

Intent NBI

Adolfo Perez-Duran

CTO, CyRetix

Chair ONF Project Boulder (Intent NBI)

ODL Contributor

adolfo@cyretix.com

[linkedin.com/in/aperezduran](https://www.linkedin.com/in/aperezduran)





Attribution: <https://bookstobrews.files.wordpress.com/2015/08/2e01e9640e3d944e43c1e62d0218d6f5.jpg>



Don't Tell Me What to Do,

Tell Me What You Want.



Context

Intent REQUEST

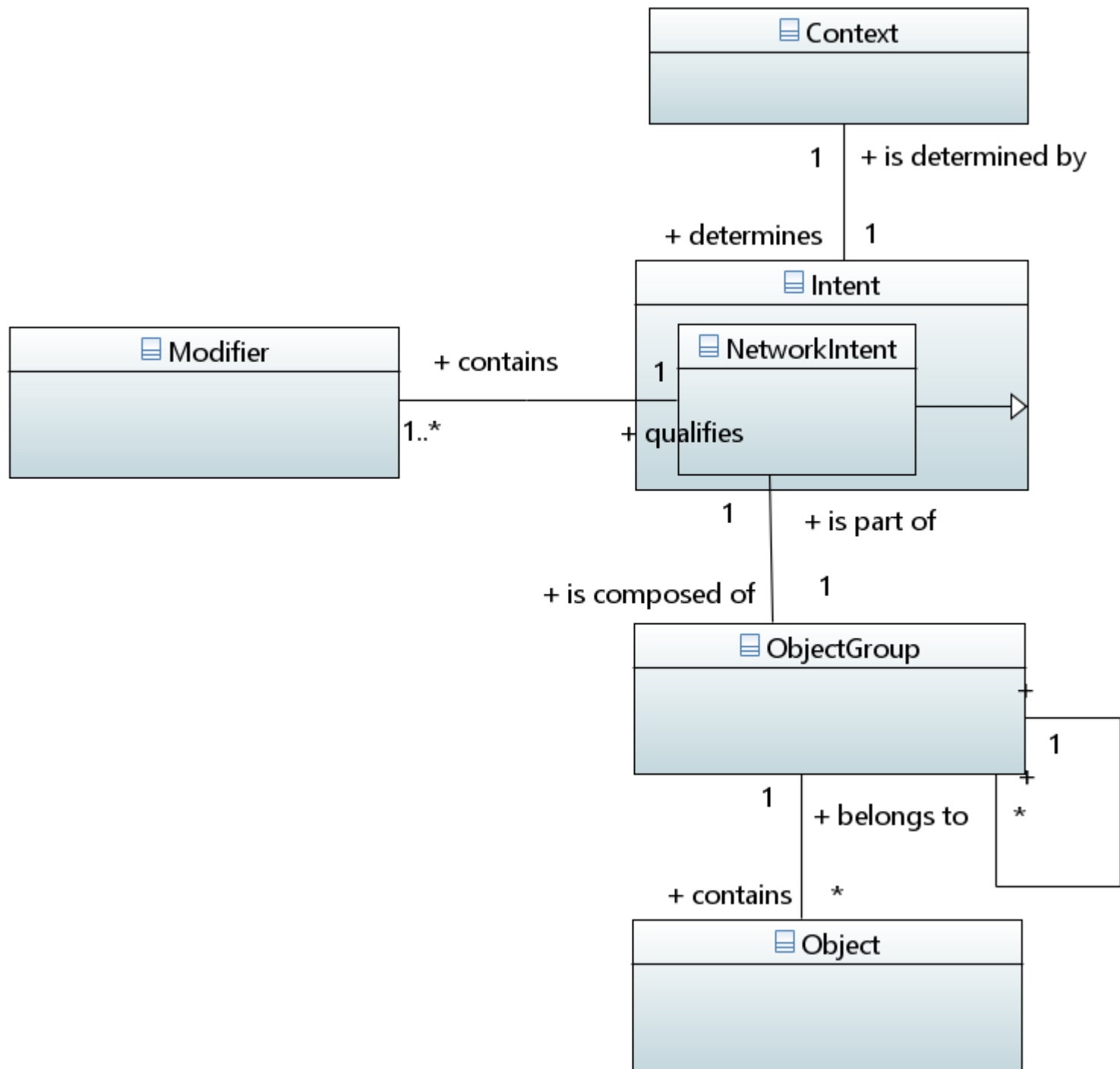
Context=Food Replication

“Tea, earl gray, hot”

