

# **Building Cyber-Physical Sensing Infrastructure based on Service-Controlled Networking**

**Takashi Kimata, Koji Zettsu.**  
**National Institute of  
Information and Communications Technology**



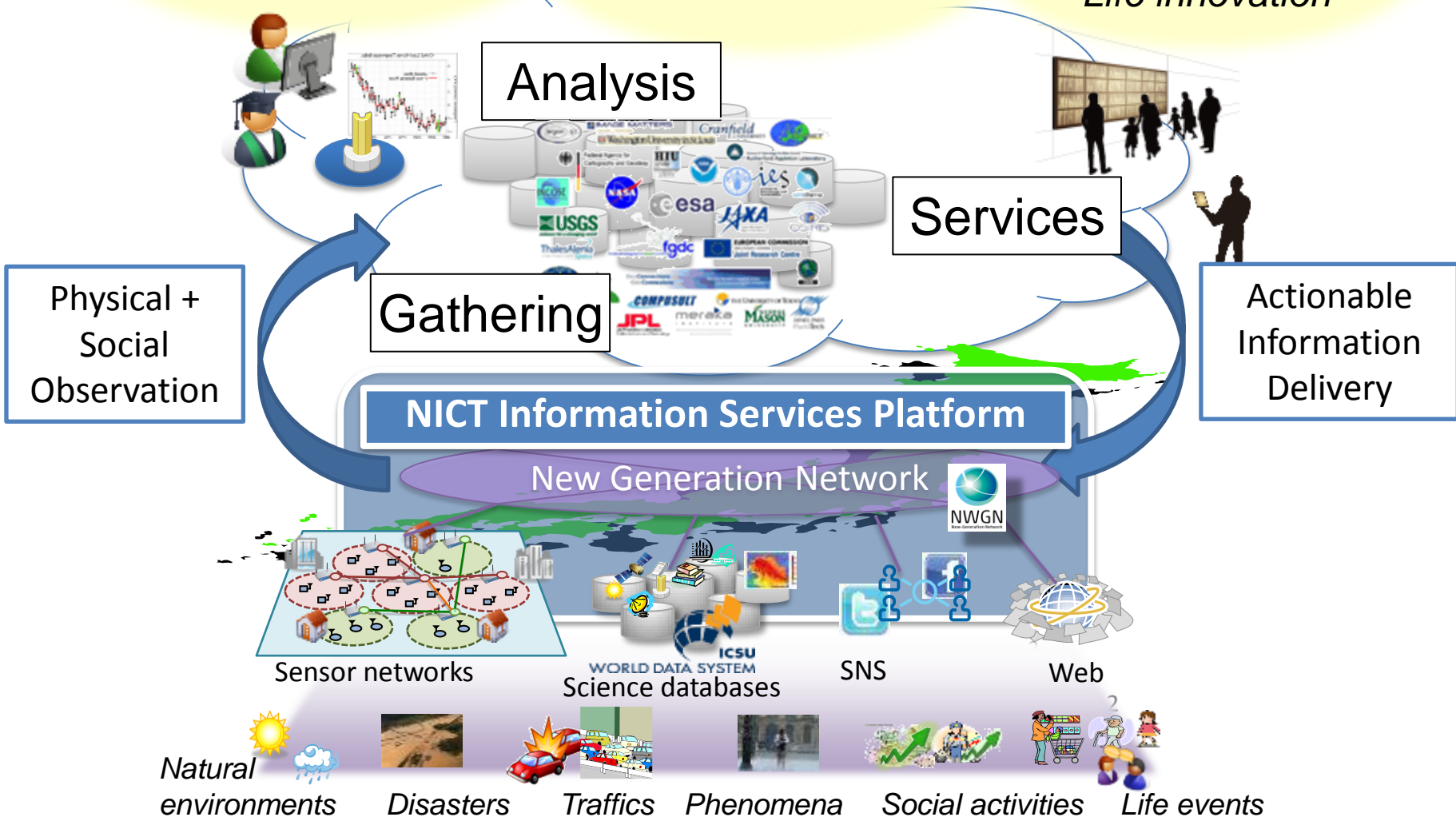
Asia-Pacific Advanced Network 35<sup>th</sup> Meeting  
January 14<sup>th</sup>, 2013

# ICT Platform Knowledge Language Grid

*Environmental application science*

*Disaster response*

*Smarter society & Life innovation*



# SCN Demo video: Geo-Social Sensing(GSS) App



The screenshot displays the Geo-Social Sensing (GSS) App interface, divided into two main sections: Atomic Sensor Services and Virtual Sensor Monitor.

