# Open Issues in configuration Management

Omar Cherkaoui , Ph.D.

UQAM

# Tout outil de Configuration doit

## OSS and Hardware

**CLI**
**IOS, IOX, Cat OS,**
**JUNOS,TL1**
**Interface**

## Network Services

**VPN, ACL,**
**Line Bundle, , LDP,**
**MPLS,**
**MPLS/TE**

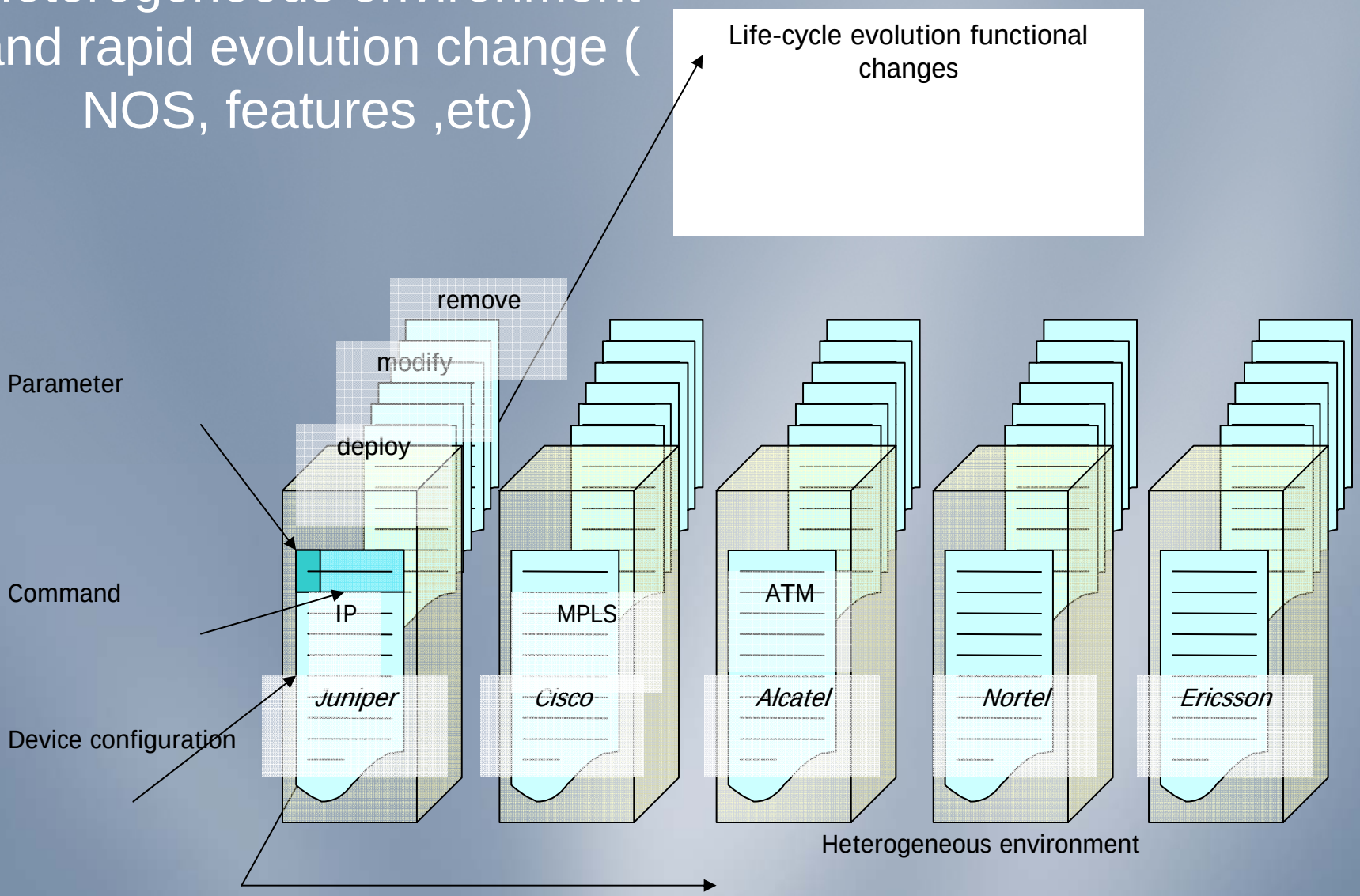
## Network Topology

**Interface**
**(Common, GigE, POS),**
**IP addresses**

## Network Routing

**RIP broadcast,**
**IGRP, OSPF** , ISIS

# Heterogeneous environment and rapid evolution change ( NOS, features ,etc)



# Establish  an automatic deployment services

- Introduce validation and test steps
- Introduce the validation step before the deployment
- Keep track on the configuration network
- Synchronization with  must source of platform information bases
- Abstract the IOS, CLI, Equipments, feature services

