# lecture 06: centralized control —opportunities and challenges

5590: software defined networking

anduo wang, Temple University

TTLMAN 402, R 17:30-20:00

some materials in this slide are based on lectures by

Jennifer Rexford https://www.cs.princeton.edu/courses/archive/fall13/cos597E/

Nick Feamster http://noise.gatech.edu/classes/cs8803sdn/fall2014/

# NOX, Onix

# challenges

## performance

- low control-plane latency

## scalability

- large topology, huge volume of events, flow initiations

## reliability

- handle equipment (and other) failover gracefully

# opportunities

simplicity

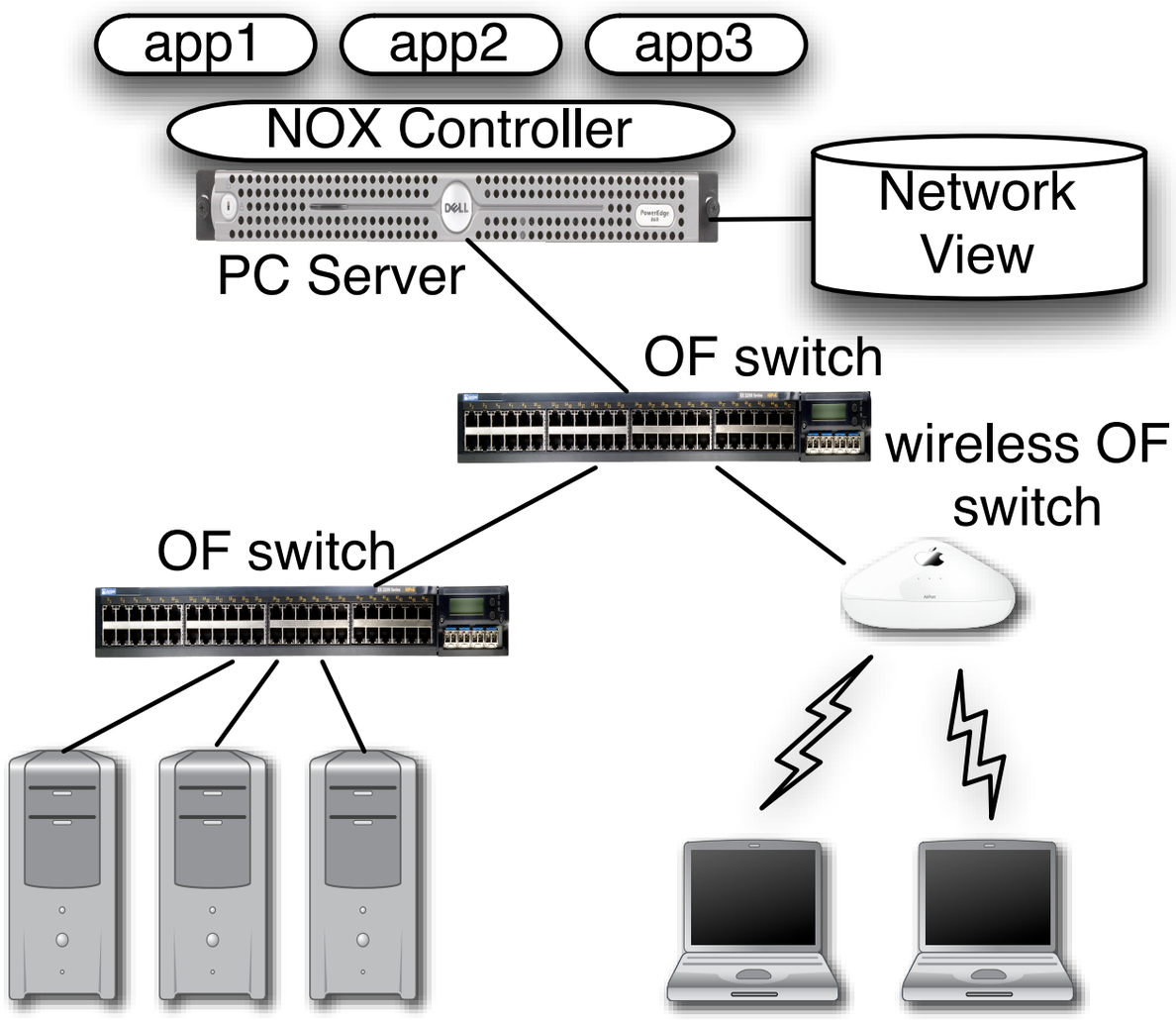
- use centralized controller to customize control