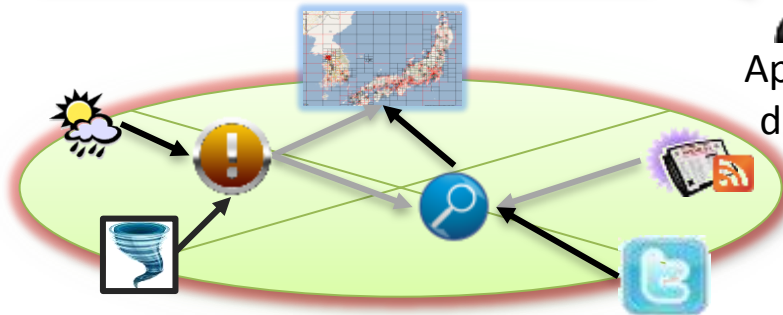
**Atomic Sensor Services:**

- Social Sensor Services:**
  - Twitter: 15,704,207
  - RSS - News: 6,196,000
  - Hospitals: 106,400
  - ConventionFacility: 46,700
  - MajorLandmarks: 6,848,000
  - Railways: 65,000
- Natural Sensor Services:**
  - Weather - Rain: 16,716,004
  - Weather - Temperature: 6,848,000
  - Weather - Wind-Speed: 6,316,000
  - Weather - Snow: 6,174
  - Pollen: 7,620
  - Typhoon

**Virtual Sensor Monitor:**

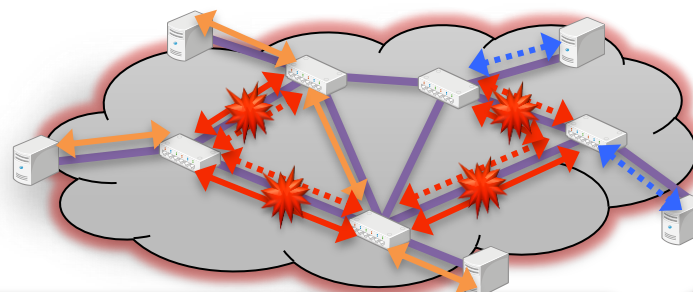
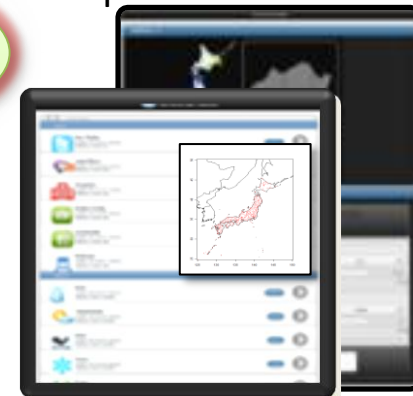
- Virtual Sensor Description:
  - Interval (seconds): 15
  - apply all condition: On
  - Sensor Service: Choose one...
  - Monitoring Target: Choose one...
  - Monitoring Condition: Choose one...
  - Apply

Agile development of sensor apps regardless of network capability



Application developer

Increasing gap seriously damages sustainability



Unexpected changes of networks

- Suffer from unforeseen traffics
- Hard to follow ad-hoc changes of sensor app's requirements

Network administrator

- On-demand configuration of networks in response to dynamic collaborations of sensing information services

