



# Towards Anticipative System Management ICNS/ICAS 2008



**Petre DINI**

**Alex CLEMM**

# About

- Current management patterns
- Problems with the events
- Acting in advance
- Missing pieces
- Q&A

# Current management patterns

# Management patterns

- Management Pattern
  - A way in which different actors interact in order to achieve a management purpose
  - Traditional management patterns
    - Request-response-based management
    - Event-based management
- Shift in networking trends impact the way in which management is approached
- Let us:
  - Explore past and existing management patterns
  - Assess impact of changing context on those patterns
  - Point out directions for further work and research

# Networking shifts

- Networks are becoming more intelligent
  - Tasks that used to require managing, no longer do
    - Load balancing, content switching, routing, ...
    - Autonomic management, “self-management”
  - Less micro-management required
- Shift in what is perceived as the main management issue
  - 90ies: Complexity because of number and heterogeneity of **devices**
  - Today: Complexity because of number and heterogeneity of **services** and service dependencies
    - Provision new services
    - Testing, troubleshooting
    - Definition of meaningful service level objectives + how to monitor
    - Understanding performance implications on underlying network
- Conventional management wisdom no longer always applies

# Management trends

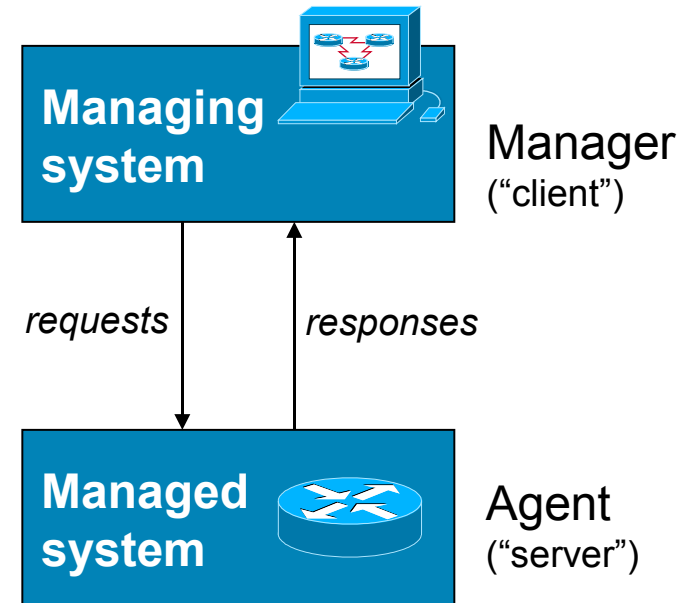
<b><i>Conventional Context</i></b>	<b><i>Conventional management approach</i></b>	<b><i>New context</i></b>	<b><i>New (additional) management approaches</i></b>
Network owned by and on premises of network provider	Manager-initiated management	Customers own equipment on their premises, as private networks	Agent-initiated management
Equipment, storage are expensive	Critical resources require careful mgmt	Operations support is expensive	Redundant resources, requiring little or no mgmt
Customers need service guarantees (voice!)	Services with guarantees	Some customers don't care about guarantees	No longer guarantees

# Management trends

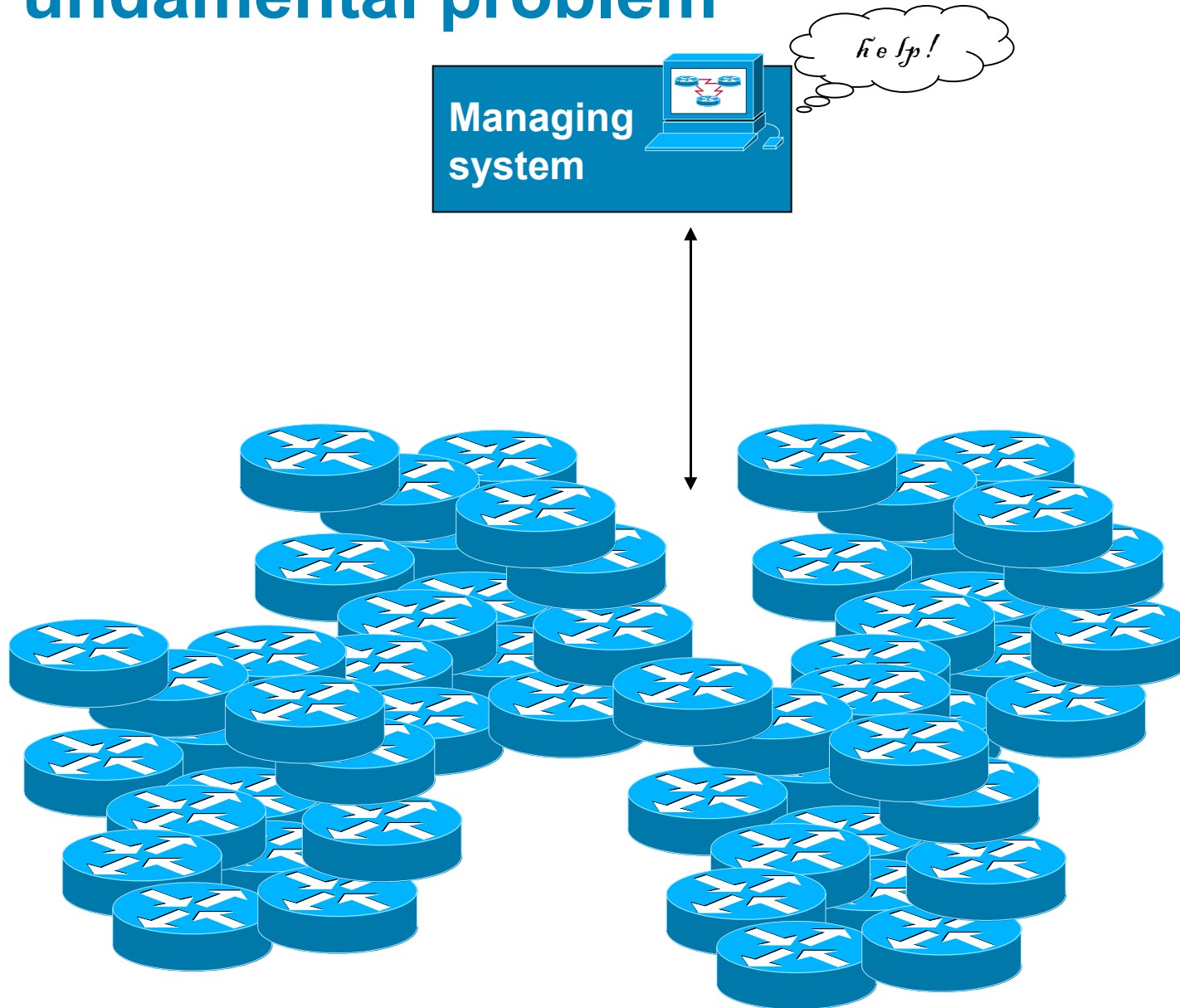
<i>Conventional Context</i>	<i>Conventional management approach</i>	<i>New context</i>	<i>New (additional) management approaches</i>
Customers buy complete service from one provider	Management of complete services	Customers will assemble their own services	Distinction between “data pipes” and services components on top
When something breaks, fix it	Reprovisioning	Networks need to be redundant enough to not require fixing	Rejuvenation
Customers are end users	Help	Customers can be administrators	Self-help

# The Basics: Request/ response pattern

- Conceptually simple
- Well understood
- Until today, basis for most management products, operations environments, management standards
- Problems with this pattern:
  - Scale
  - Heterogeneity
  - Responsiveness