# Main issues for configuration management

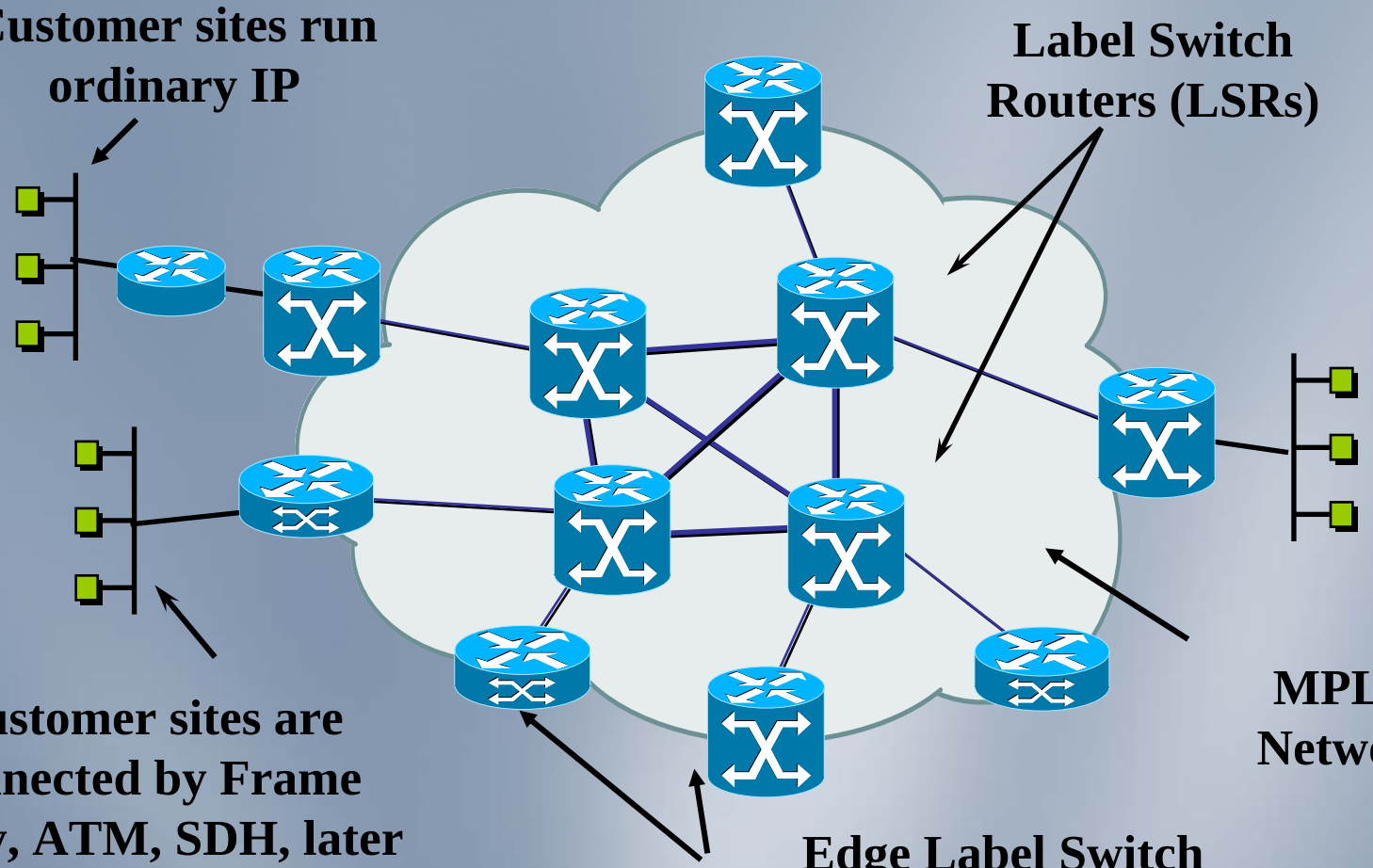
Dependency with network Operating Systems