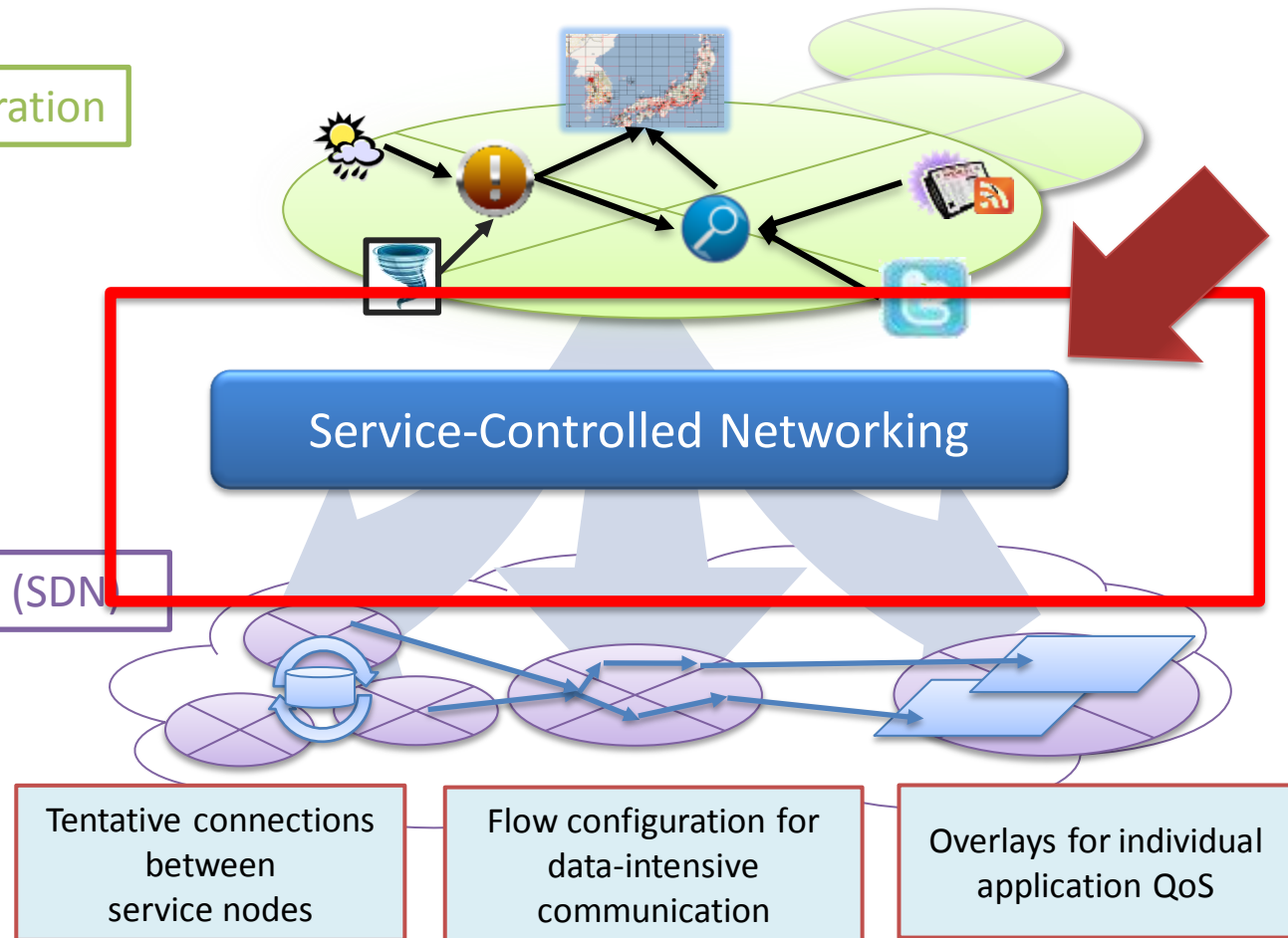
## CPSenS service collaboration

- Service discovery
- Message exchange
- Status monitoring

Middleware

## Programmable network (SDN)

- Node search
- Path configuration
- QoS overlay



# SCN Demo video: Reconfiguration of Network



Log dlog

del gsa cs  
send 1  
send 10  
send 20  
mode step  
mode timer  
mode direct  
20120607

PIRA	TwDA	TeQU	WIPR	WISE	RaPR	GeGA	RaSE	RaDA	TwPR	EaQU	TeDA
TwSE	RaPR	WlQU	RaSE	RaQU	EaDA	TePR	TeSE	TwQU	WIDA	RaDA	EaPR
EaSE	RaQU										

service traffic RaDA - RaPR: 5.493  
service traffic RaPR - RaDA: 11.216  
service traffic RaSE - EaQU: 4.395  
service traffic EaQU - EaSE: 9.109  
service traffic RaQU - RaPR: 4.12  
service traffic RaPR - RaQU: 8.452  
service traffic RaSE - RaQU: 4.12  
service traffic RaQU - RaSE: 8.392  
service traffic TeSE - TeQU: 2.197  
service traffic TeQU - TeSE: 4.587  
service traffic RaSE - RaQU: 3.296  
service traffic RaQU - RaSE: 6.544

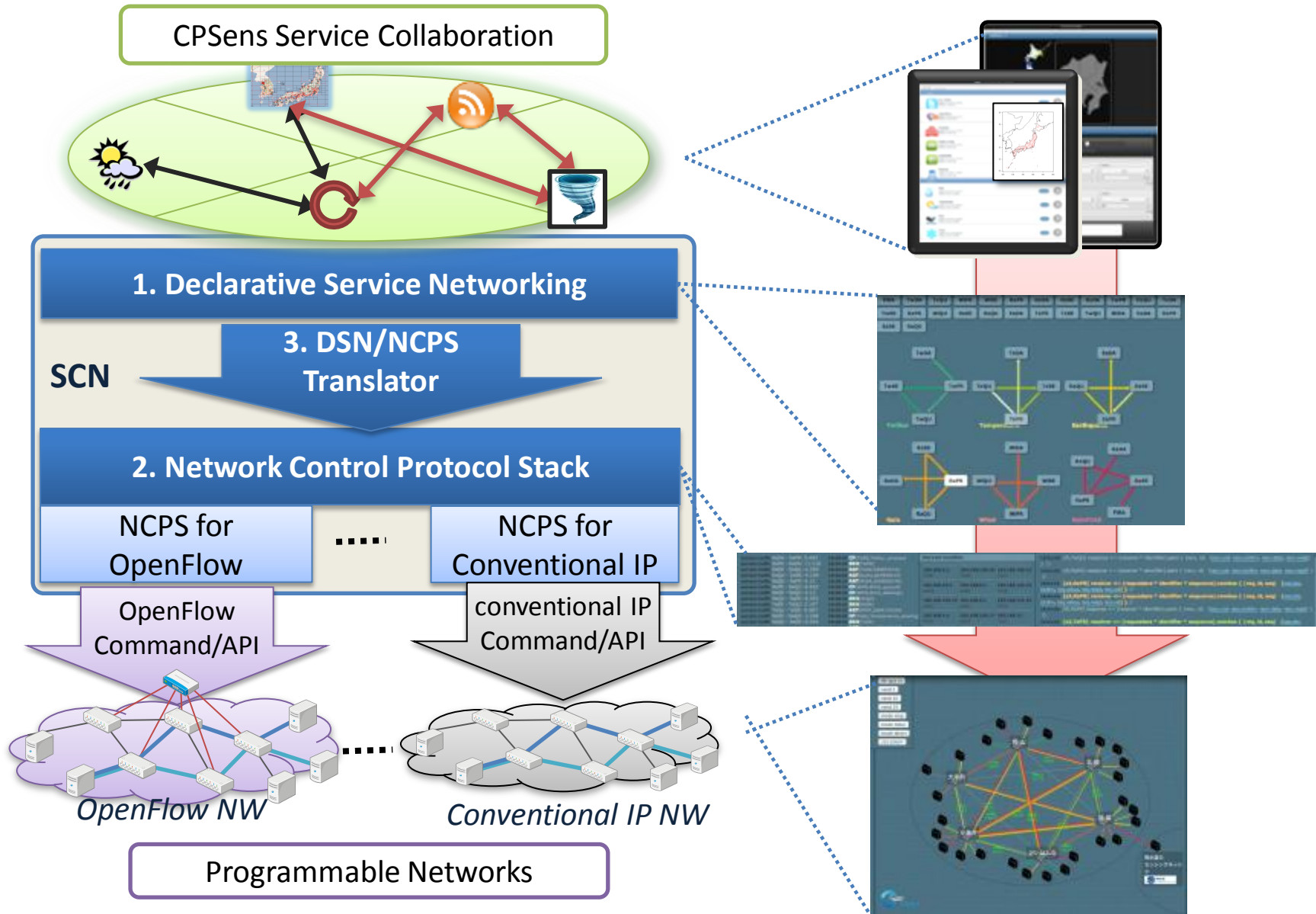
19:29:40 CP(TwSE, Twitter\_process)  
19:29:49 SRH(TwPR)  
19:29:53 RGT(TwSE,935873710)  
19:30:05 RGT(GeGA,857890414)  
19:30:32 RGT(WIDA,2055066749)  
19:30:43 CP(WIPR,Wind\_datamgt)  
19:30:42 SRH(WlQU)  
19:30:42 SRH(WIDA)  
19:30:32 RGT(WIPR,2694176154)  
19:30:05 CP(TeQU,Temperature\_sensing)  
19:28:04 SRH(TeSE)