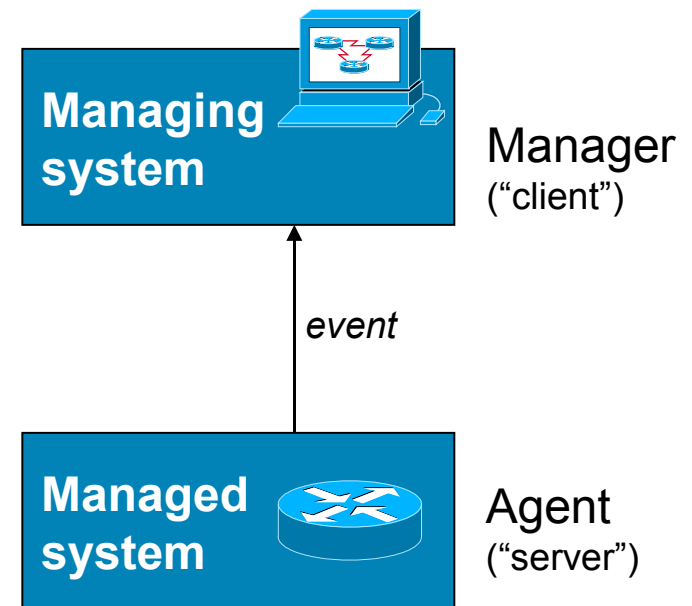


# Fundamental problem

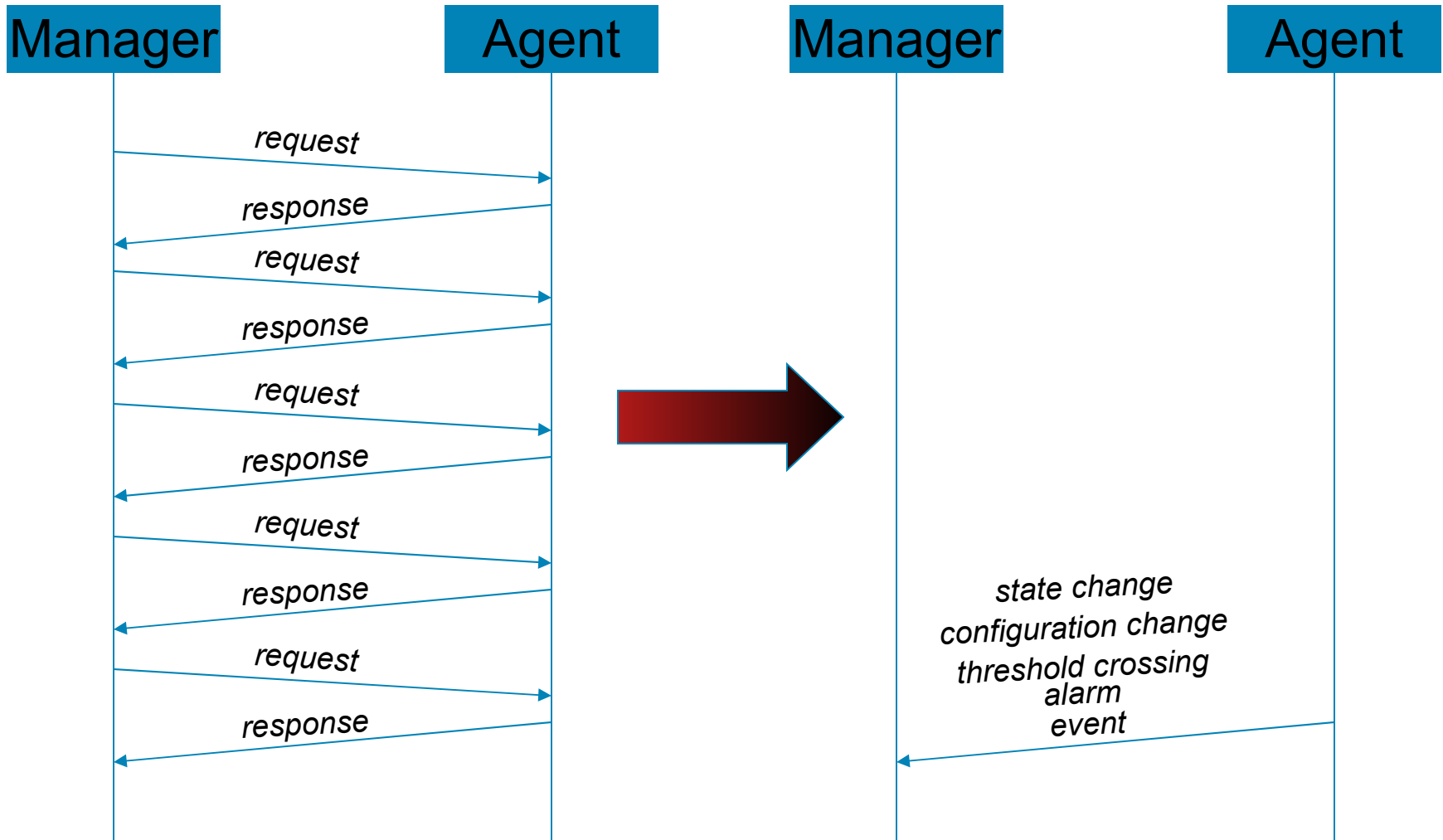


# The Basics: Event-based management

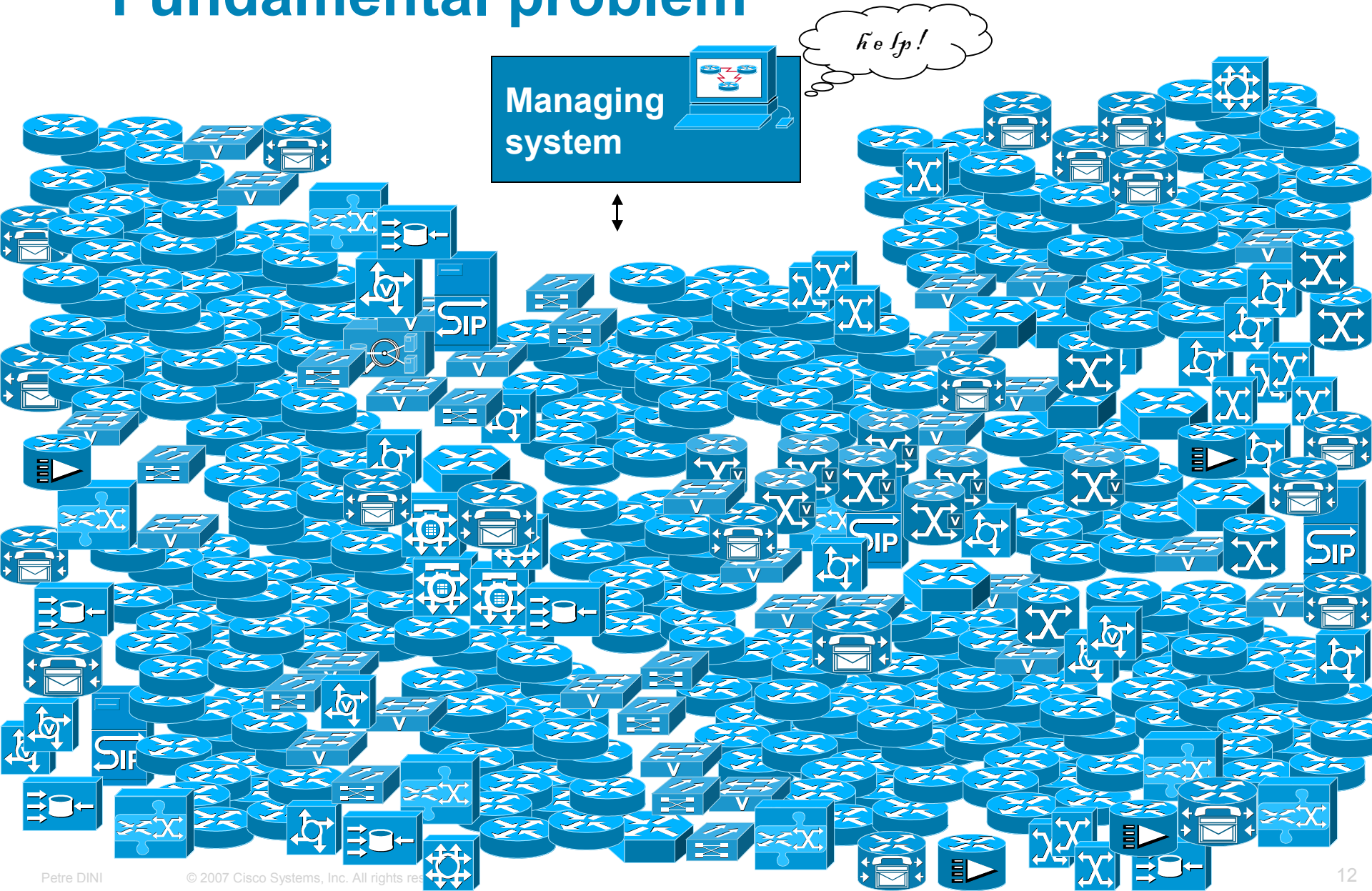
- Addresses scale, responsiveness issues
- Popular in monitoring  
but application in all other areas as well, e.g. configuration
- Well understood, not as often applied



# Event-based management



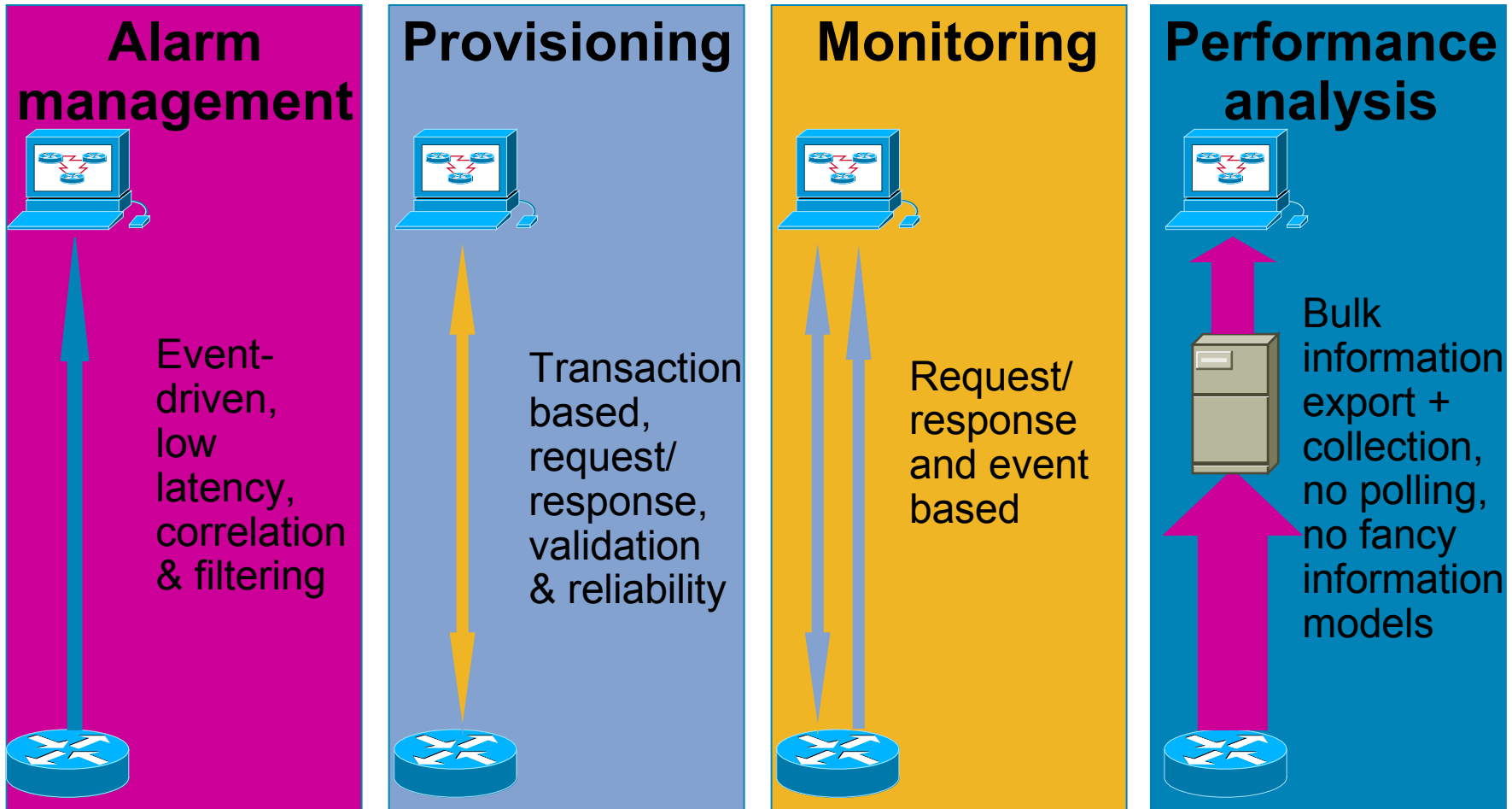
# Fundamental problem



# The Basics: Vertical partitioning

- “Divide et Impera”
- Distinction between functional areas
  - OAMP
  - FCAPS
  - Fulfillment, Assurance, Billing
- Typically aligns with management support organizations
- Per se, no impact on patterns
  - But each functional area can run its own pattern
  - Ultimately, patterns may diverge to best support the particular function

# Vertical partitioning



# Special-purpose protocols

- Special purpose: optimization of certain patterns
- Netflow/ IPFIX
  - Export of flow information
  - Huge data volumes; ship off to collectors for further analysis
- Netconf
  - “Glorified config FTP”
  - Facilitates configuration versioning, management transactions
- Syslog protocol
  - Structured data
  - Reliability provisions
- Emergence of management protocols for particular managed technologies and market segments
  - TR-69 (DSL Forum), PacketCable
- Compare with early 90ies: “one size fits all” protocols – SNMP, CMIP, TL-1