generality

- wide range of applications with diverse requirements

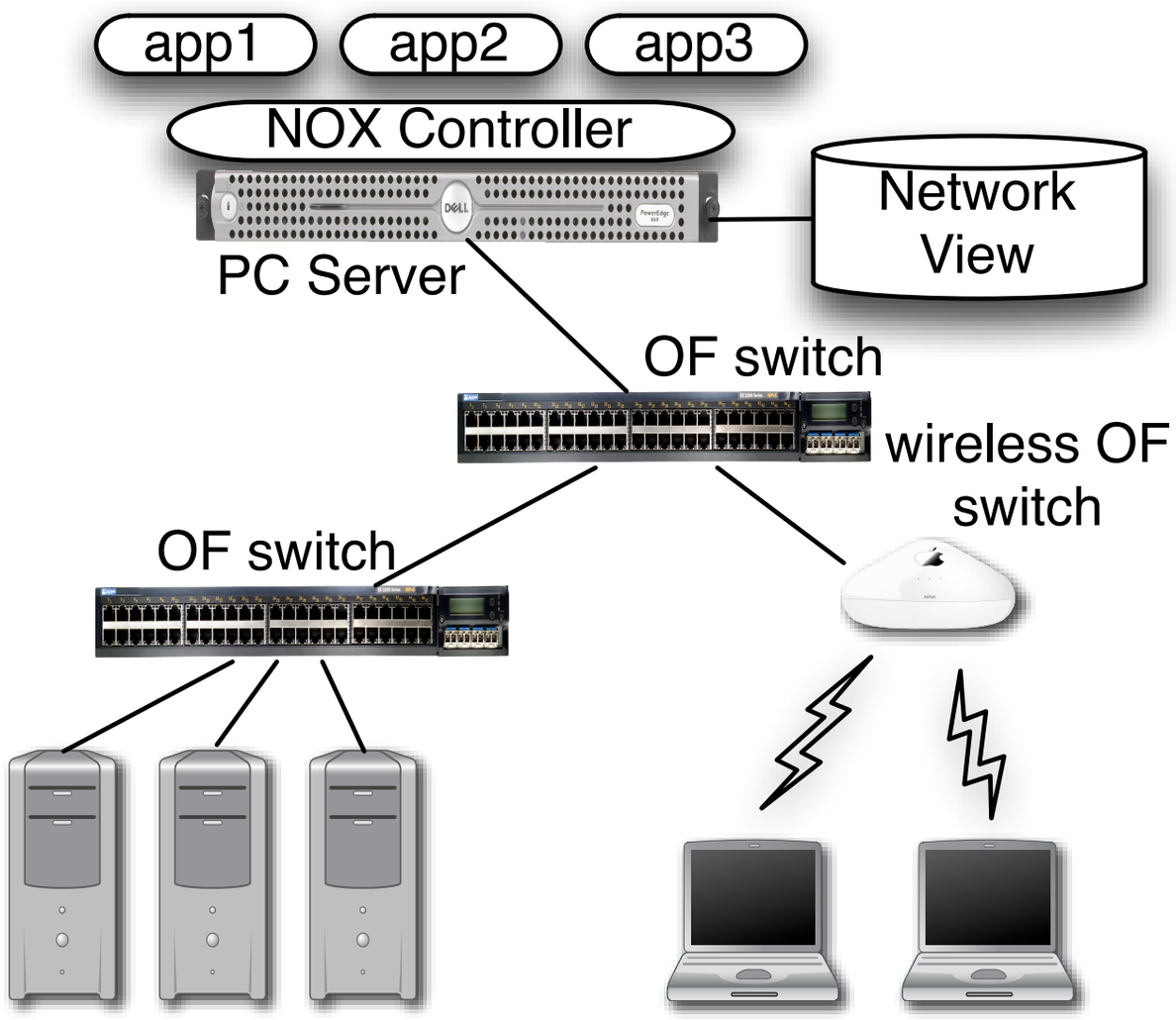
NOX

# overview

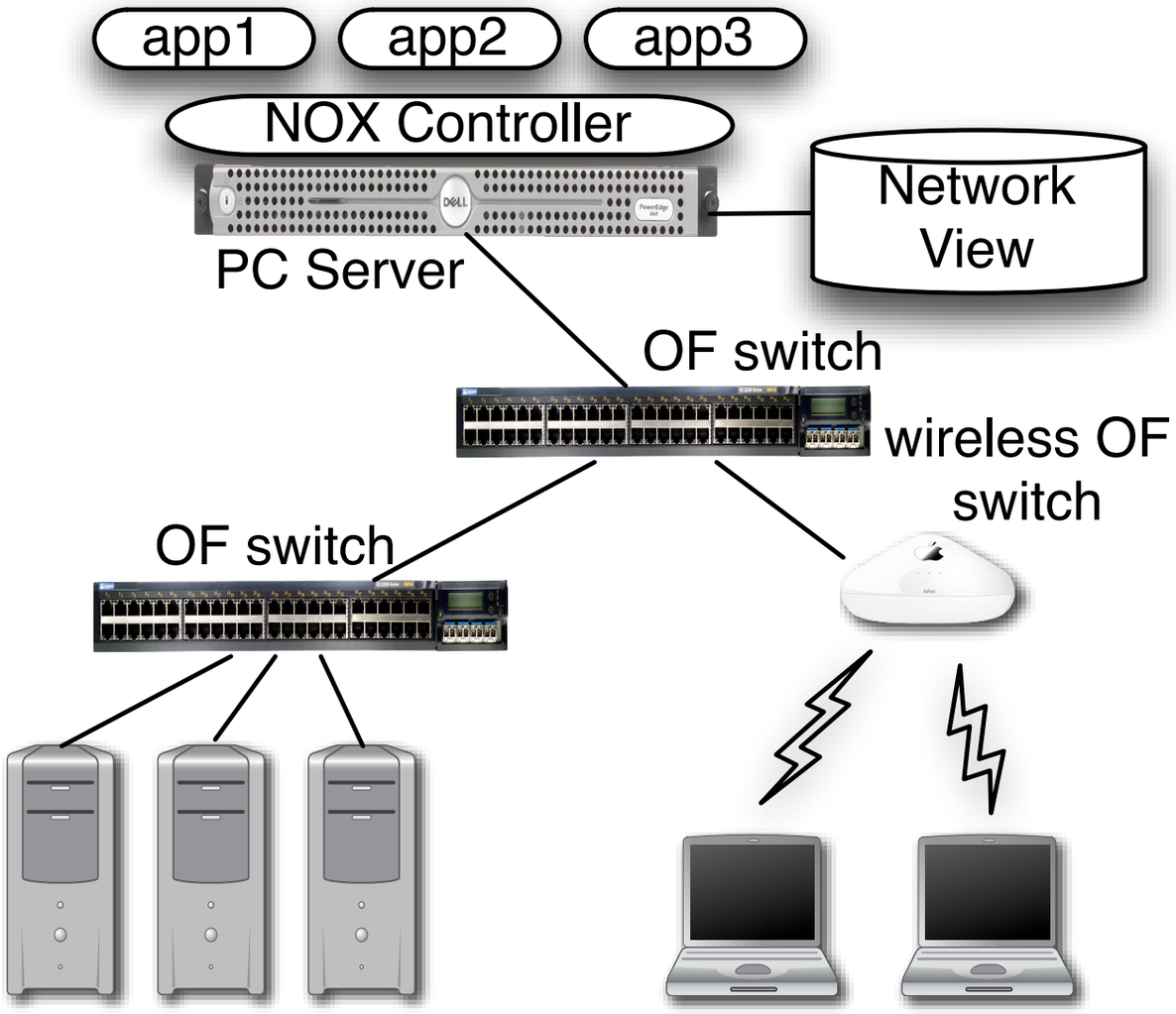



*"simple switches enslaved to a logically centralized decision element"*

# overview


app1
app2
app3
NOX Controller
Network View
PC Server
OF switch
wireless OF switch
OF switch