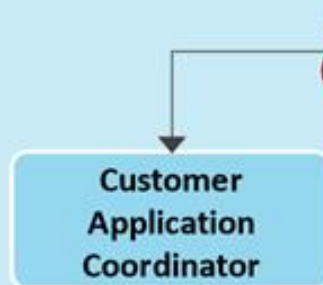
Object: Tea

Modifiers: = earl gray, hot

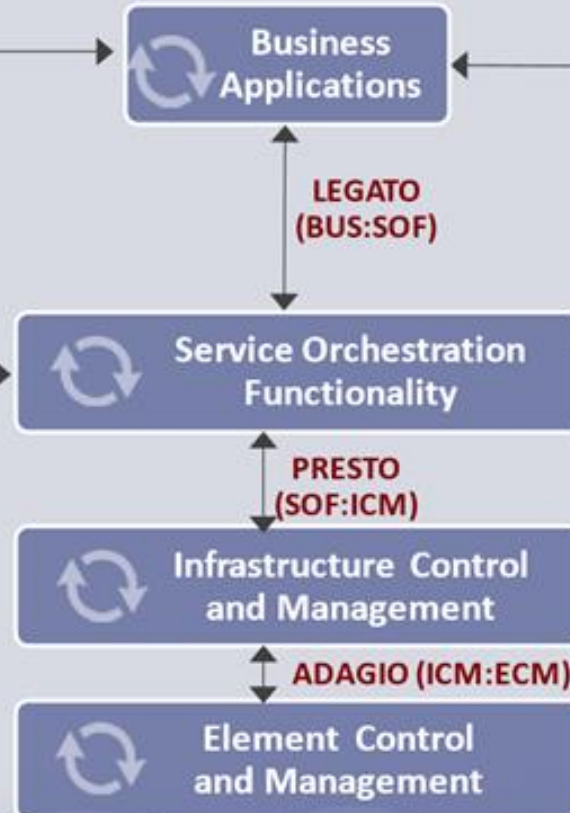
Domain := Networking
Context := Network Services



Customer Domain

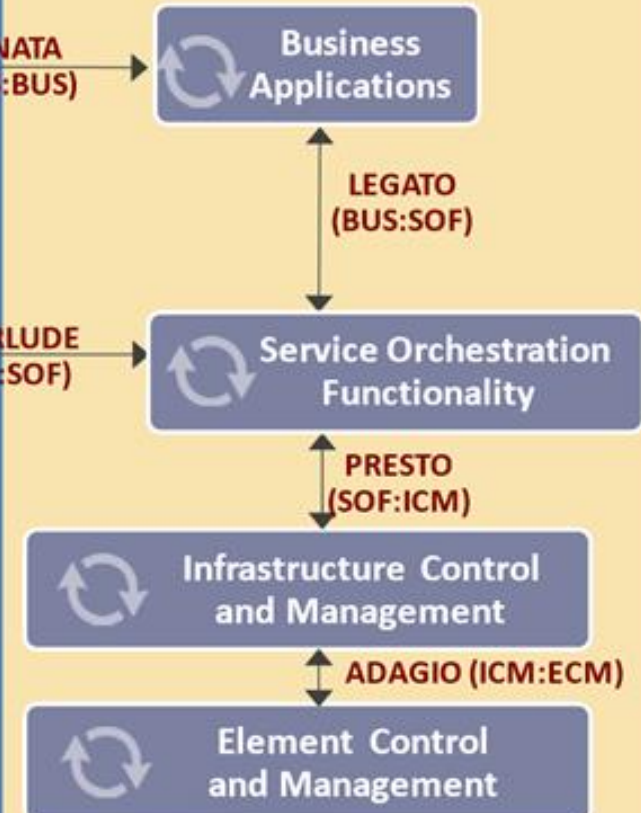
CANTATA
(CUS:BUS)ALLEGRO
(CUS:SOF)

SP Domain

LEGATO
(BUS:SOF)PRESTO
(SOF:ICM)

ADAGIO (ICM:ECM)

Partner Domain

SONATA
(BUS:BUS)INTERLUDE
(SOF:SOF)LEGATO
(BUS:SOF)PRESTO
(SOF:ICM)