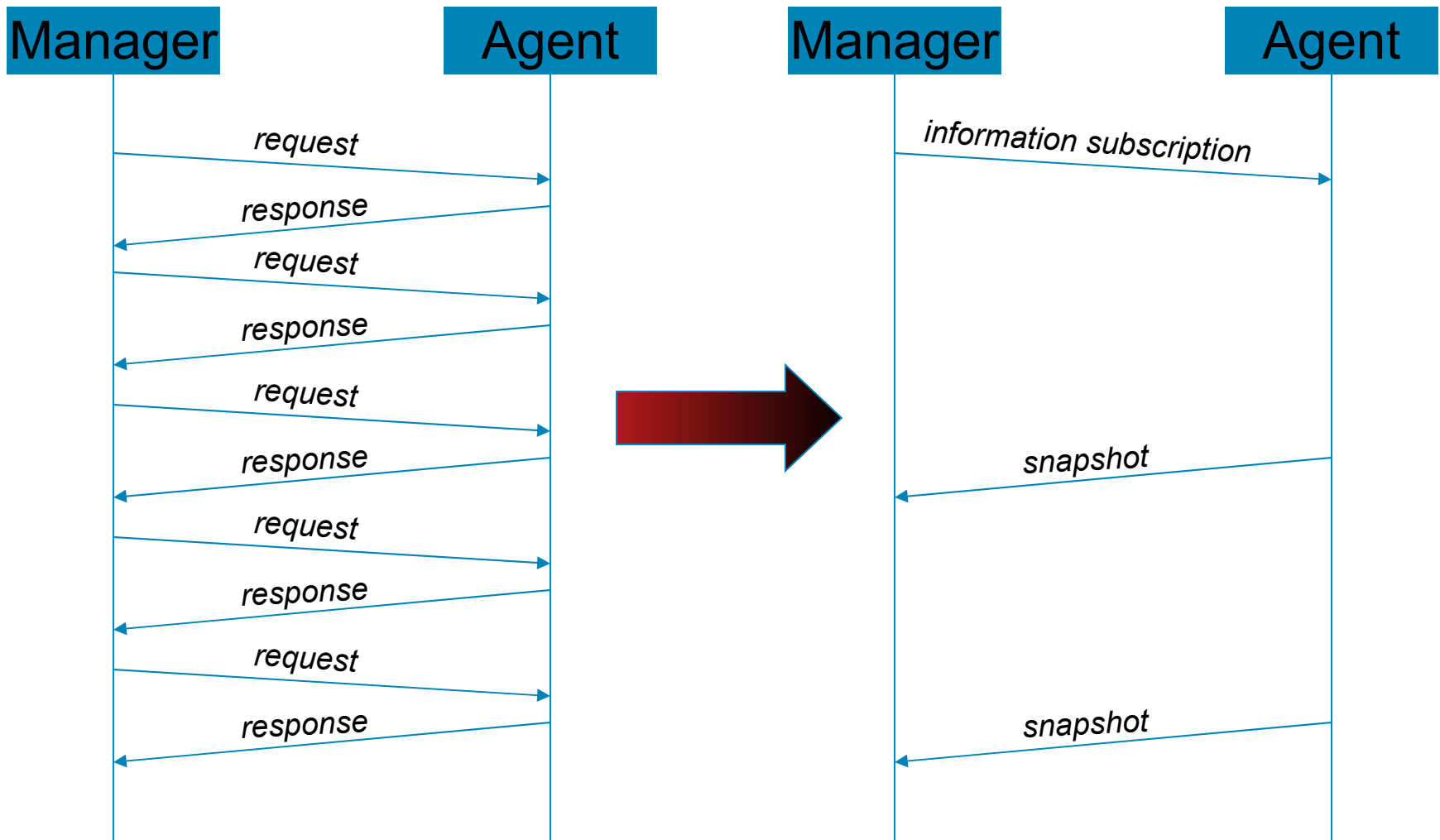
# The Basics: Horizontal partitioning – layering

- Introduction of management hierarchies
- TMN
- Management by delegation
  - MOM
  - Pollers, Netflow collectors, ...
  - Offloading of simple, repetitive tasks
- Management by objectives
  - Policy-based management
- Additional hierarchy layers are not necessarily on top
  - Virtualization as example of mgmt functions moving into network

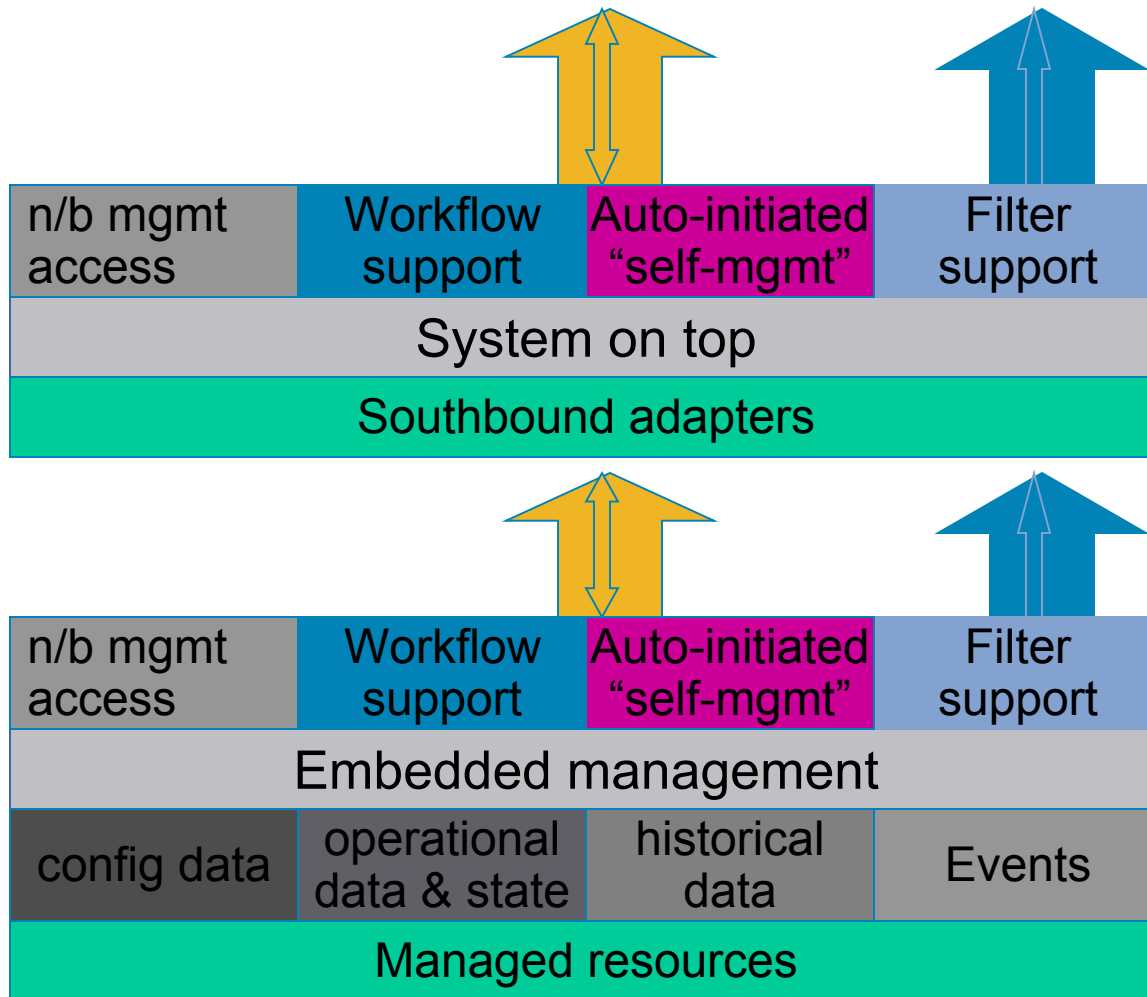
# Optimizing management patterns

- Reducing the amount of information exchanged
  - Events: Less noise, more signal – event filtering + correlation
  - Communicate policies, not micro-management
  - Anticipate information of interest
- Reducing management exchanges within each management interaction
  - Replace events that result in subsequent polling with events that anticipate additional information required
  - Replace polling by subscription for information export
- Reducing required management interactions
  - Convert polling- to event-based management where possible
    - change notifications
    - anomaly detection
    - requires reliable events
  - Close control loops: Self-management, autonomic systems
- Subject of continuous improvements

# Example: Periodic information export



# Increasing management intelligence



- Reduce amount of information exchanged
- Reduce exchanges required for each mgmt interaction
- Reduce frequency of mgmt interactions

# Inherent limitations of traditional patterns

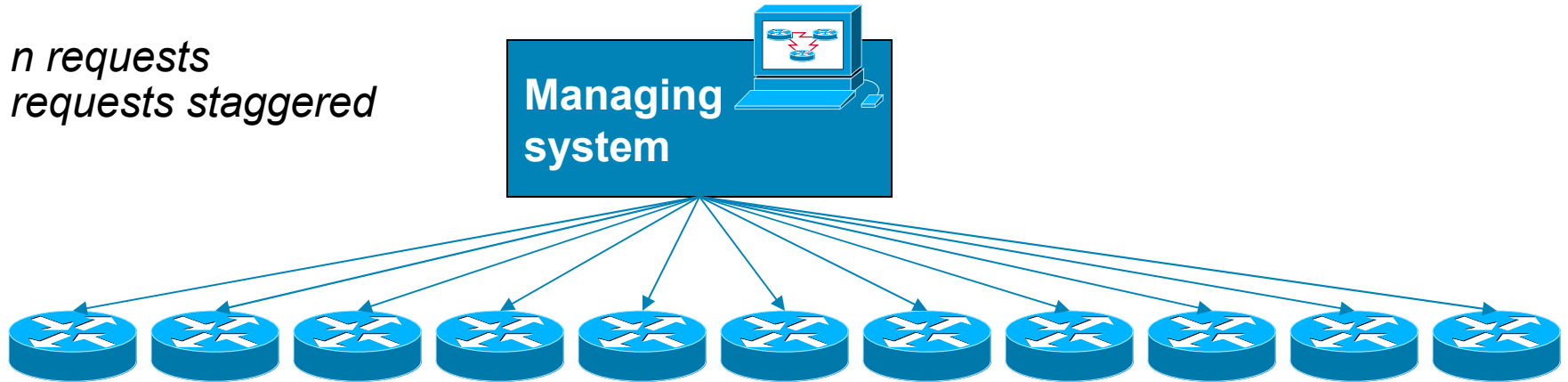
- Scale and heterogeneity are not the only problems...
  - How do you manage private networks behind firewalls that you have no access to?
  - How do you distribute security updates when you don't know everything that's in the network?
  - How do you deploy a service involving equipment of a residential customer without touching it?
  - How do you activate a configuration so that it takes effect at the same time across the network?
- But they continue to be problems
  - How do you identify the top 10 utilized links in your network, right now?
- Need to move beyond traditional patterns and rethink how systems in management can communicate

# Newer patterns: publish/ subscribe

- Deployment scenario:
  - More managed resources out there than you care about
  - Iterating through devices individually too low-performant
- Approach: Publish/ subscribe
  - Publish events on topics
  - Interested parties subscribe
  - From direct naming + addressing to associative addressing
- Examples
  - Upgrade all images of a certain revision
  - Inventory

# Publish-subscribe

*n requests  
requests staggered*



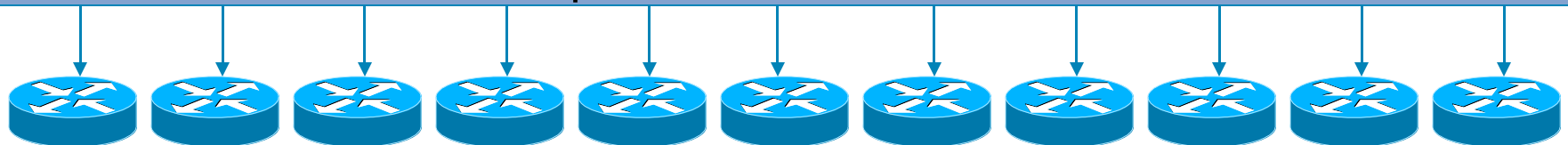
*1 request only  
synchronized delivery*