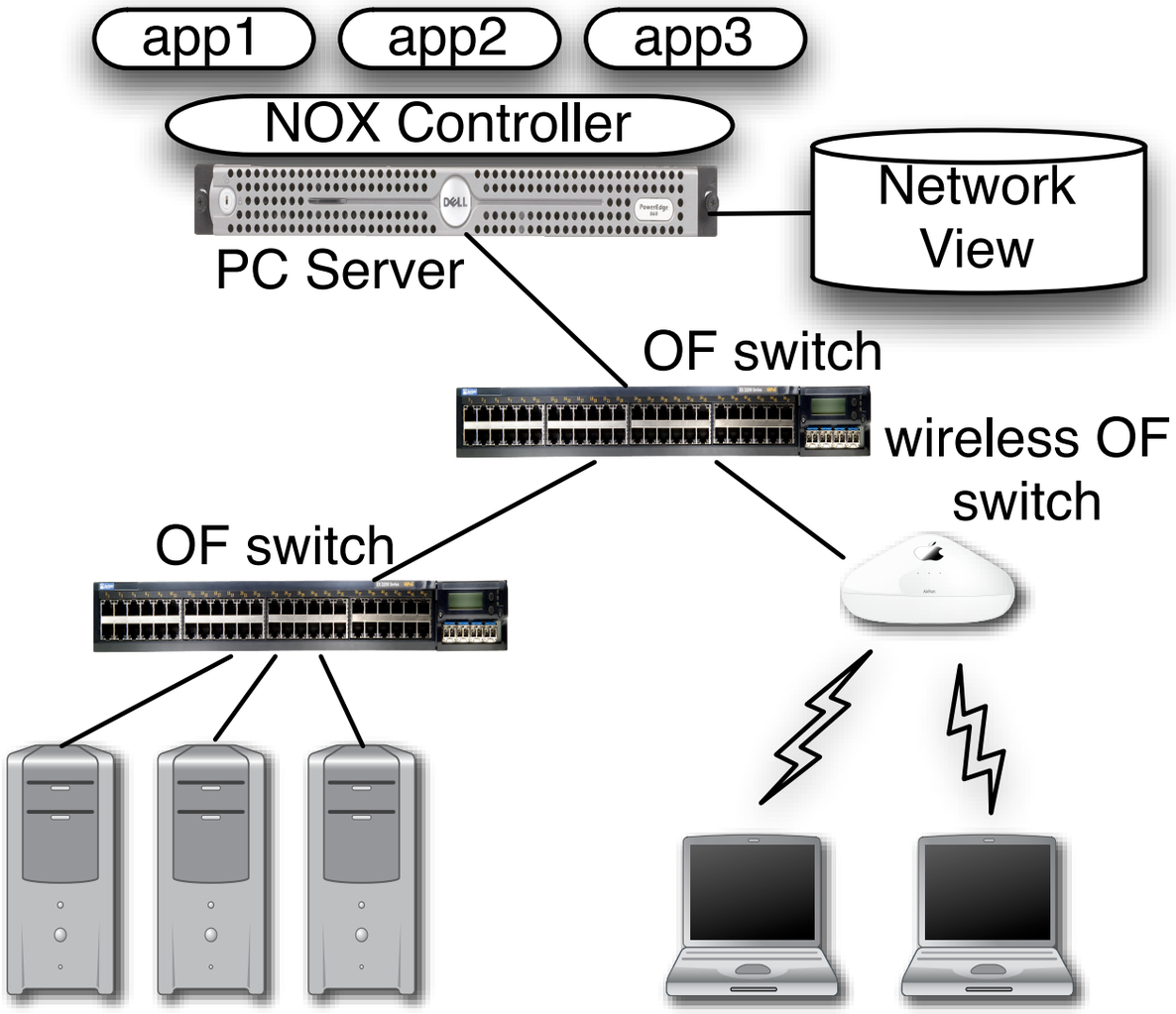
# overview




## network view

- topology
- locations of network elements
  - users, hosts, services
- *NOT* include traffic

