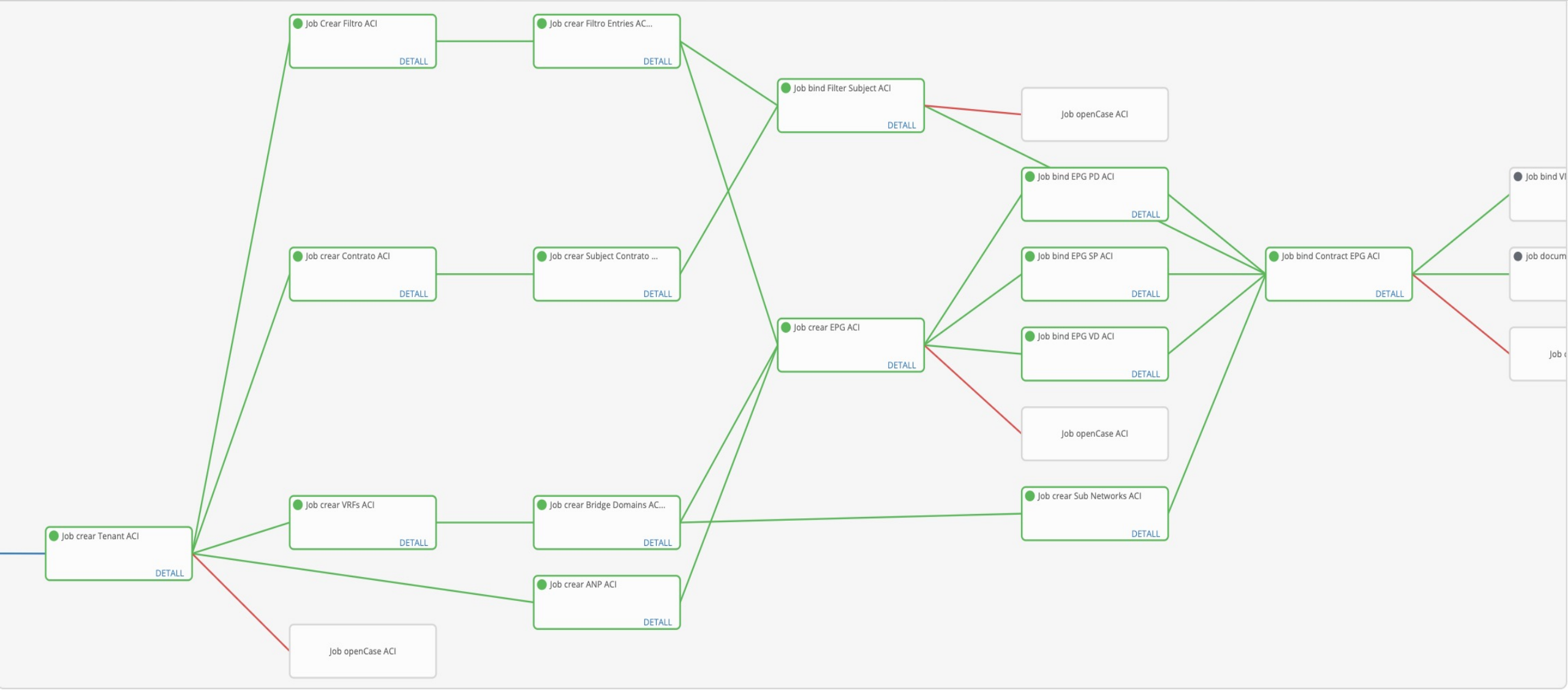
NEP@L CI/CD:NSO-Service-ACL configuration