Examples:

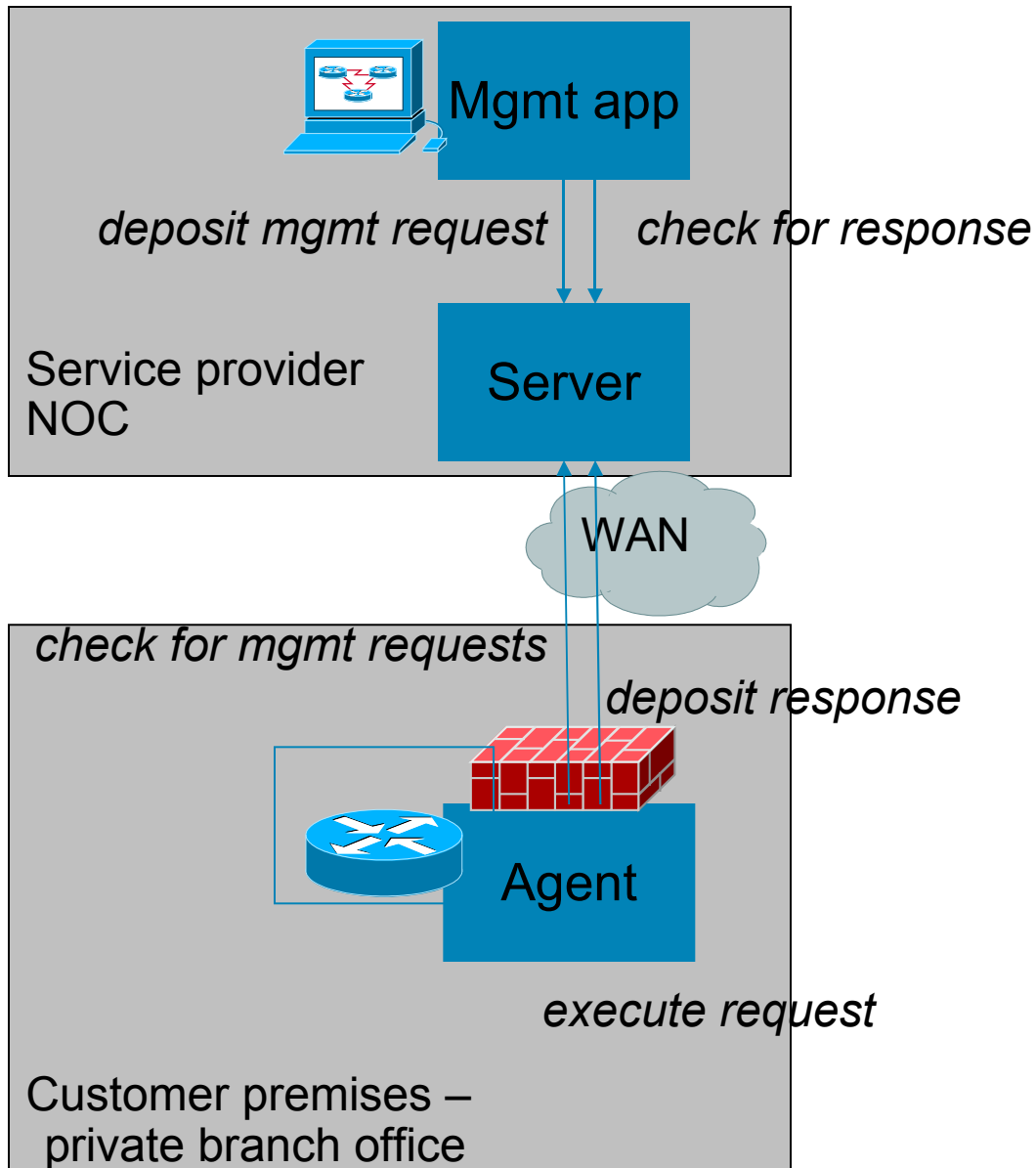
- upgrade-image
- 28xx/upgrade-image
- activate-config
- collect-snapshot

publish/subscribe bus



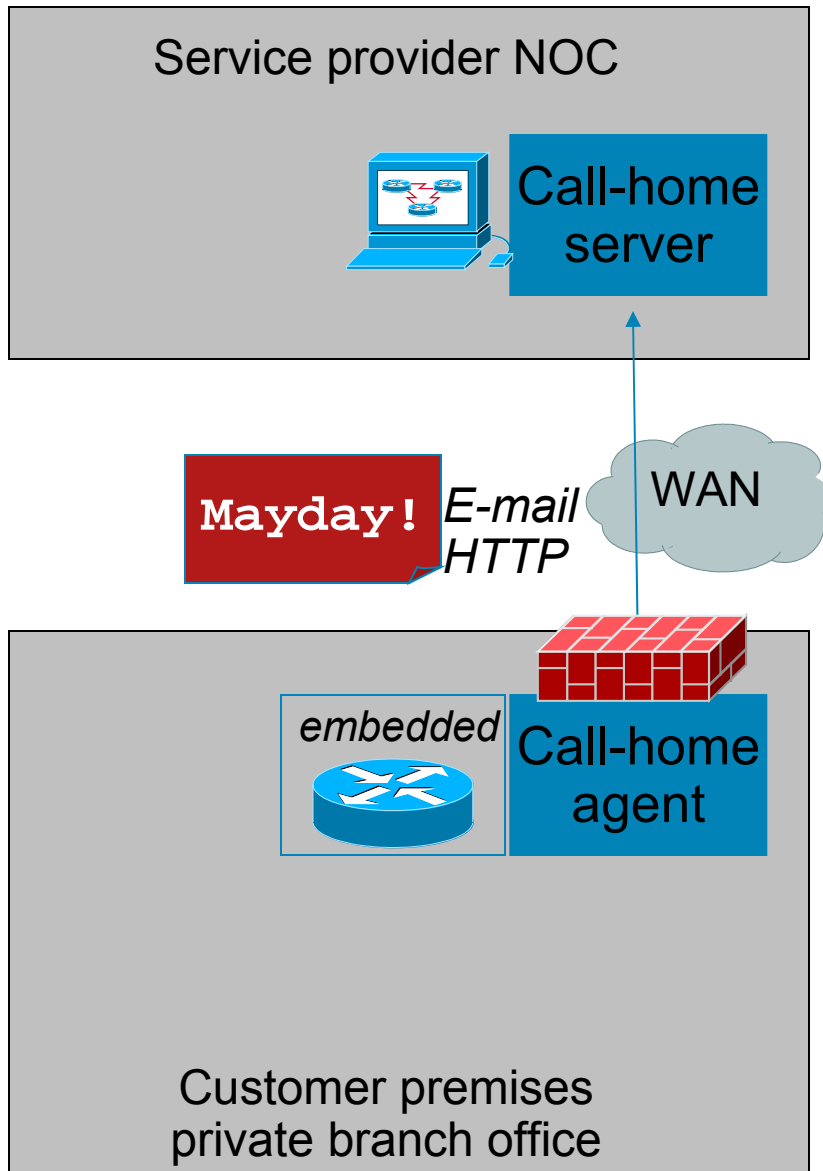
*subscribe to topics*

# Agent-initiated management patterns



- Deployment scenarios:  
Reaching devices for management a challenge  
CPEs, NAT
- Device reaches out when it needs to be managed  
Reboot  
Periodic
- Reversal of “traditional” pattern
- Examples: CNS, DSL, Packetcable

# Agent-initiated management – Call-Home



- (3) Offline analysis, no polling reqd.
- (2) Send message w/ event + add. info
- (1) When device in distress, collect
  - inventory, sw image, ...
  - alarm conditions, log
  - current config

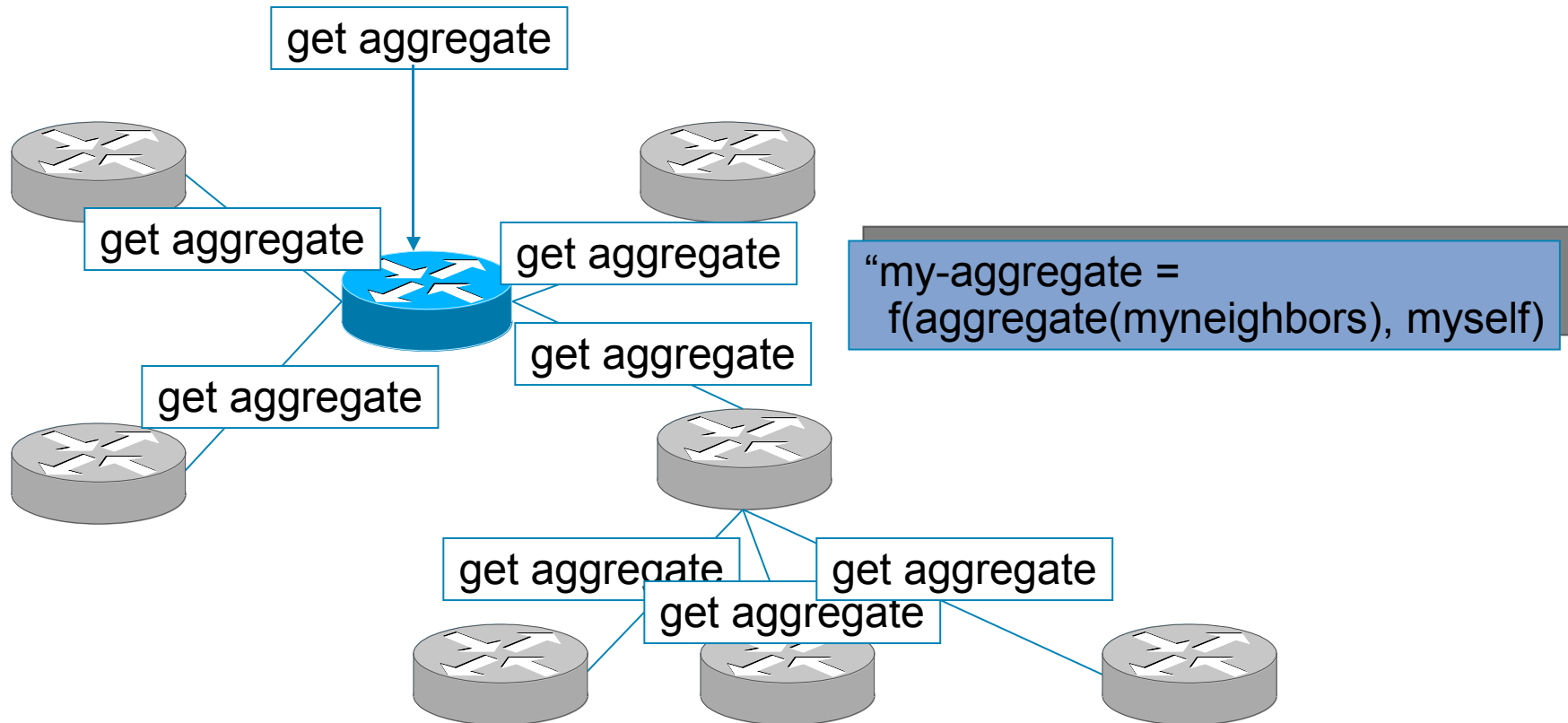
When device in distress,  
even if alarm can be sent,  
additional interactions to troubleshoot  
device required