Information Model

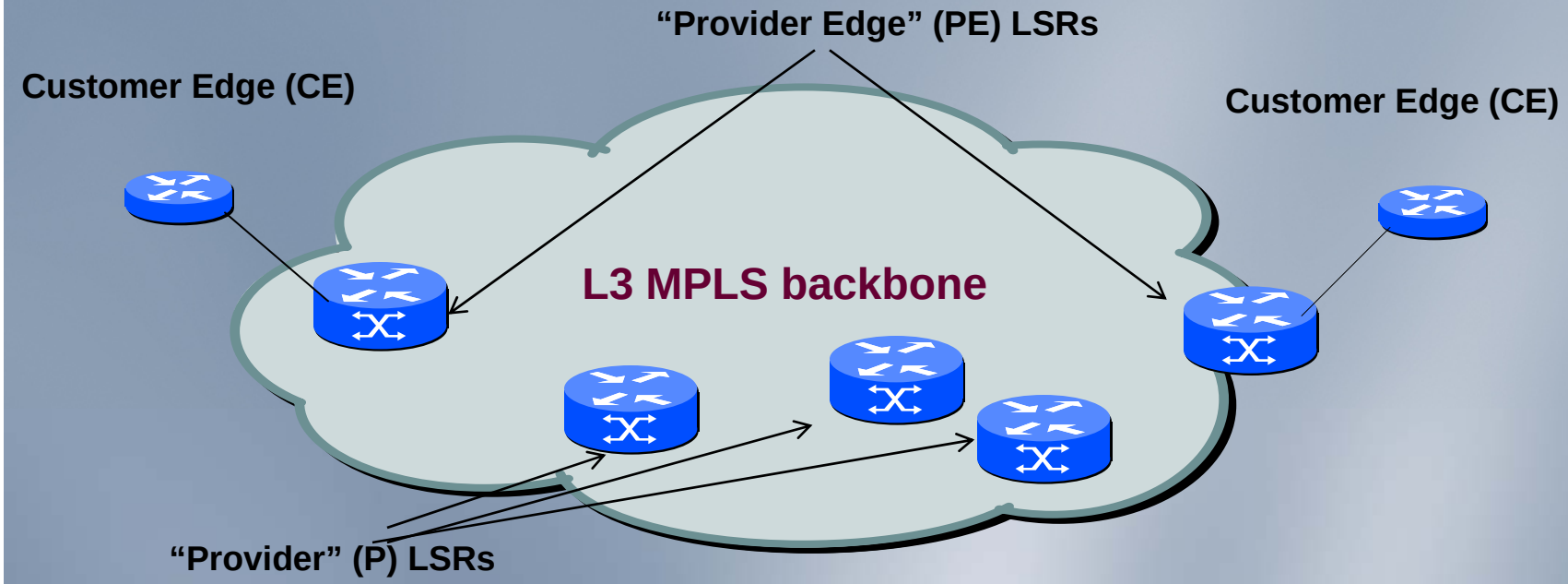
Adaptable Protocol

Security

# Sites


Customer sites run ordinary IP
Label Switch Routers (LSRs)
Customer sites are connected by Frame Relay, ATM, SDH, later Cable, xDSL, Dial
MPLS in the Network Core
Edge Label Switch Routers (LSRs)

# VPN Manager for MPLS




| Service deployment | Number of router | # major Steps | # parameters | Get States |
|---|---|---|---|---|
| Add site ( *n* sites) N=5 | *n* PE and *1* CE | 4 | 16 n by PE and 7 by CE (100) | 3 |
| Add VPN ( *n* sites) N=5 | *n* PE and *n* CE | 6 | 16 n by PE and 7 by CE (150) | 5 |

# Extensible by design
# Protocol

- Multi- level Protocol
  - Get device or multi-device state
- Transactional (client/server)
  - Atomic two phase or three phases (
- Consistency

- Connection/ Protocol State
- Session initiator
- Multiple controlling servers
- Ressource lock
- State Updates

# Primitives in Data- and Object-centric Protocols

From a very abstract viewpoint, the following set of essential management protocol

- primitives is needed for data-centric or object-centric management protocols:
  - GET, SET
  - CREATE, DELETE
  - SEARCH (or at the very least ITERATE)
  - LOCK, UNLOCK, COMMIT, ROLLBACK
  - NOTIFY someone about an asynchronous event
  - EXECUTE or INVOKE an operation or method
- Protocols that lack some of the primitives have proven to be problematic
- The locking primitives are needed to support transactions across sets of devices
- Command-centric protocols on the other hand usually have a very rich set of primitives (often hierarchically structured).