Service location		
192.168.9.1	192.168.169.26	192.168.159.21
TwPR	TwPR	TeQU
192.168.13.1	192.168.8.1	192.168.119.15
RaDA	RaPR	TeDA
192.168.159.23	192.168.9.1	192.168.119.18
EaQU	TeSE	WIDA
192.168.4.1	192.168.138.17	192.168.3.1
WISE	EaDA	TeSE

19:41:06 [s2,TeSE] receiver <~ (requesters \* Identifier \* sequence).combos ( [req\_id, seq] [seq\_dst, id, bhv, req\_value, req\_reqid, req\_uid] ) >  
19:41:06 [r0,RaQU] response <~ (receiver \* Identifier).pairs ( [rcv\_id] [rcv\_uid, rcv\_srcbhv, rcv\_data, rcv\_reqid] ) >  
19:41:06 [s2,EaPR] receiver <~ (requesters \* Identifier \* sequence).combos ( [req\_id, seq] [seq\_dst, id, bhv, req\_value, req\_reqid, req\_uid] ) >  
19:41:06 [s2,EaPR] receiver <~ (requesters \* Identifier \* sequence).combos ( [req\_id, seq] [seq\_dst, id, bhv, req\_value, req\_reqid, req\_uid] ) >  
19:41:06 [s2,EaPR] receiver <~ (requesters \* Identifier \* sequence).combos ( [req\_id, seq] [seq\_dst, id, bhv, req\_value, req\_reqid, req\_uid] ) >  
19:41:06 [s2,EaPR] receiver <~ (requesters \* Identifier \* sequence).combos ( [req\_id, seq] [seq\_dst, id, bhv, req\_value, req\_reqid, req\_uid] ) >

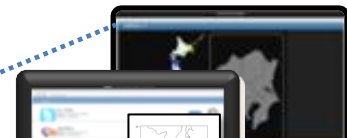
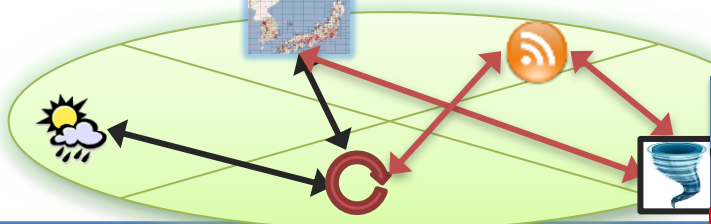
雨量のセンシングネットワーク

# SCN Architecture



# SCN Architecture

CPSens Service Collaboration



DSN

```

Overlay GeoSocialApp
// ① Registration Nodes
R1 REGIST(GeWE) <- IDENT(GeoSocialWeb, GeWE, "192.168.XXXXX")
// ② Search Nodes
F1 FIND(RaQU) <- REQUEST(GeWE, RaQU)
F2 FIND(TwQU) <- REQUEST(GeWE, TwQU)
// ③ Create Paths, Send Messages
S1 SEND(GeWE, RaQU, "50 <= Rain" & "1000 < SampleRate") <- FIND(RaQU)
S2 SEND(GeWE, TwQU, "escape" & "4000 < Record") <- FIND(TwQU)
    
```