# Collaborative patterns/ peer-to-peer

- Non-hierarchical, collaborative, “flat” management schemes
- Often involve an overlay peer-to-peer management topology
  - Peers execute management requests on behalf of other peers
  - Can involve formation of ad-hoc, short-lived hierarchies
  - Blurring boundary between “managing” and “managed system
  - Requires digital trust schemes (digital certificates, signatures, etc)
- Offload simple management tasks into the network
  - “Simple” for individual nodes
  - Scale problem if conducted centrally
- Potential applications
  - Aggregation of information (next slides)
  - Dissemination of information
  - Cross-device monitoring, detection of anomalies
  - Cross-device diagnostics
- Promising, still largely futuristic

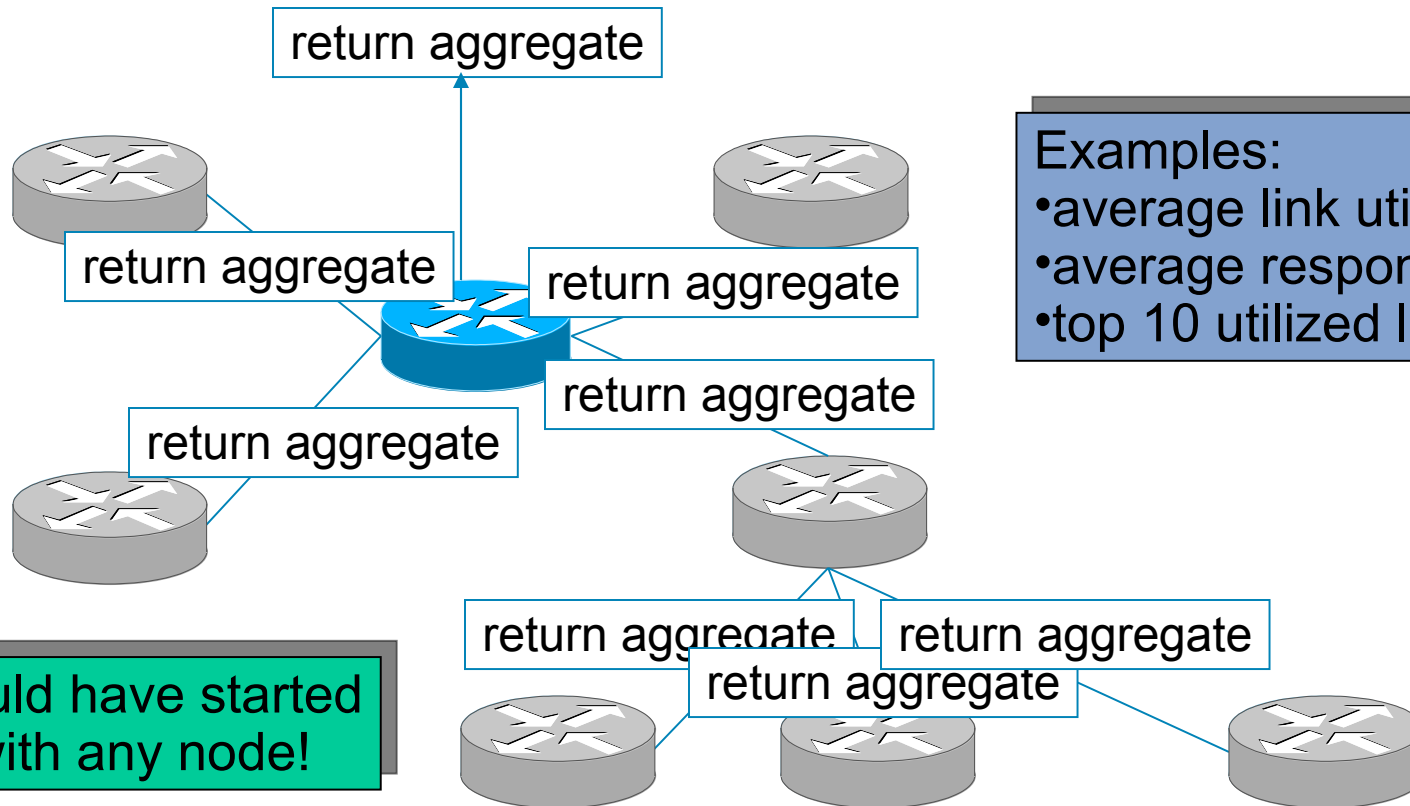
# Collaborative patterns

- Example:  
Generic Aggregation Protocol (GAP) and derivatives  
*Stadler et al (KTH), Raz et al (Technion)*



# Collaborative patterns

- Example:  
Generic Aggregation Protocol (GAP) and derivatives  
*Stadler et al (KTH), Raz et al (Technion)*



Examples:

- average link utilization
- average response time stats
- top 10 utilized links

Could have started with any node!

# Concluding on management patterns

- Management communication patterns need to evolve with networking context
- Some areas to think about going forward:
  - Complexity models for management tasks and associated patterns
  - Identification of new, more efficient patterns for common tasks
  - Improvements in exception-based management
    - Anomaly detection
  - Collaborative peer-to-peer management for various applications
    - Collaboration between devices
    - Collaboration between users
    - Network diagnostics, effective information dissemination, ...
  - What impact if any does increased service-orientation have (versus network and device-orientation)
  - Where does new standardization make sense
    - e.g. agent-initiated management?

# Considering the events

# Positioning

## Issues

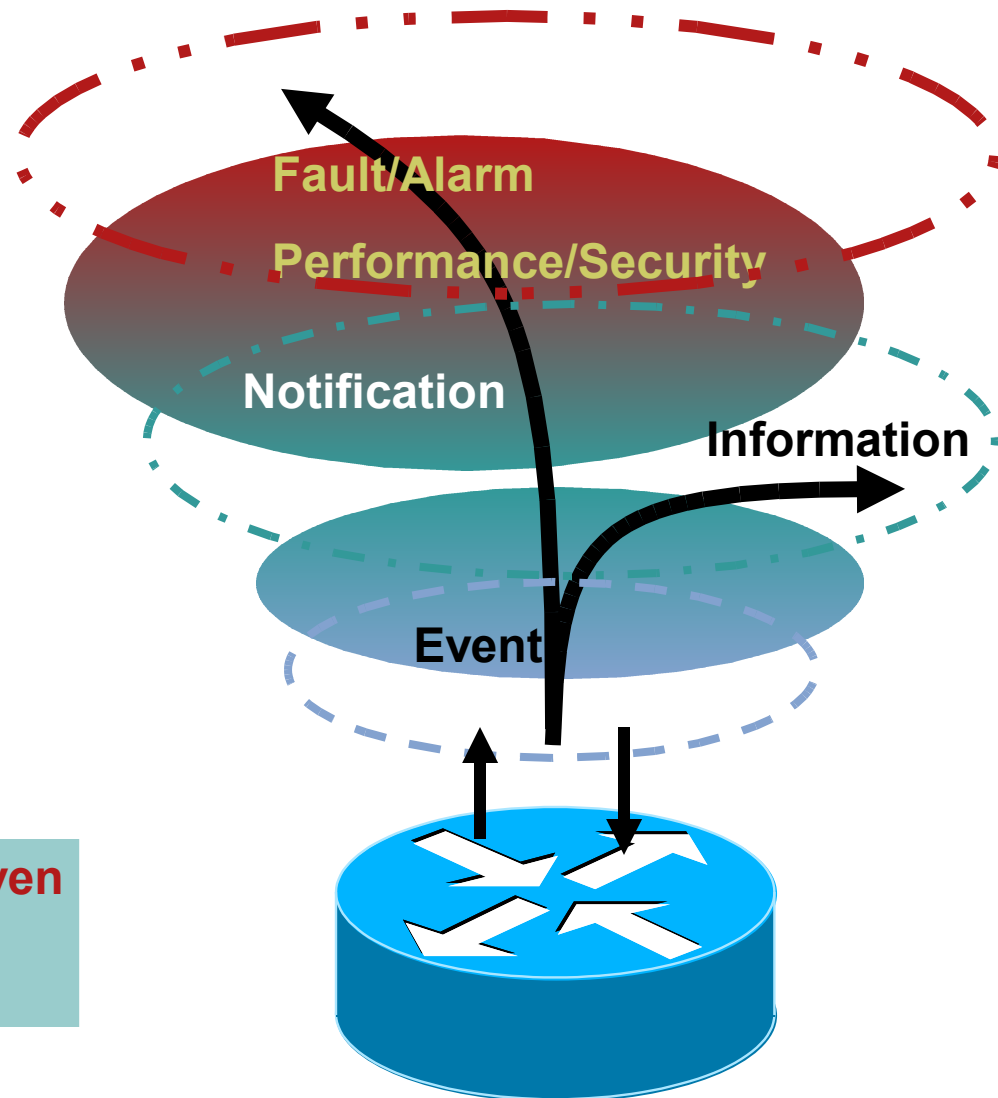
- Event definition
- Event transport
- Event processing
- Business-driven events

# Positioning

- Layered event process architecture
  - Issuing events
  - Processing events
    - ? Performance
- Information bus
  - Publishing events
  - Subscribing to events
    - ? Access/ transport
- Towards autonomic event processing
  - Network smartness vs. network management