# *Order and Distribution of Operations*

The sequencing of the configuration operations:

their causal order, their distribution on multiple equipments and their transactional configuration and validation in this context.

- **Order of Operations**
- **Order of Configuration Operations**
- **Order of the Validation Operations**
- **Configuration Distribution**

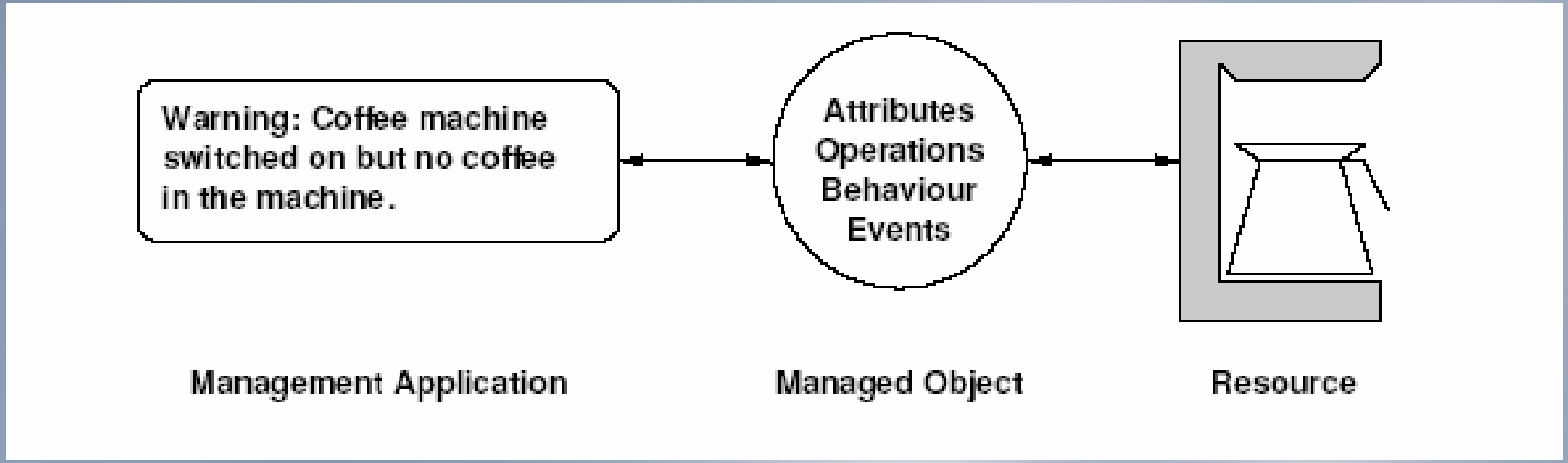
# Information Model?

- An *information model* that defines *management abstractions* of
  - Profiles and policies
  - Devices, media and protocols
  - Services
- Must be extensible to new devices and services as well as new uses