ADAGIO (ICM:ECM)

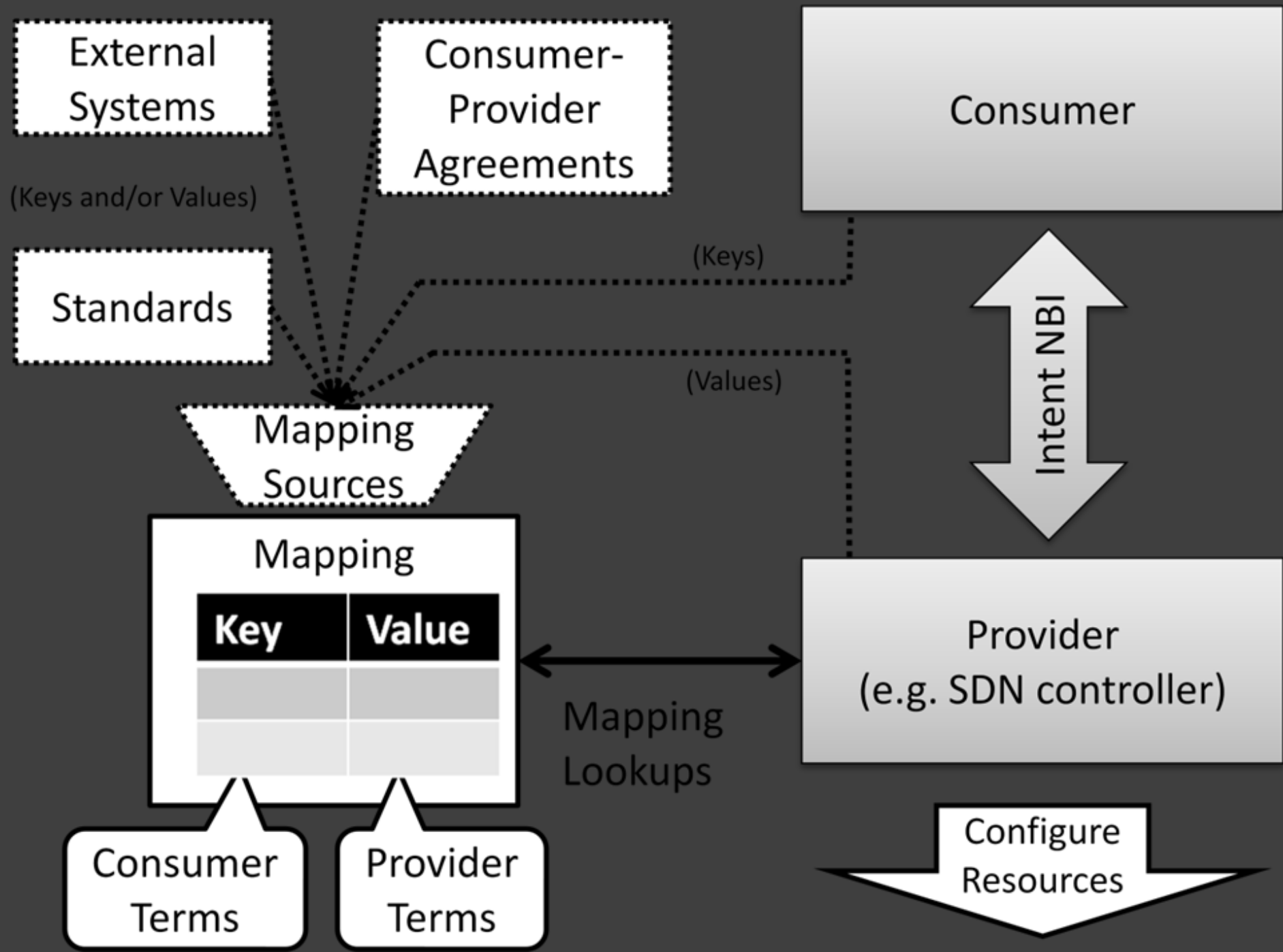
Network Infrastructure

Context=Network Services

Domain := Networking

Abstraction Level (Layer) := LSO Legato Interface Point

Expected Outcome/Capability := Point to point connectivity



Intent REQUEST

Context:= Point-to-Point Connectivity

Connect endpoint a and endpoint z with a bandwidth of 100 Mbps, low latency and high availability.