# Get the infrastructure behavior

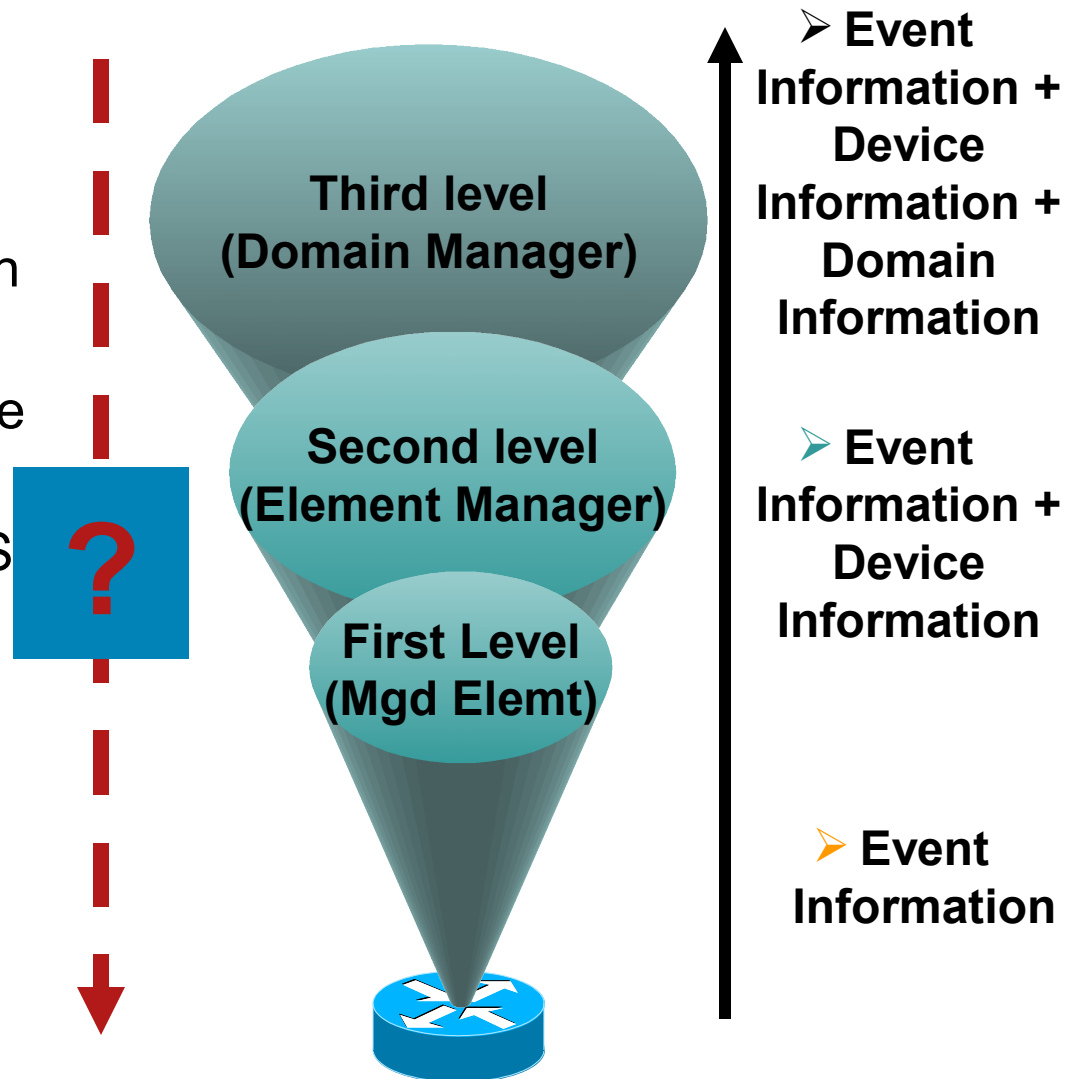
- Act (pre-emptive, proactive, reactive,...)
- Correlate (diagnostic, troubleshooting, impact, root cause, ...)
- Get status (push/poll)



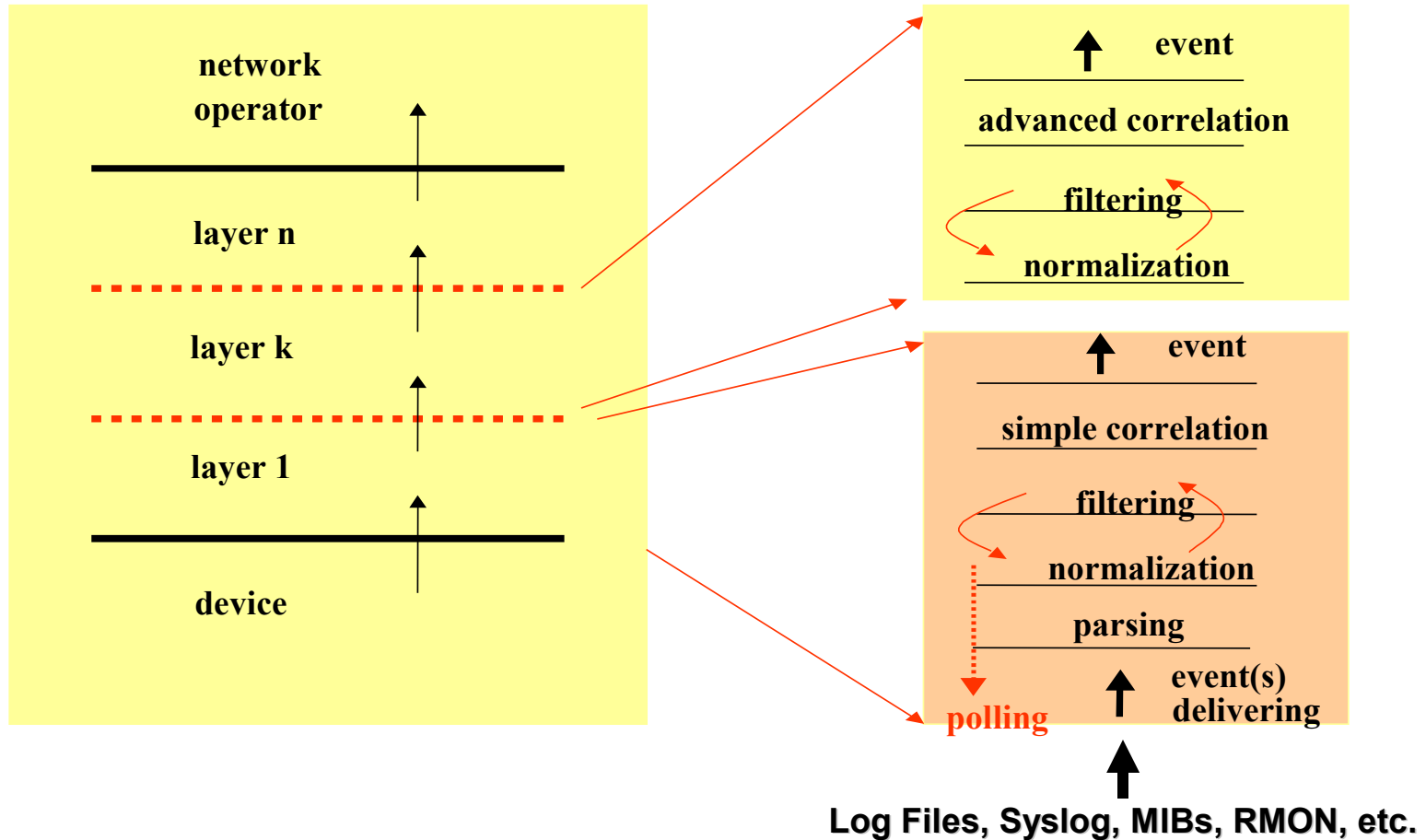
**All operations can be policy-driven**  
- top-down  
- bottom-up

# Bottom-up vs. Top-down

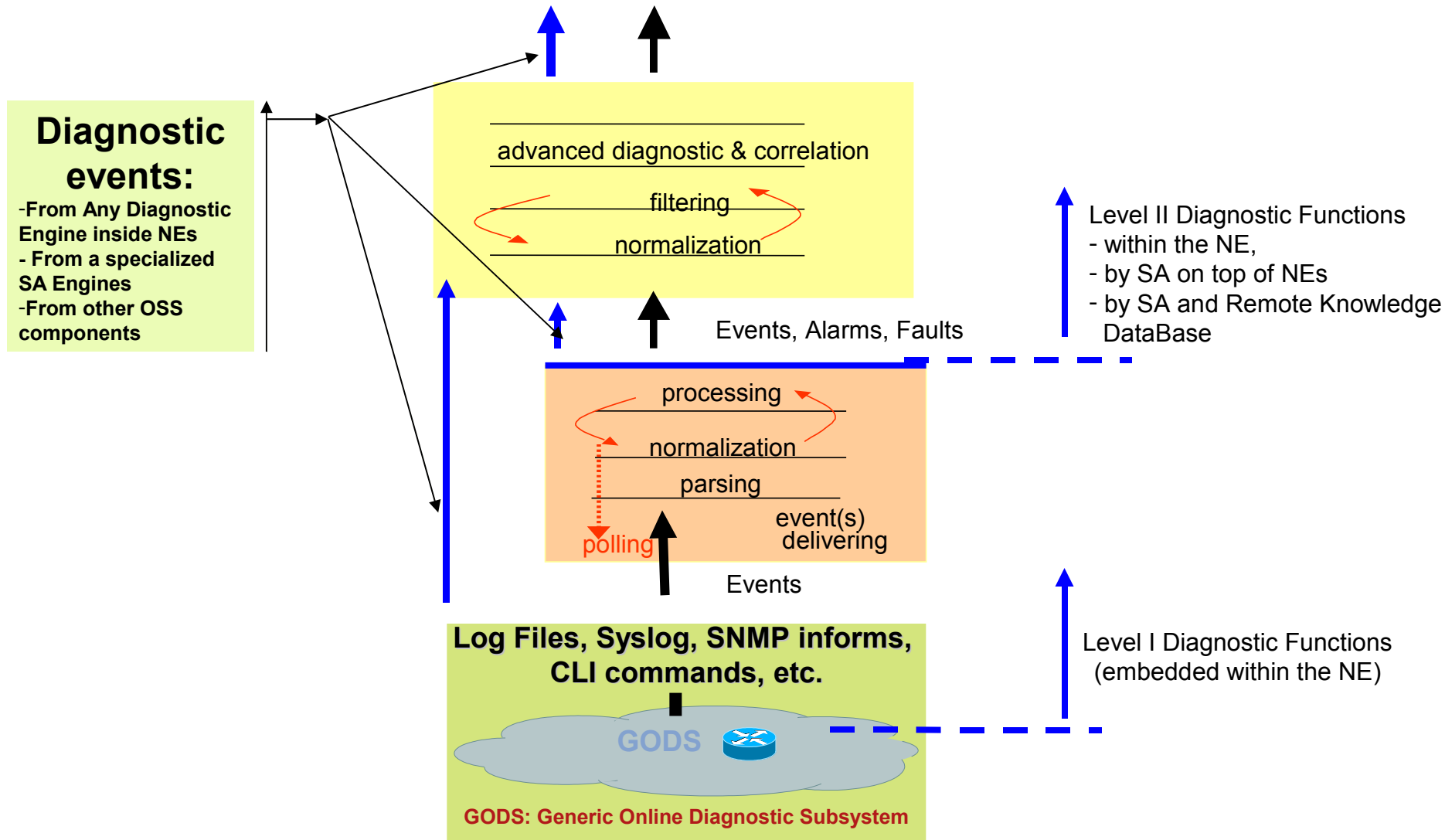
- Domain Manager enriches with domain information
- EMS enriches with multi-device information
- Notification Engine collects OS notifications