# overview



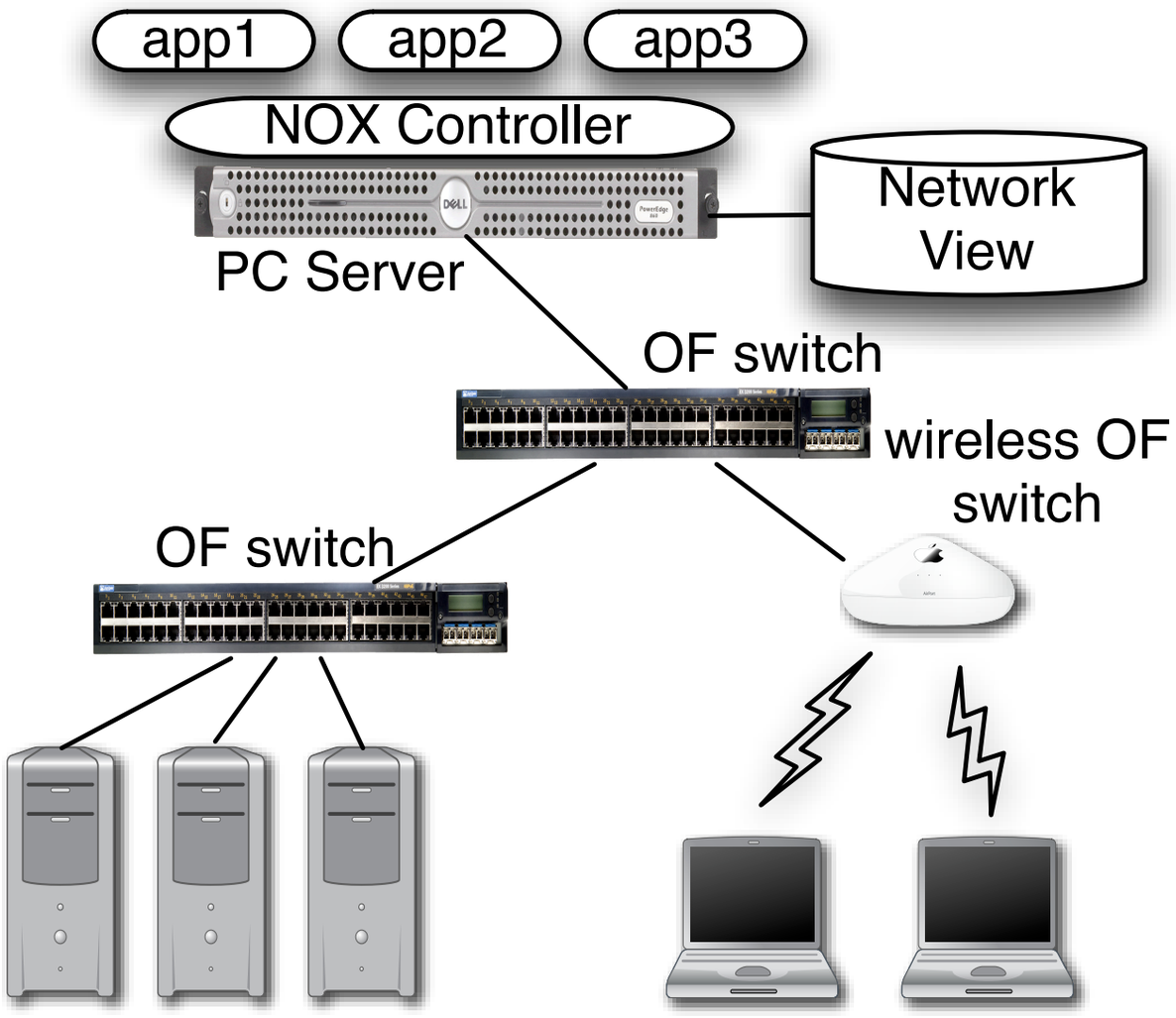

## network view

- topology
- locations of network elements
  - users, hosts, services
- *NOT* include traffic

## granularity

- flow based

# overview




## network view

- topology
- locations of network elements
  - users, hosts, services
- *NOT* include traffic

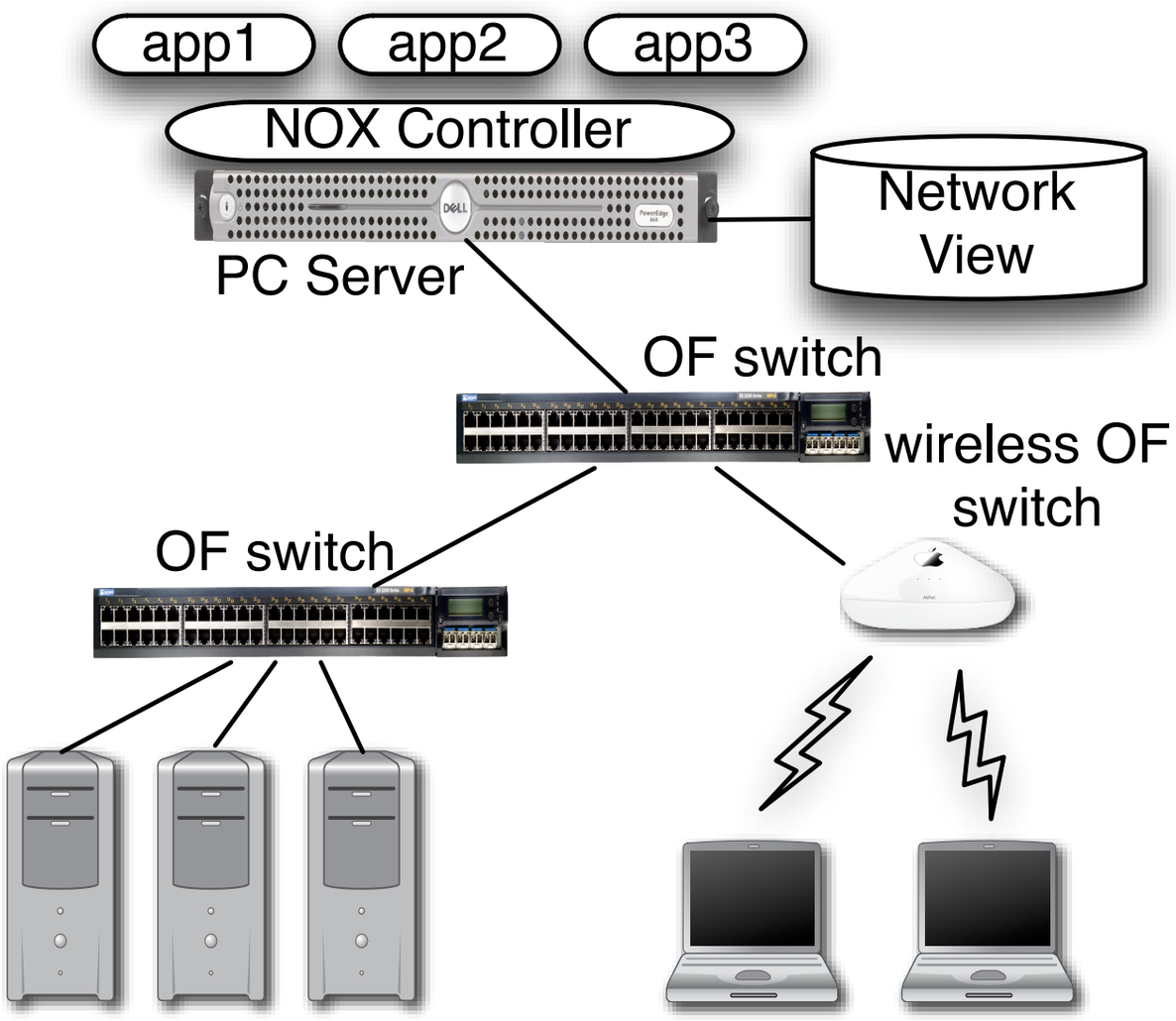
## granularity

- flow based

## switch abstraction

- OpenFlow
  - flow table
    <header: counter, actions>

# overview




## NOX extends Ethane

- scaling to large systems
- allowing general programmatic control

# programmatic interface

## events

- event handler, executed in order of its priority
  - applications register event handler