Intent REQUEST

Context:= Point-to-Point Connectivity

Connect building A and building Z with a bandwidth of 100 Mbps, low latency and high availability.

Objects: Connection (Apparent adjacency, Adjacency effect)

Object Groups: (Building A, Building Z)

Modifiers: Bandwidth = 100 Mbps, low latency, high availability

Intent REQUEST

Context:= Point-to-Point Connectivity

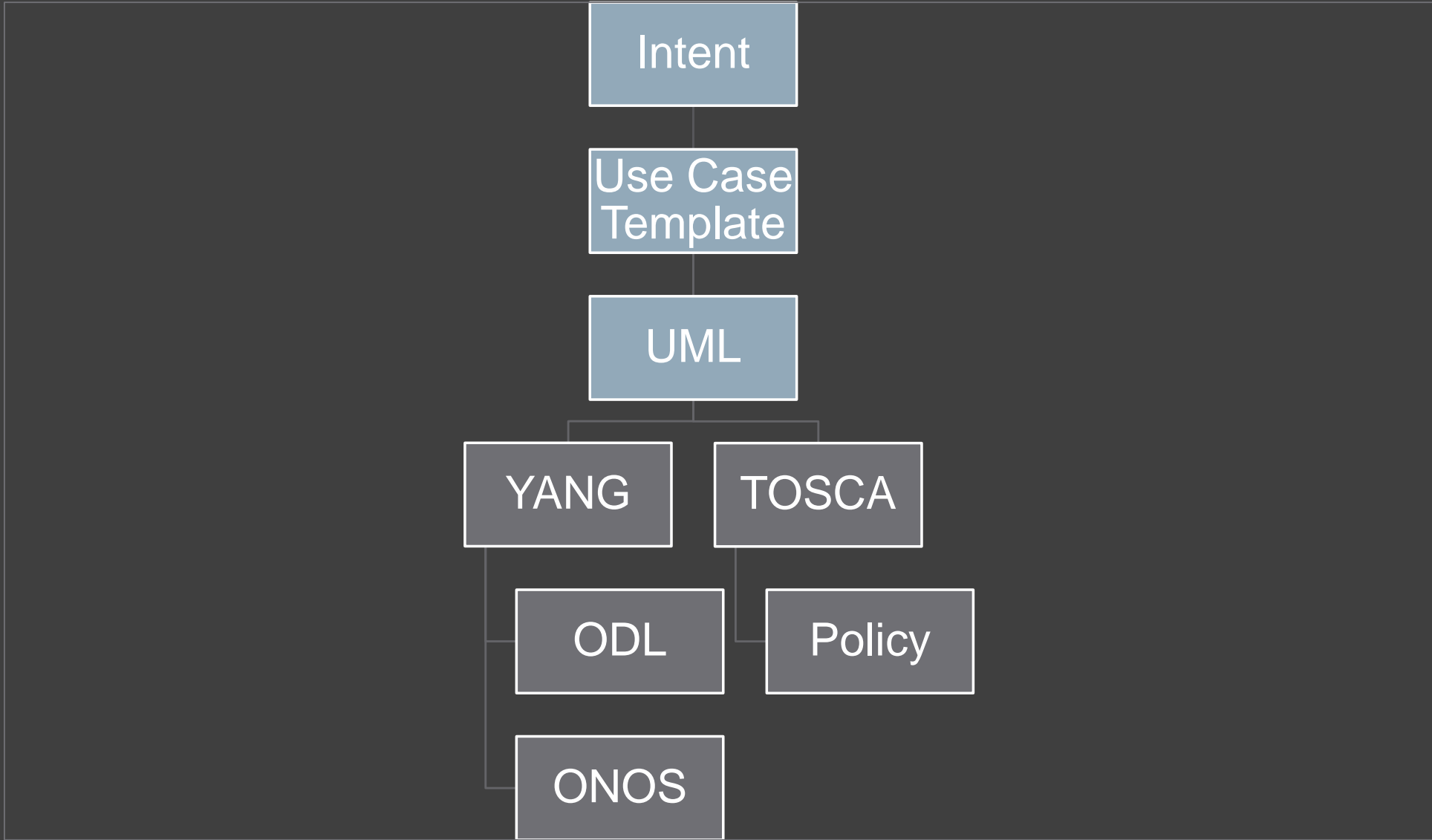
Connect building A and building Z with a bandwidth of 100 Mbps, low latency and high availability.