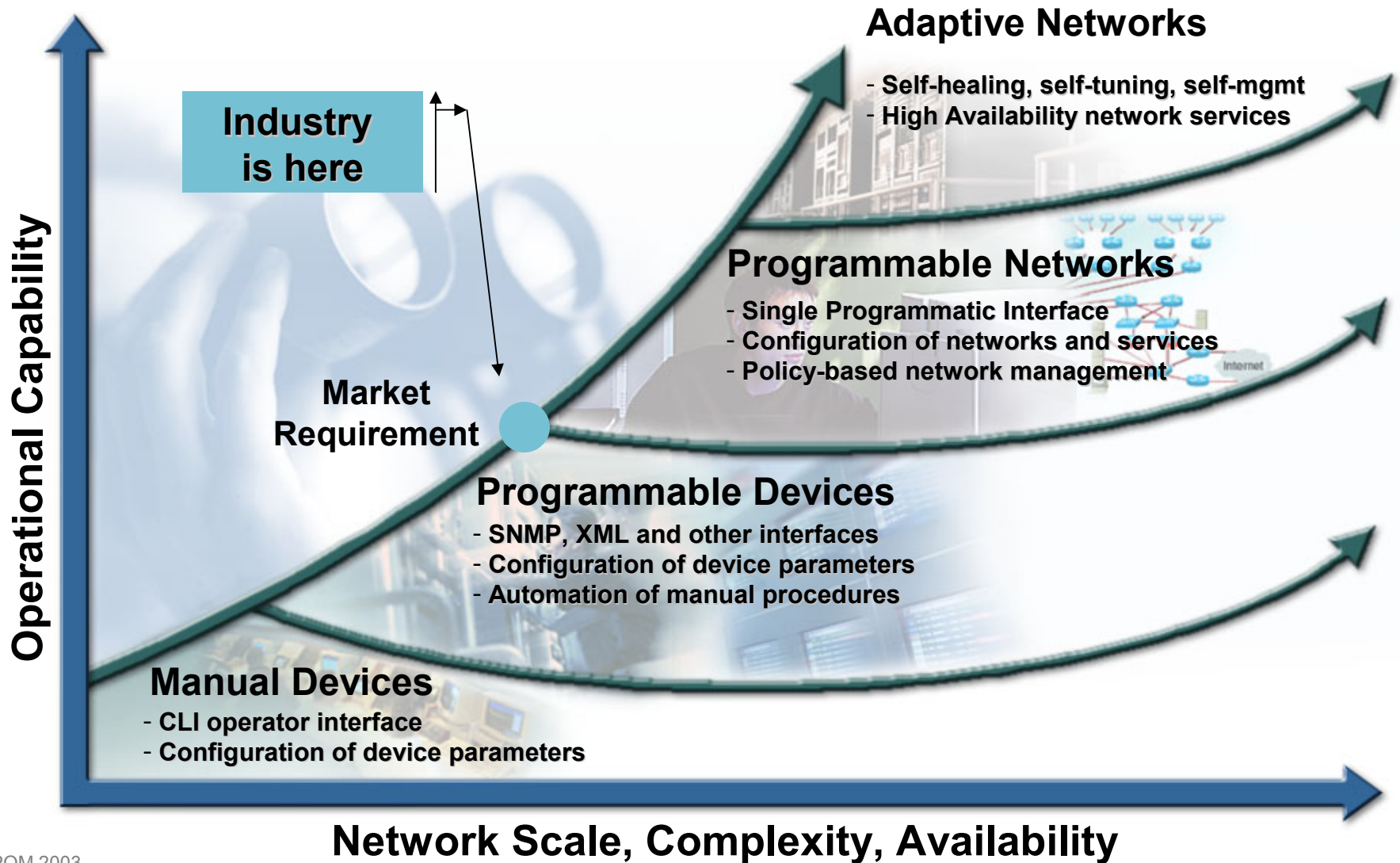
# A Layered Processing View



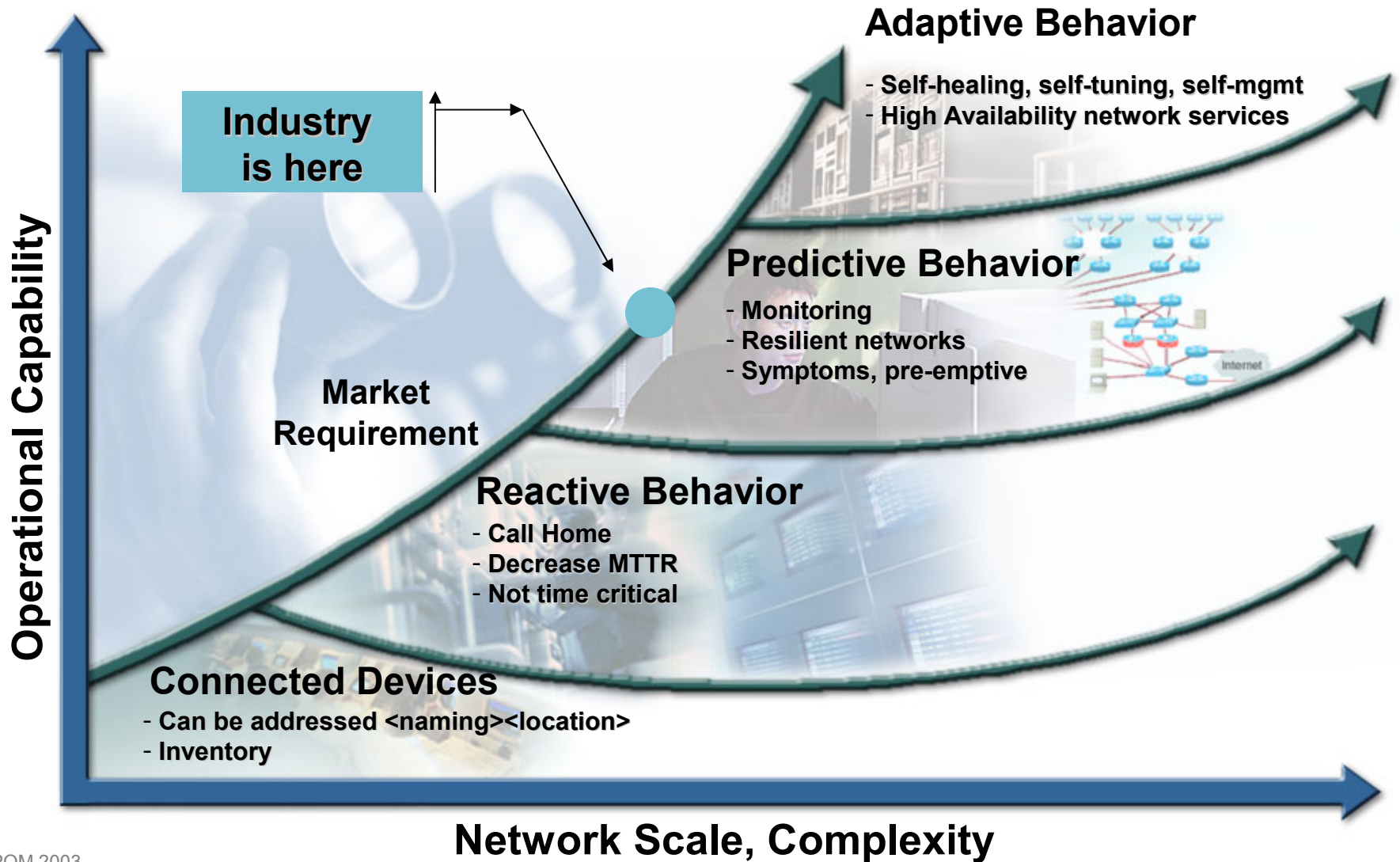
# Multi-level diagnostic



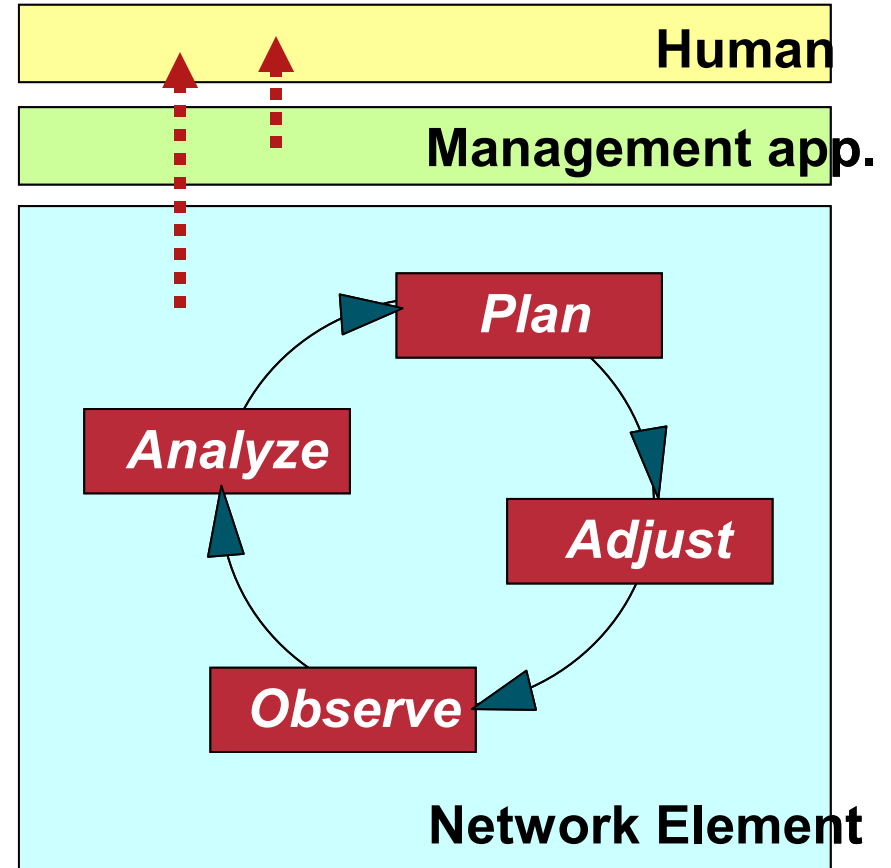
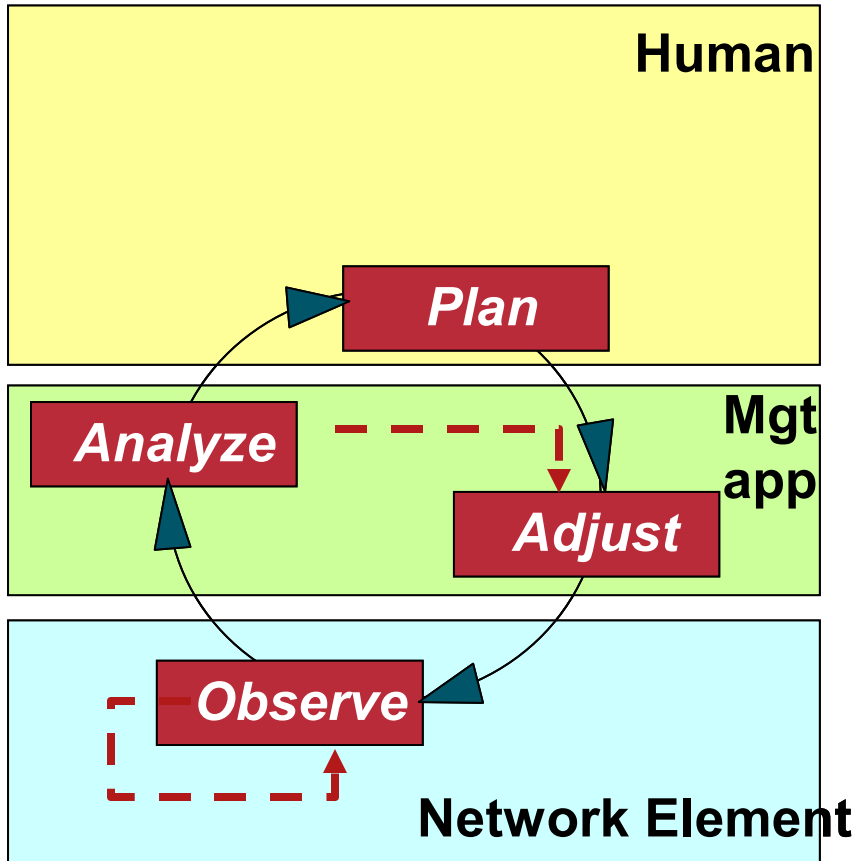
# Evolution of Network Manageability