# Data- vs. Command- vs. Object-centric Approaches

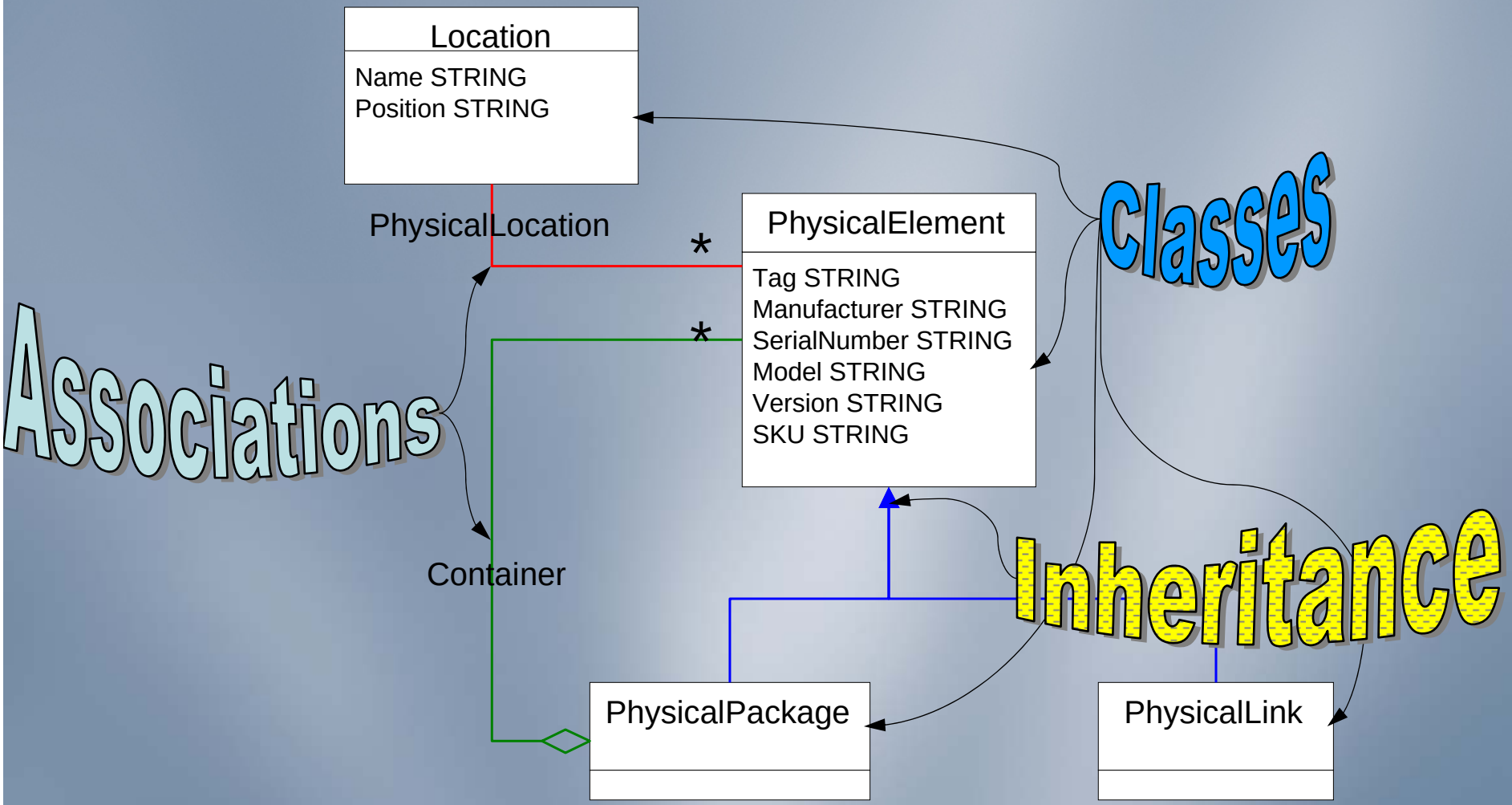
- Data-centric approach:
  - The device is represented as a collection of data objects which represent all the properties and capabilities of a device.
  - The management protocol manipulates the data objects representing a device.
  - Example: Internet management (SNMP) approach
- Command-centric approach:
  - The device is considered to be a stateful black box.
  - A set of commands can be send to the device to (a) change the state of the device or (b) to retrieve data about the current state of (portions of) the device.
  - Examples: Command line interfaces of routers or switches
- Object-centric approach:
  - The device is represented as a collection of data objects with associated methods.
  - This is basically a combination of the data- and the command-centric approach.
  - Example: OSI management approach (CMIP)