## network view and namespace

- maintained by “base” applications
  - user, host authentication
- enables topology independent management applications

## control

- exert through OpenFlow

# scalability

## differing timescales and consistency requirements

| | packet arrival | flow initiation | changes in network view |
|---|---|---|---|
| **timescales** | millions per second (10 gbps link) | one or more orders of magnitude less | tens of events per second |
| **consistency** | local storage (switches, controller instances) | | global, consistency across controller instances |

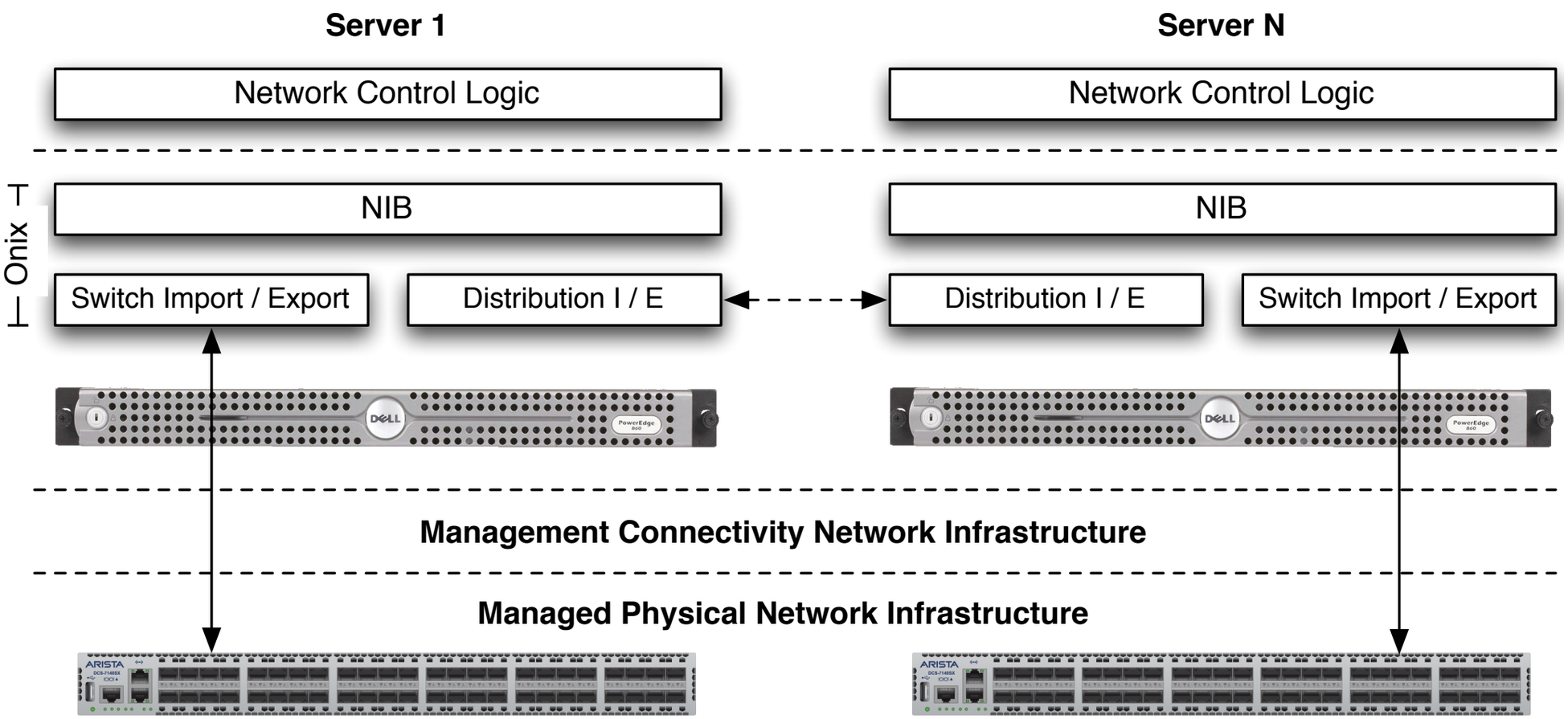
ONIX

# ONIX

## extends Ethane, NOX, RCP by

- far more general API
  - WAN, public cloud, data-center
- flexible distribution primitives
  - retaining performance/scalability trade-offs
  - without re-inventing distribution mechanism