# Evolution of Network Smartness



# Autonomic Components



(a) Typical management control loop (b) Closed management control loop in autonomous network

# Challenging Issues

Too Many

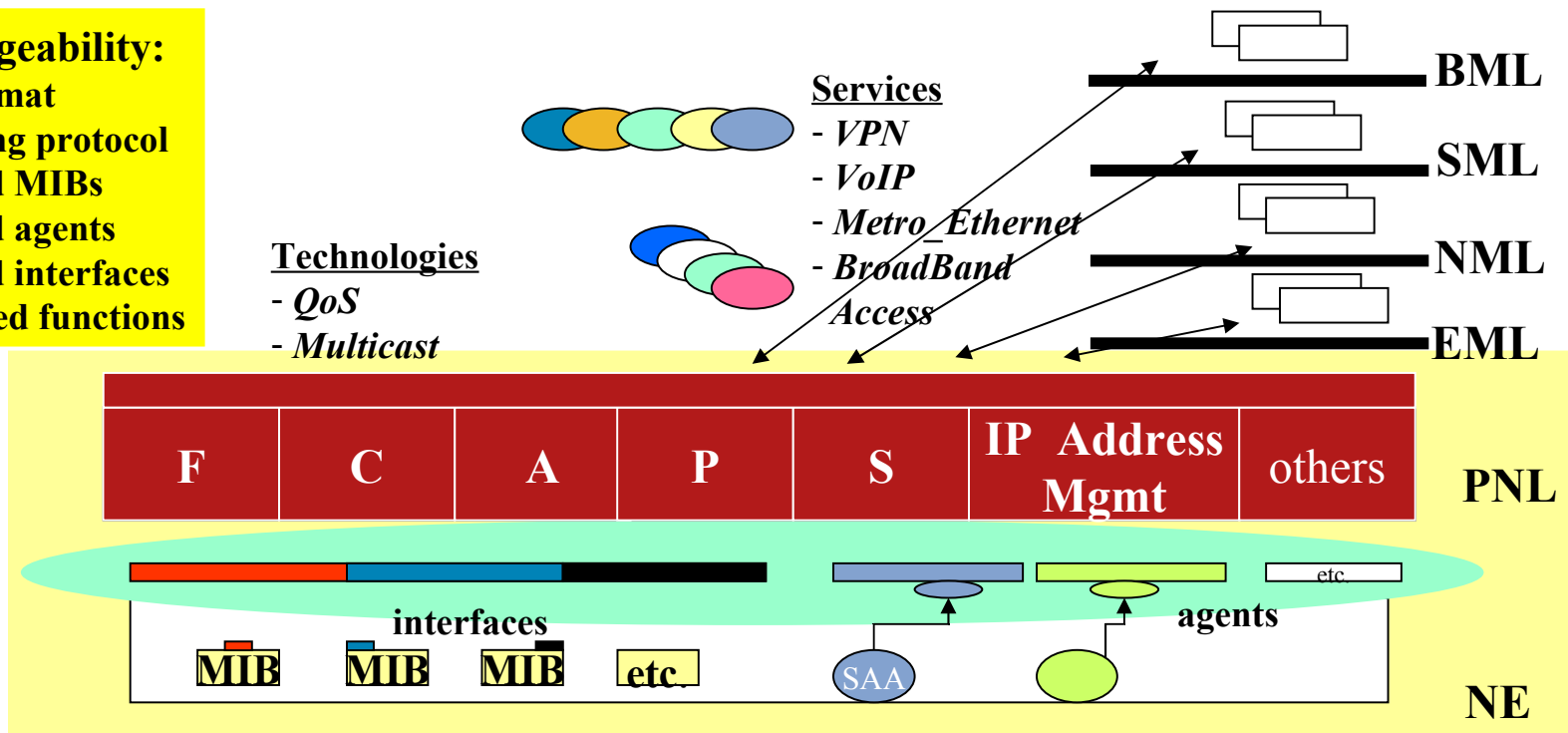
# Syntax Issues



- Various formats
- Myriad of conversions needed
- Lack of syntax control

## NE Manageability:

- ? data format
- ? conveying protocol
- ? required MIBs
- ? required agents
- ? required interfaces
- ? embedded functions



# Syslog Message “Body” Format in the IOS

**\* Sep 20 01:12:31: %SYS-5-CONFIG\_I: Configured from console by vty1 (144.254.9.79)**

**CONSOLE**

Timestamp

IOS Component

Severity

Mnemonic

Message-text

Timestamp from the server

**SERVER**

**Sep 20 01:07:00 router.cisco.com 571: Sep 20 01:12:31: %SYS-5-CONFIG\_I: Configured from console by vty1 (144.254.9.79)**

Router

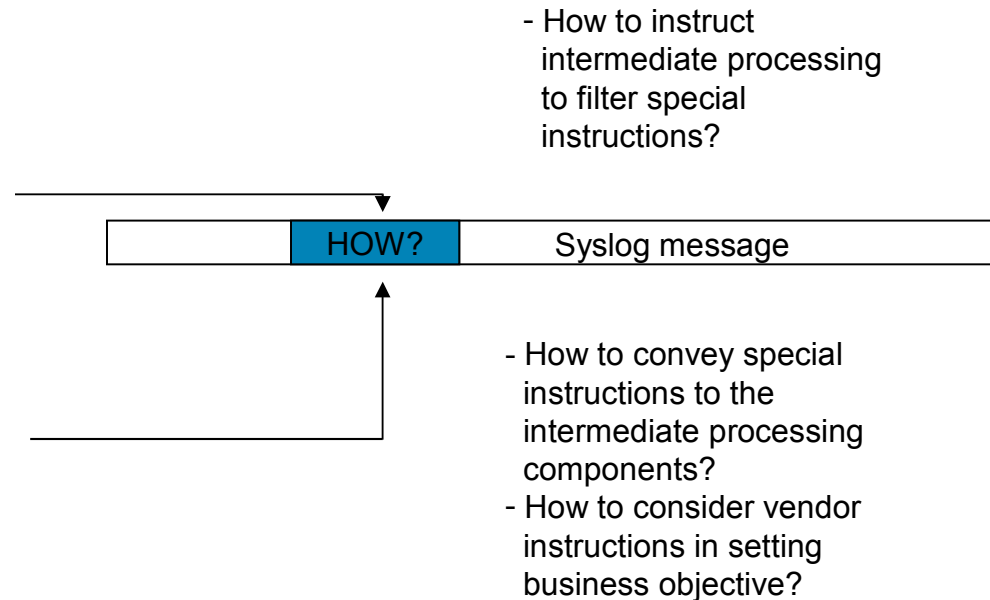
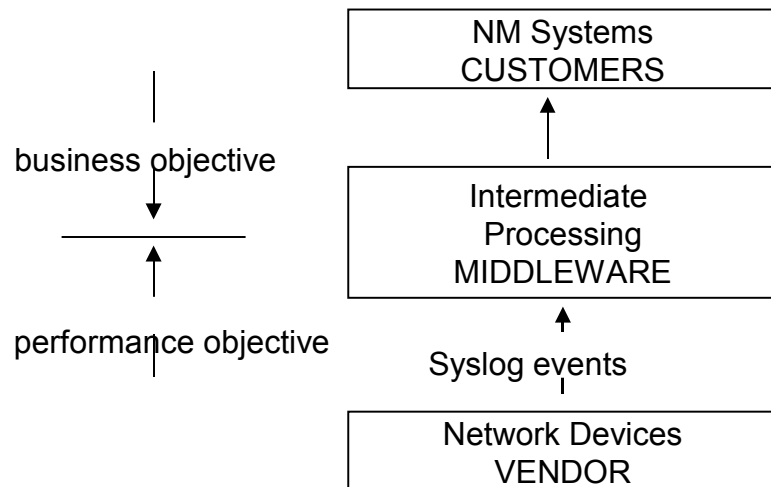
Timestamp from the router

- NTP is needed!
- Header:level can be different than Body:severity



# Semantic Issues

- Naming
- Context-defined
- Smart events



# XML Tagging is Not Enough

## % versus <XML>

: <<a><b>><c>>  
: (((a)(b)c))

1. <a>            <b>            <c>  
   ?            ?            ?

2. <<a> --- r1 -- <b>> -- r2 -- <c>  
          ?                    ?

e.g.,  
<a> -- Interface (? OID)  
<b> -- Port    (? OID)  
<c> -- Severity

?  
Tag table (??)  
Tag List:  
<name><semantics>

?  
Tag relationships

?  
Naming service  
required

- Despite the problems caused by its use:
  - – The messages don't have a standardized definition
  - – Priority is geared toward UNIX problems
  - – Priority is not used consistently
  - – Not reliable
  - – Not secure
- some key features, (i) ease of use for developers, (ii) familiarity, and (iii) ubiquity makes it a workable solution.

# Timestamps issues

- Format
- Clock-free event sources
- Sources-destination timestamps
- Delay tolerant networks
- Localizing processing
  - Local synchronization
  - Wide synchronization
- Reliable timestamps

# Adding Security to Event Transport

- Entity authentication
- Message Authentication
- Privacy
- Data integrity
- Signatures

# Putting an End to Unreliability

- Reliable transport mechanism
- Partially reliable transport [weak link]
- ?
  - event itself [seq numbers]-based
  - window-based
  - context-based

# Example: Syslog

[ field1 ] % [ field2 ] % [ severity ] % [ priority ] % [ mnemonic ] % [ free form field ]

## Well identified fields

[timestamps]

[facility ]

[severity ]

[priority]

[mnemonic]

## Free form field (the richest in semantic)

[..English plain text..]

## Field separator

%

## Issues

- Number of fields varies
- Value space of the fields is not uniform/standardized
- Semantic of timestamps is not uniform/or not defined
- Mnemonic is not modeled
  
- The English text is only humanly readable/useful
  
- Automation is difficult due to the “natural language processing” needs

# Things started to get fixed

- Syslog, SNMP/MIB: IETF
- Adaptive message format: IBM/Cisco
- Intrusion detection format: IETF
- Anomaly report format: OASIS
- Incident handling format: IETF
  
- NGN management : ITU-T [Focus group]

# Acting in advance, aka anticipating

# NOC, Systems

- ~ 7% events are considered
- No rules
- ? 93% ?
  
- Acting mode
  - Reactive
  - Real-time
  - Proactive
  - More-than proactive

# Management systems

- Non-autonomic systems

  - Monitor/report, off-line [reactive, Call petre@1-302...

  - Monitor/on-line control [ctrl protocols]

  - Monitor/on-line management [streaming processing ]

- Autonomic systems

  - Who is doing what?

  - What is the status?

  - Where the conflicts are?

  - Who solve the conflicts?

# Proactive vs. Anticipative

- Proactive: something happened  
performance decreasing  
QoE  
Intrusion attempts
- Anticipative: nothing happened yet, but  
Microsoft ‘paper clip asking...’  
People: intuition, imagination  
Systems: prediction, anticipation

# Open issues

- Non-system related knowledge
- Context-based knowledge representation languages
- Methodology to design autonomic components
- Trustiness in anticipation // reputation
- Types of system events to be considered
  
- ! 95% events are Syslog , very little % are “notifications”

**Thank you!**

# Q and A