1. Declarative Service Networking

3. DSN/NCPS Translator

SCN

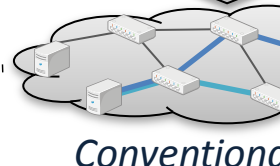
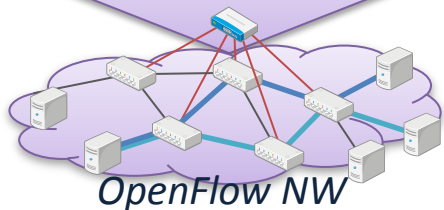
2. Network Control Protocol Stack

NCPS for OpenFlow

NCPS for Conventional

OpenFlow Command/API

conventional Command/API



OpenFlow NW

Conventional IP NW

Programmable Networks

Translation

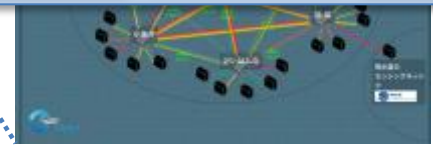
NCPS for OpenFlow

DSN Interpreter

Svc Regist Cmds Node Search Cmds Path Creation Cmds Status Cmds

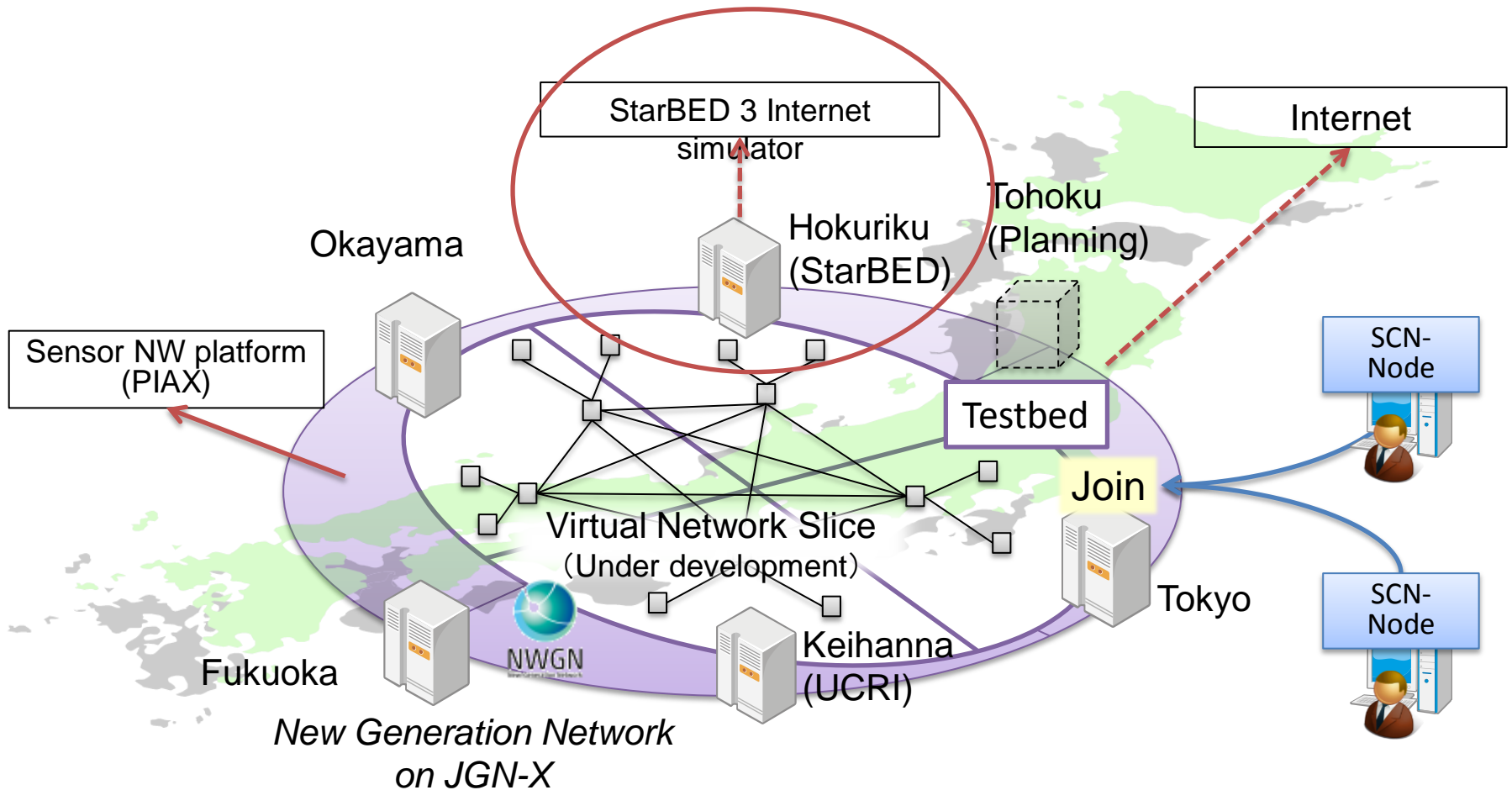
Network Manager(NwM)