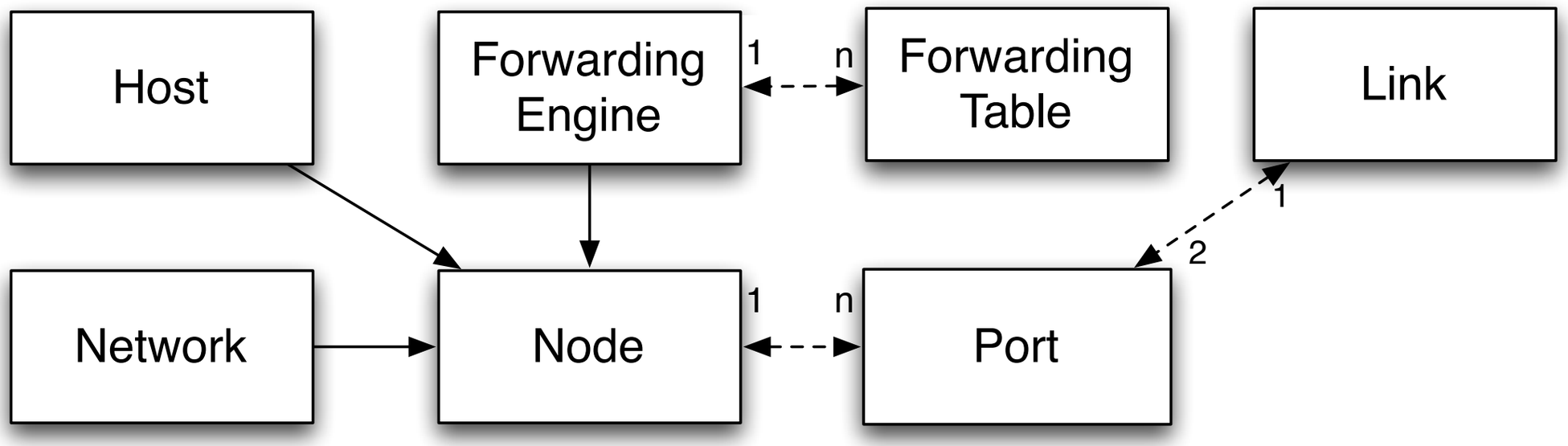
# ONIX overview




## Onix

- exposes unified view, disseminates network state view to other instances

# ONIX API: NIB




## NIB (network information base)

- apps (asynchronous) read, write, register notifications of changes
- Onix provides replication distribution
- apps provide conflict resolution, dictates consistency

# scalability

# scalability

*goal*

- *NIB not exhaust memory, # events not saturate CPU*

# scalability

## *goal*

- *NIB not exhaust memory, # events not saturate CPU*

## mechanism

- partitioning
  - a ONIX instance handles subset of network
- aggregation
  - a ONIX instance exposes NIB as aggregation to other instances
- consistency & durability
  - durability, strong consistency ⟵ transactional database
  - volatile state ⟵ memory based one-hop DHT

# reliability

## network element and link failures

- control logic steers traffic around the failures

# reliability

## network element and link failures

- control logic steers traffic around the failures

stringent requirement converge time:
backup paths with fast failure in the network element

# reliability

# reliability

## network element and link failures

- control logic steers traffic around the failures

# reliability

## network element and link failures

- control logic steers traffic around the failures

## ONIX failures

- running instances detect failed node and take over
- multiple instances simultaneously manage each network element

# reliability

## network element and link failures

- control logic steers traffic around the failures

## ONIX failures

- running instances detect failed node and take over
- multiple instances simultaneously manage each network element

## connectivity failures

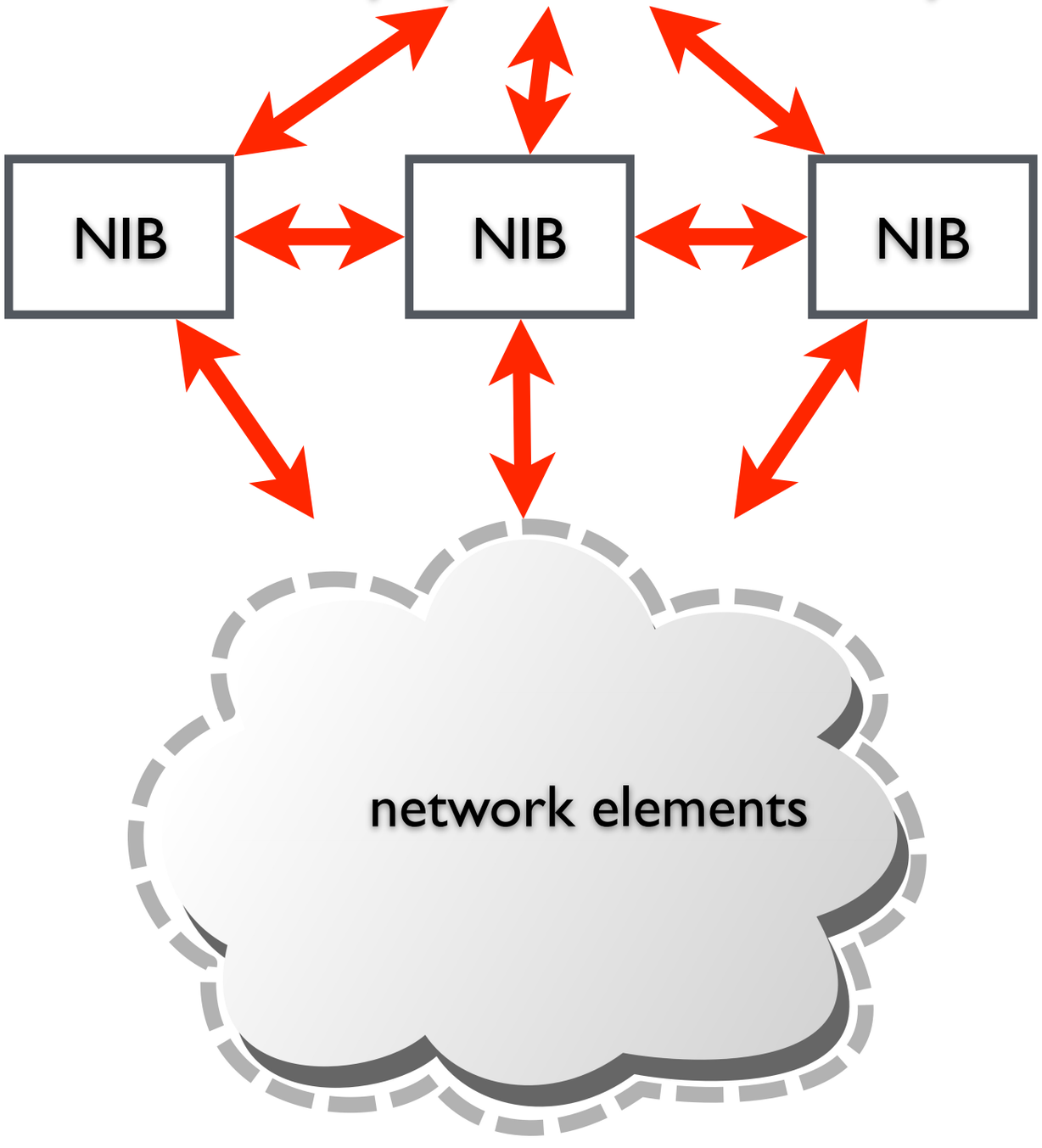
- use the management network for control traffic, isolating from forwarding plane disruption