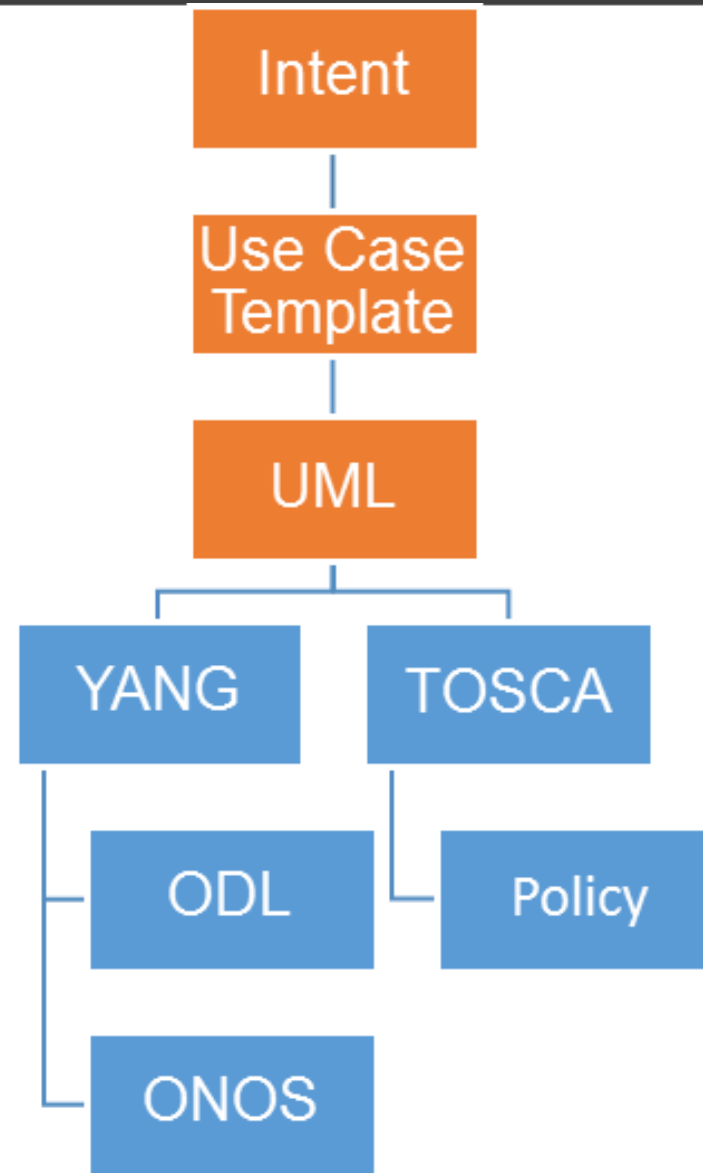
Objects: Connection (Apparent adjacency, Adjacency effect)

Object Groups:

Modifiers: Bandwidth = 100 Mbps, low latency, high availability,
Location A, Location Z

Mapping Intent







Network Intent Composition

Applications describe WHAT is desired, not HOW to provide services.

Controller manages network based on “Intent” (e.g. behaviors and policies)



Provides general and abstract policy semantics instead of Openflow-like flow rules.

Uses existing OpenDaylight Network Service Functions and Southbound Plugins to control both virtual and physical network devices.



1. Create a basic topology using mininet

```
sudo mn --topo single --mac --controller=remote,ip=127.0.0.1,port=6633 --switch ovsk,protocols=OpenFlow13
```

2. Run karaf distribution and install odl-nic-core-mdsal, odl-nic-console, odl-nic-listeners

```
karaf>feature:install odl-nic-core-mdsal odl-nic-console odl-nic-listeners
```

3. Create an Intent qos:

```
intent:qosConfig -p High_Quality -d 46
```

Note: The DSCP value ranges from 0-63

4. Create an Intent to allow traffic between two devices applying the QoS service

```
karaf>intent:add -a ALLOW -t 00:00:00:00:00:01 -f 00:00:00:00:00:02 -q QOS -p High_Quality
```

5. Verify if a new rule was created on OF switch with a field 'mod_nw_tos:184'

Expressing QoS Intent



THANK YOU