Active Service Lists Controller Lists QoS Regulator FlowTable Mgmt

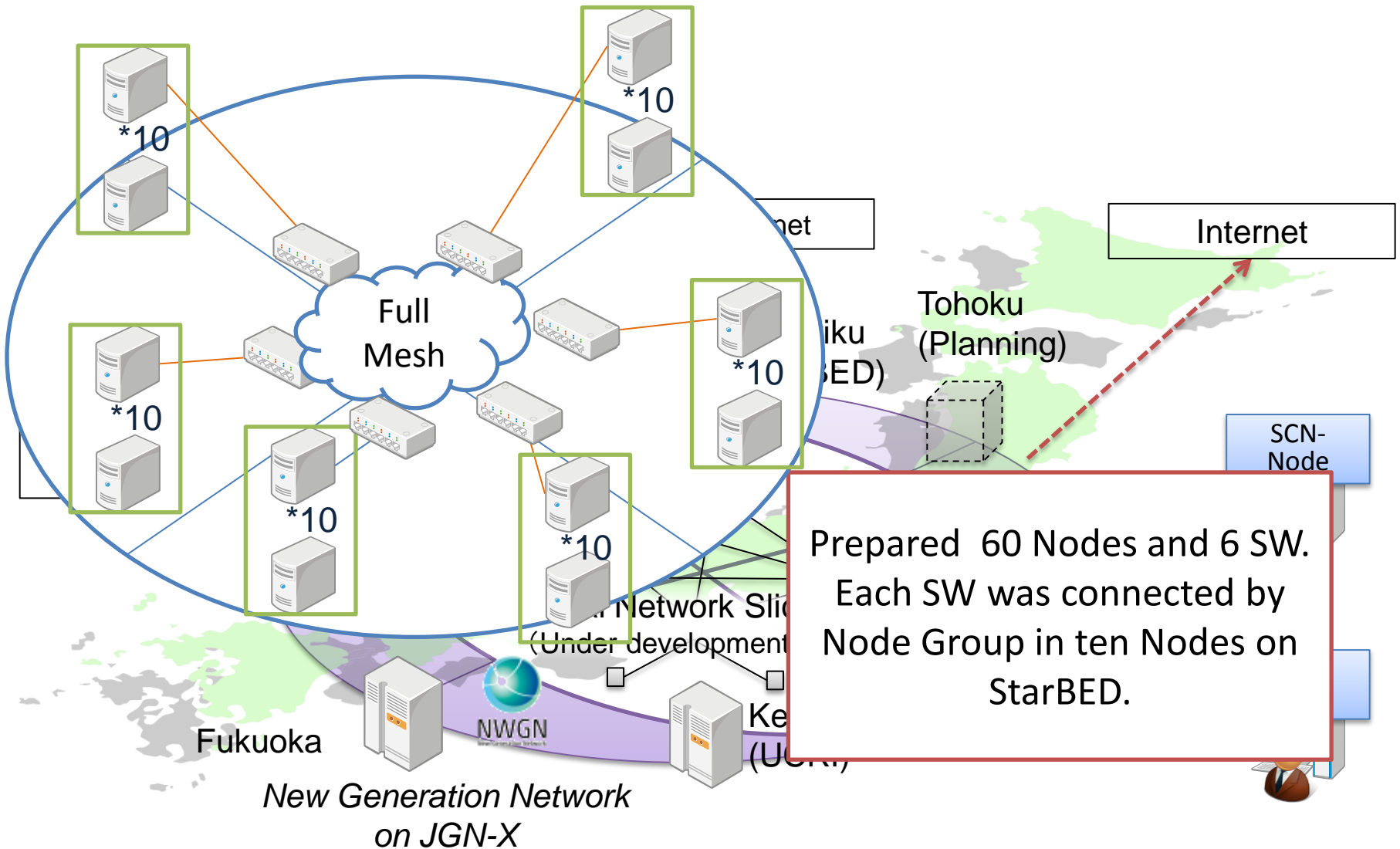


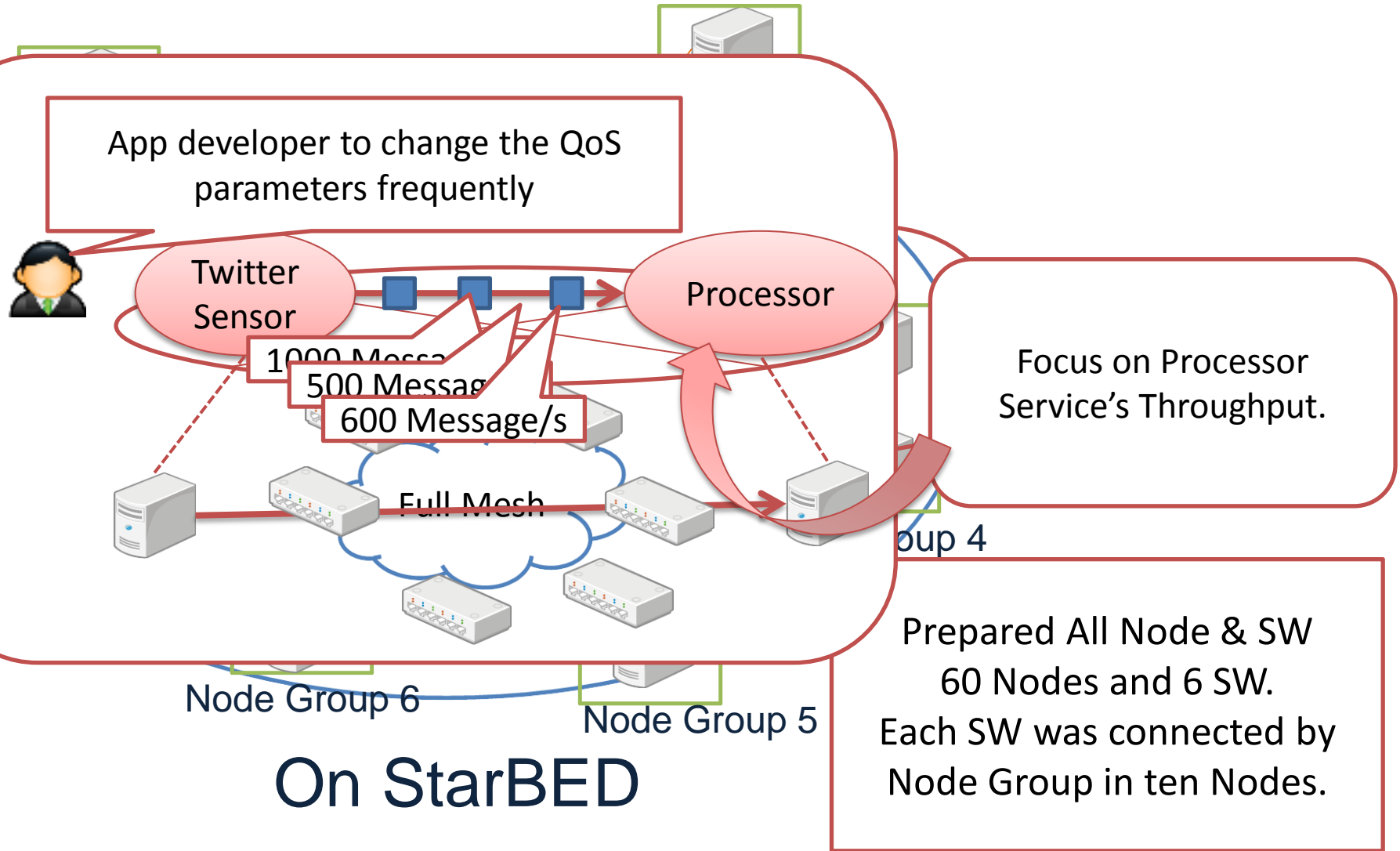


# Knowledge-Language Grid Testbed

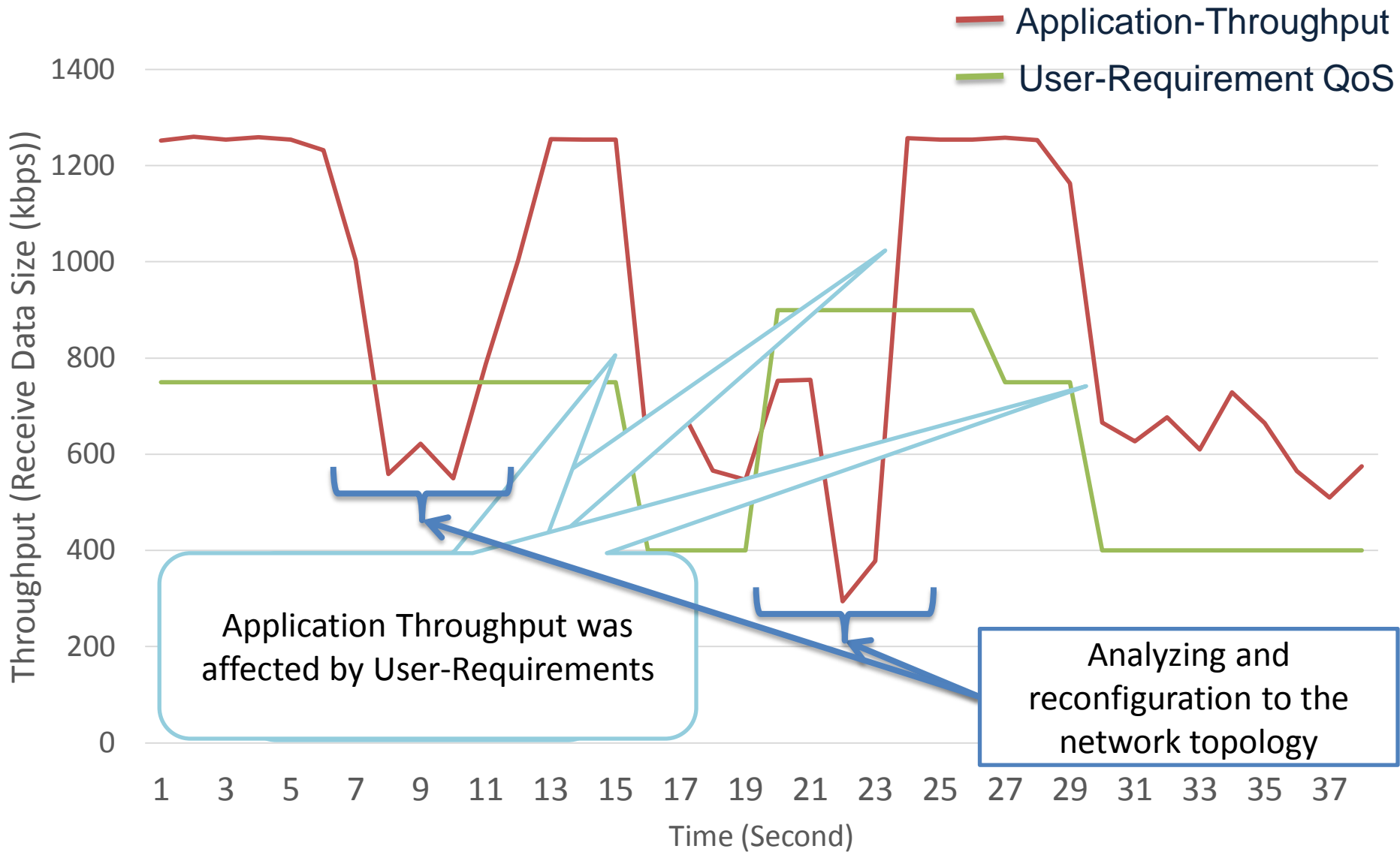


# Knowledge-Language Grid Testbed

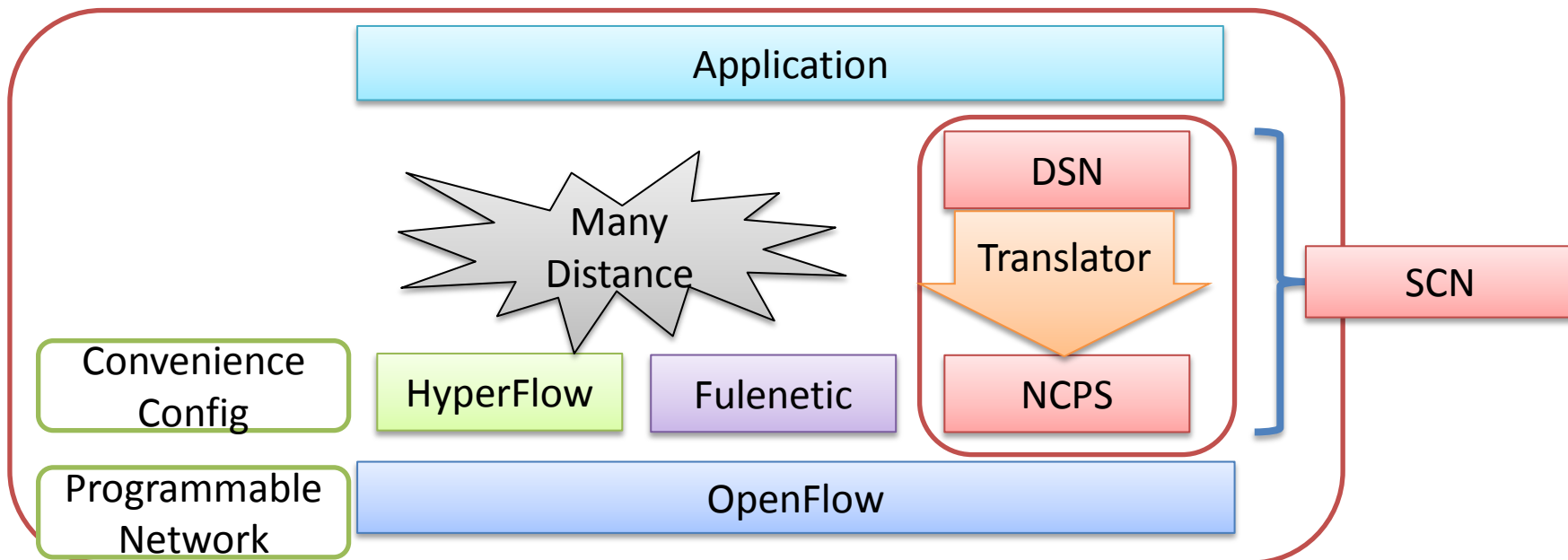




# Performance Consideration



- High-level Language of OpenFlow.
  - HyperFlow@University of Toronto
  - Frenetic@Princeton University



- Defined the description method of the application demand.  
= DSN
- Building the organization reconstruct of the networks of the application demand. = NCPS
- Building the function to convert the networks QoS from the application demand. = Translator
  
- Future Work
  - Expanding SCN to Multi-Domain Networks.
  - Expanding DSN to have detailed descriptions.
  - Building to the arbitration function for multi-user requirement

- Thank you for your attentions.