# Abstraction of Managed Objects (MOs)



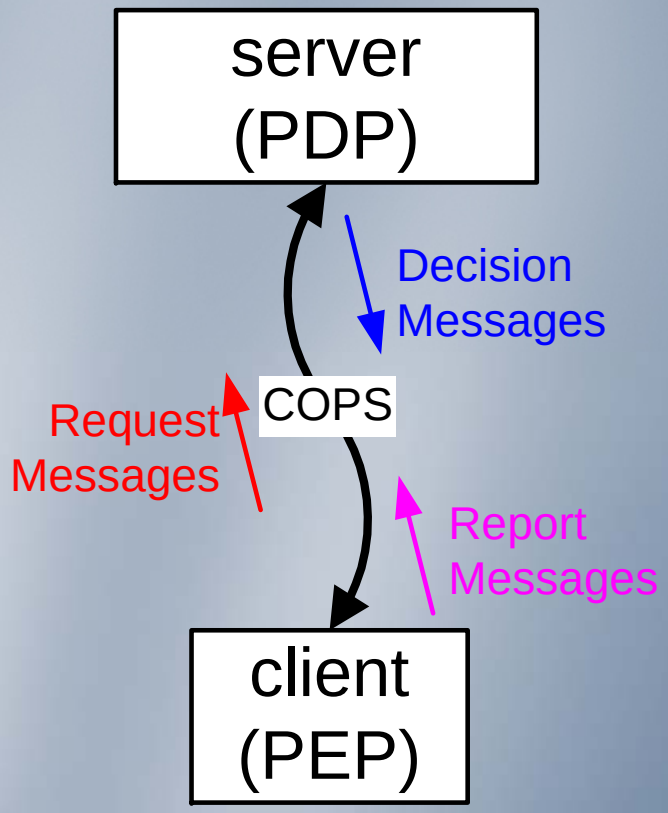

- A managed object is the abstracted view of a resource that presents its properties as seen by (and for the purpose of) management (ISO 7498-4).
- The boundary of a managed object defines the level of details which are accessible for management systems.

# Expressing the Schema:UML



# COPS Operation

- Connection-oriented, stateful protocol (TCP)
  - Simple client/server architecture
  - Efficient message passing
- Extensible by design
  - Outsourcing mode for managing signaled QoS
  - Configuration mode for managing provisioned QoS
- Secure communications
  - authentication and integrity
- IETF draft standard



# NetConf Solution

- The goal of the NetConf solution is to create a standard protocol for programmatic configuration of networks.

- The NetConf protocol suitable for network configuration, with the following characteristics:
  - Provides retrieval mechanisms which can differentiate between configuration data and non-configuration data
  - Is extensible enough that vendors want to provide access to all configuration data on the device using a single protocol
  - Uses a textual data representation, that can be easily manipulated using non-specialized text manipulation tools.
  - Supports network wide configuration transactions (with features such as locking and rollback capability)
  - Is as transport-independent as possible
  - Provides support for asynchronous notifications

# Issues to Resolve those issue thru Netconf

- Transport mappings
  - BEEP, HTTPS, SSH
- RPC Layer
  - SOAP encoding, xmlconf RPC, or simple request/response
- Advanced XML features
  - WSDL templates, XPath filtering
- Protocol Operations
  - Add, Modify, Delete Variants
    - Operation as element above data model, element within data model, or attribute within data model elements
  - Advanced operations: mandatory or optional
    - Checkpoint, Rollback, Locking
  - Multi-device operation support
  - Error Handling
  - Notifications
    - Use of Secure Syslog (RFC 3195) or SNMP-like notifications

# Some Issues for NetConf

- Peer to Peer
  - Unicast, connection-oriented, synchronous transactions
  - Either end can initiate the connection
- Session Based
  - User authentication and some protocol characteristics decided at session startup
- Extensible Operational Model
  - Base features + standard extensions + vendor extensions
  - Extensions determined by capabilities exchange at session startup

# Some Issues: Netconf

- Transport Independent, but certain requirements of transport are assumed
  - Connection-oriented
  - Most security features at transport layer, such as encryption and user authentication
- XML data encoding
  - Good balance between human and machine readable syntax
  - Config content can also be XML-wrapped (CLI) text
- Separation of protocol and data model
  - Will identify any data model issues which affect the protocol
  - IETF and vendors will create data models independently of the protocol development
  - XML Schema (XSD) will probably be used for initial data types and data modeling language

# COPS vs. NETCONF vs. SNMP

| Criteria | COPS | NETCONF | SNMP | Disadvantage/advantage |
|---|---|---|---|---|
| Connection | Reliable, TCP | BEEP/SOAP/ SSH | Non-reliable, UDP | Policy information size limitations, overhead of retransmission of full UDP payload |
| Session initiator | Initiator PEP (router) | Server | SNMP Server | COPS has automatic fail-over when server fails; |
| Protocol State | Statefull , no need for polling | Stateless | Stateless, need constant pooling | SNMP doesn't scale to PBN for large network. COPS transmits only difference in state |
| Multiple controlling servers | Not possible | Possible and likely | Possible and likely | Multiples masters may confuse the PEP |
| Ressource lock | Lock ressource actually used | Lock Ressource | None | Unlock resource may change |
| State Updates | Asynchronous, bidirectional, transactional | Asynchronous, bidirectional, transactional | SNMP Sets &traps | |
| Data Model and representation | Policy info Based with | XML/Schema | MIB | PIB designed for mass ( row) operation, röle allow virtual interface provisionning |