Recent Changes

Stage View

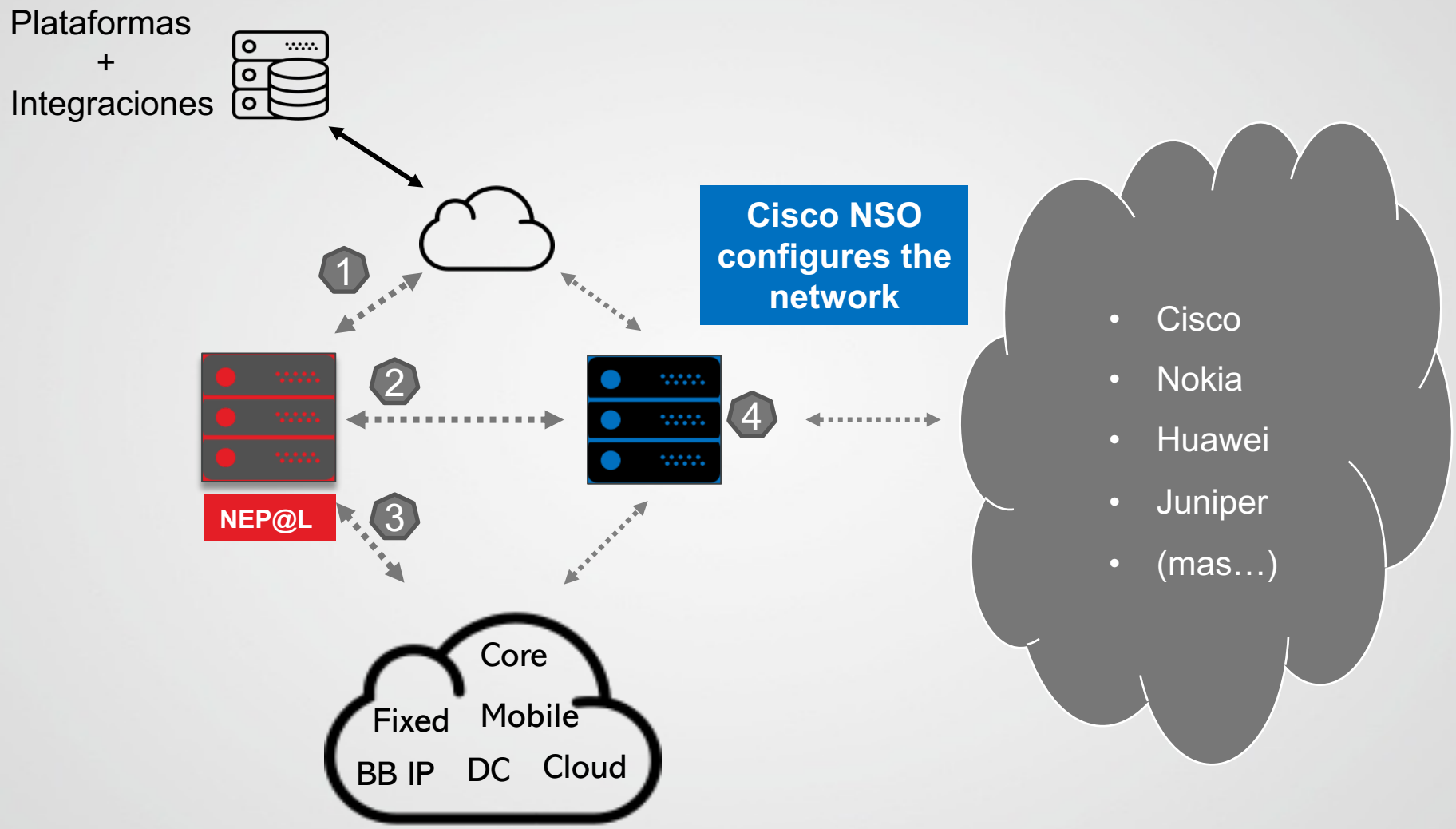
	Declarative: Checkout SCM	Code Review	Configuring Devices	Configuring ACLs	Configuring Templates	Testing	NSO Deployment	Delivery	Declarative: Post Actions
Average stage times: (Average full run time: ~4min 6s)	2s	620ms	48s	5s	4s	4s	33s	19s	39s
#211 Jul 03 16:12 — 1 commit	2s	629ms	58s	6s	5s	6s	58s	34s	1min 11s
#210 Jul 03 16:10 — 1 commit	2s	476ms failed	40ms failed	37ms failed	38ms failed	36ms failed	46ms failed	37ms failed	88ms

NEPEL gives a workflow manager to the integration



LOGICALIS
Business and technology working as one

NSO is the Configuration Manager, NEP@L does Workflow and CI/CD



LOGICALIS
Business and technology working as one

LOGICALIS

Thanks

NEP@L Framework