# scalability & reliability

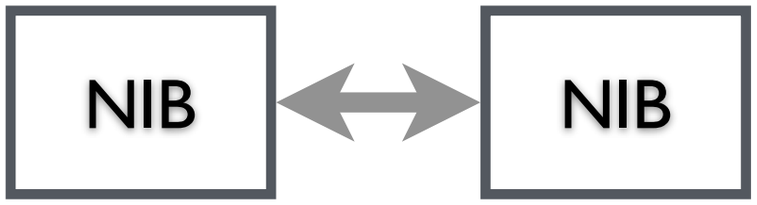
## enabling mechanism: distributing NIB

# distributing the NIB

applications with differing requirements on scalability, updates, and durability



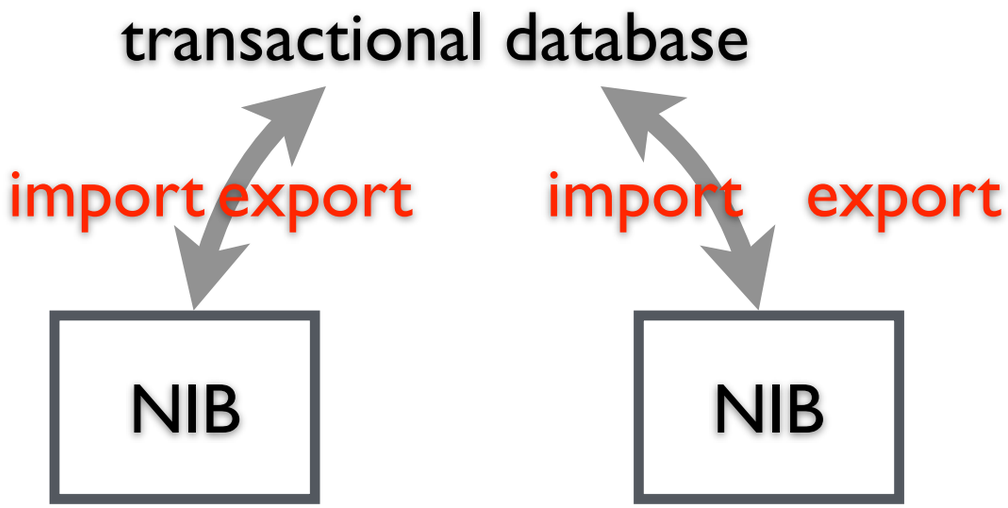
# distributing the NIB




between ONIX instances

- transactional database
- DHT and soft-state trigger

# distributing the NIB



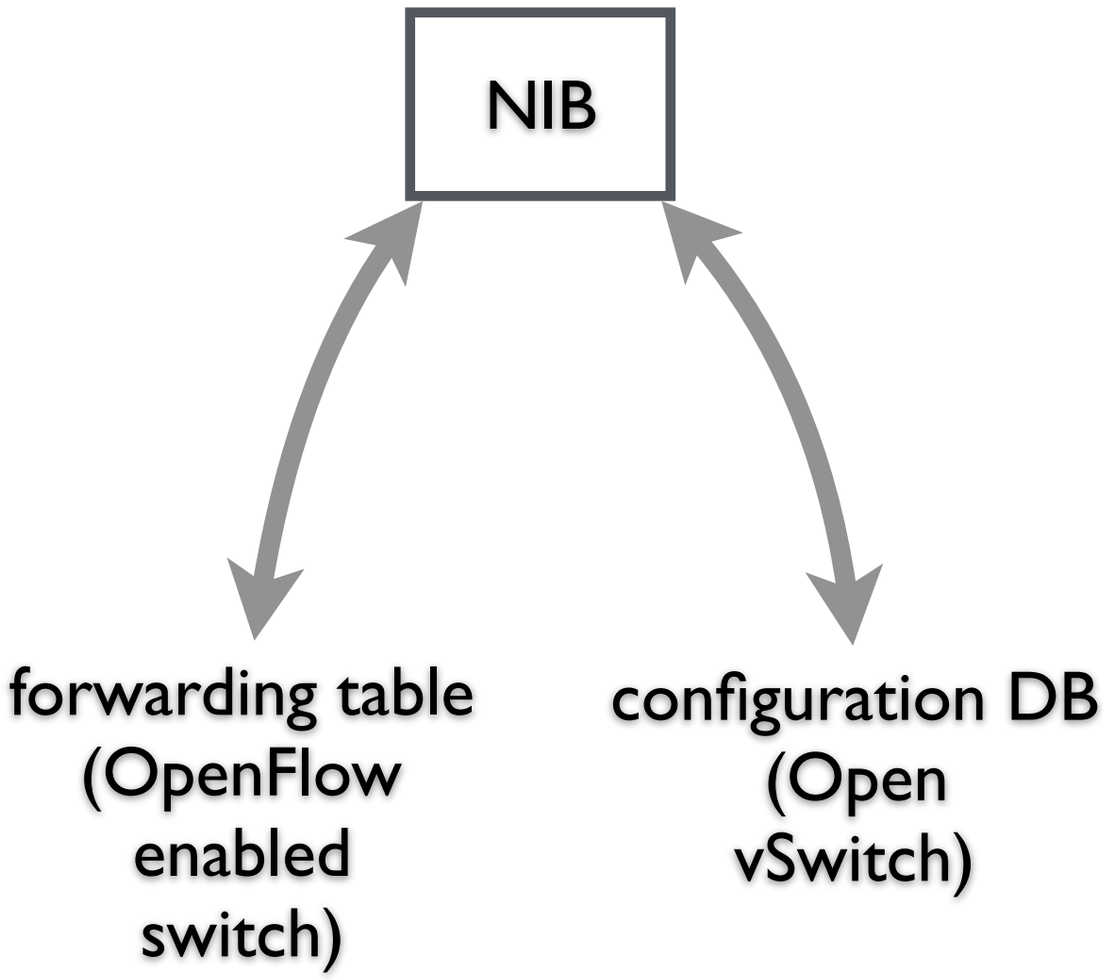

between ONIX instances

- transactional database
- DHT and soft-state trigger

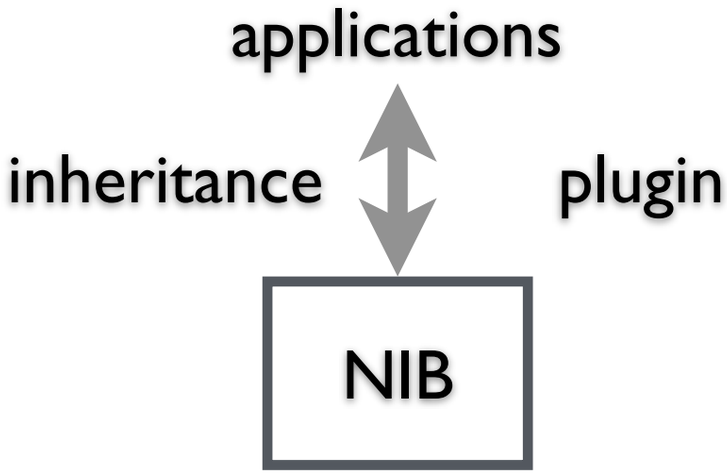
# distributing the NIB

between ONIX and network

- OpenFlow
- SQL



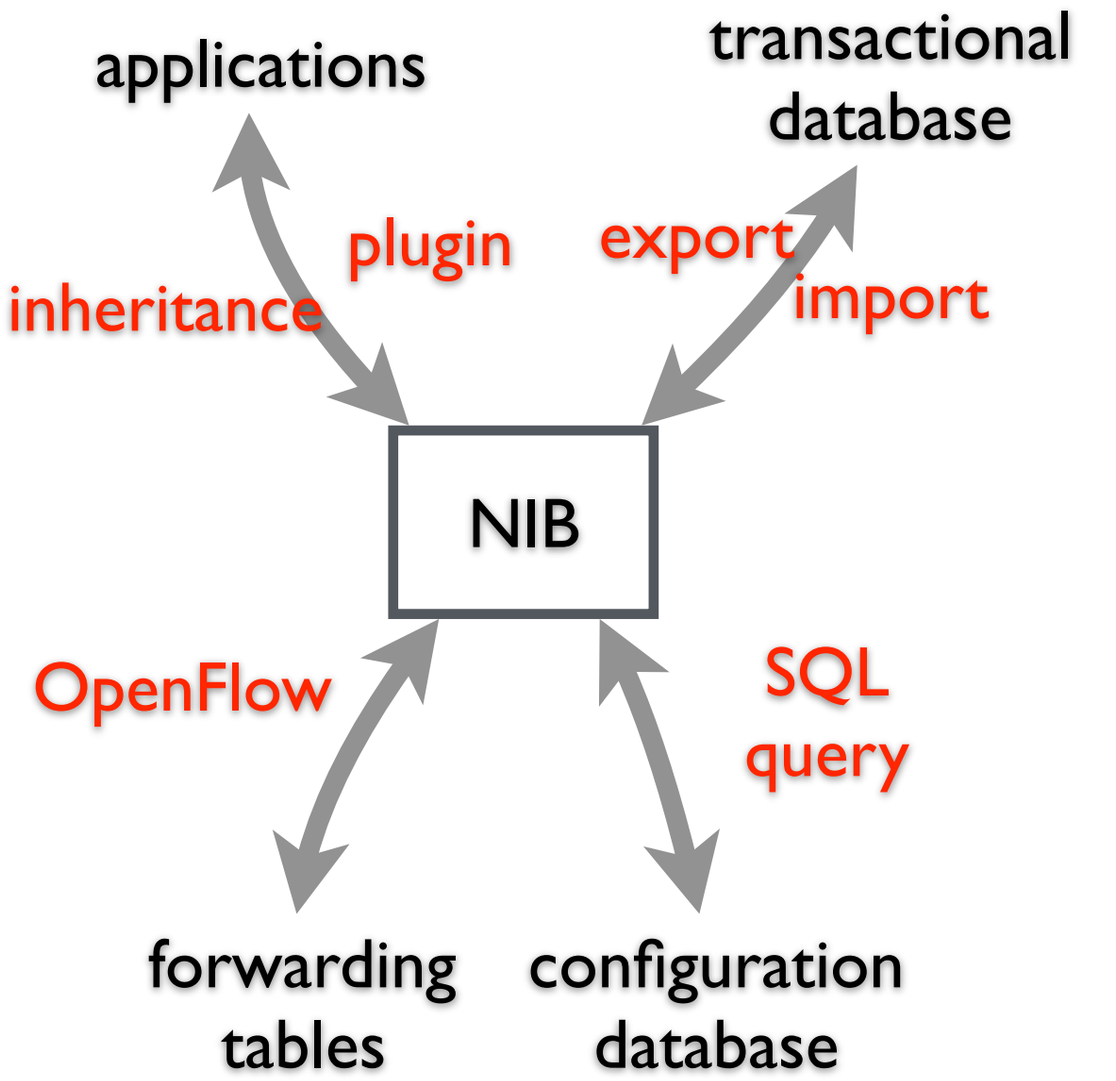
# distributing the NIB




## application-dependent conflict resolution

- by inheritance
  - applications inherit referential inconsistency detection
- by plugins
  - applications pass to import/export modules implement inconsistency resolution logic

# summary: distributing the NIB




## NIB

- the central integration point
- multiple data sources
  - ONIX instances
  - applications
  - network elements

# recap: opportunities and challenges

## performance

- low control-plane latency

## scalability

- large topology, huge volume of events, flow initiations

## reliability

- handle equipment (and other) failover gracefully

## simplicity

- use centralized controller to customize control

## generality

- wide range of applications with diverse requirements

# summary

opportunities

| | Ethane | RCP | NOX | ONIX |
|---|---|---|---|---|
| simplicity | ✓ | ✓ | ✓ | ✓ |
| API generality | | | ✓ | ✓ |

challenges

| | Ethane | RCP | NOX | ONIX |
|---|---|---|---|---|
| scalability | | ✓ | | ✓ |
| reliability | | ✓ | | ✓ |
| performance | | ✓ | ✓ | ✓ |

# discussion

## NOX

- inter-application coordination and isolation

## Onix

- a single “application” addressing several issues
- the control platform is not designed for multiple apps to control the network simultaneously