EPFL

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Web-Based Network Management:
From Pull to Push

University of Twente
January 22, 1999

Jean-Philippe Martin-Flatin
Swiss Federal Institute of Technology, Lausanne
Institute for computer Communications and Applications
ICA
martin-flatin@epfl.ch
http://icawww.epfl.ch/~jpmf/

Outline

- Pre-Web Network Management Platforms
- Problems
- Web-Based Management
- Pull Model
- Push Model
- Future Research
- Q&A

Network Management Platforms

- OSI:
  - 5 SMAs (FCAPS)
- IP:
  - 3 core functions (mandatory):
    - network monitoring
    - data collection
    - notification handling
  - other functions (optional): configuration, ACLs, billing...
  - SNMP management framework (SNMPv1, SNMPv2c, SNMPv3)
  - Examples: HP OpenView, Cabletron Spectrum, IBM Netview, Sun Solstice...

IP NMPs: 3 Core Functions

- Network monitoring:
  - detect faults in network devices and links:
    - ➠ reactive w.r.t. network faults
    - ➠ proactive w.r.t. complaints from users/customers
- Data collection:
  - gather data to build daily, weekly and monthly reports:
    - ➠ proactive in the longer term
- Notification handling:
  - quick
  - react to events generated by agents (SNMP notifications)
  - react to events generated by the manager (event correlator)

Regular Management

- Ongoing network monitoring or data collection
- Automated
- 2 modes:
  - attended mode: operators gazing at GUIs (red-icon syndrome)
  - unattended mode: automated correlation, alarms (pager, email, telephone, siren...)
- Medium-sized to large networks

Ad Hoc Management

- Troubleshooting or configuration
- Manual
- Always attended mode: administrators and/or operators
- All networks
- Replaces regular management in small networks

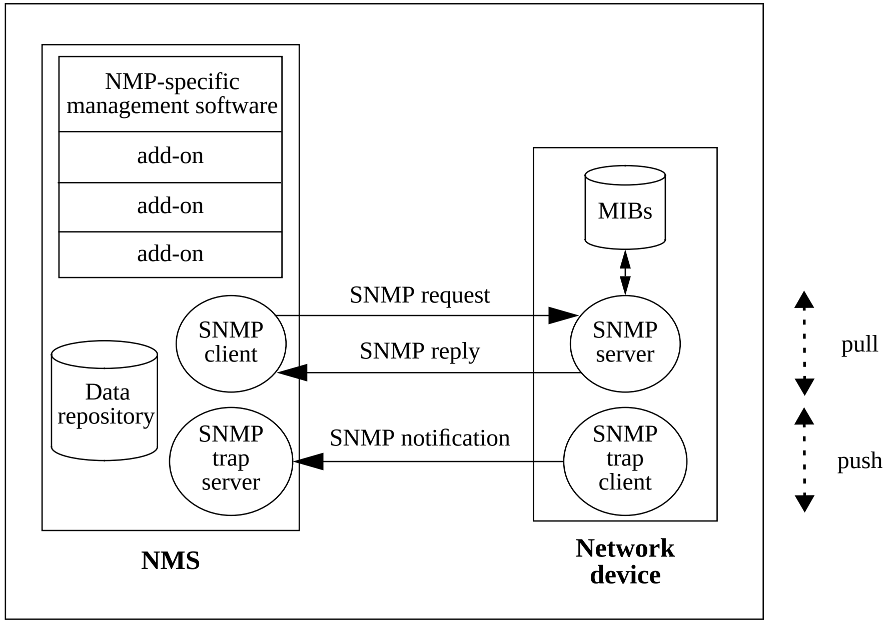
Pre-Web NMPs:
SNMP Management Framework

- Manager/agent paradigm
- Polling for data collection & network monitoring
- Unsolicited push for notification delivery
- SNMP communication protocol
- SMIv2 (ASN.1)
- BER encoding
- MIBs (generic, vendor-specific)
- ...

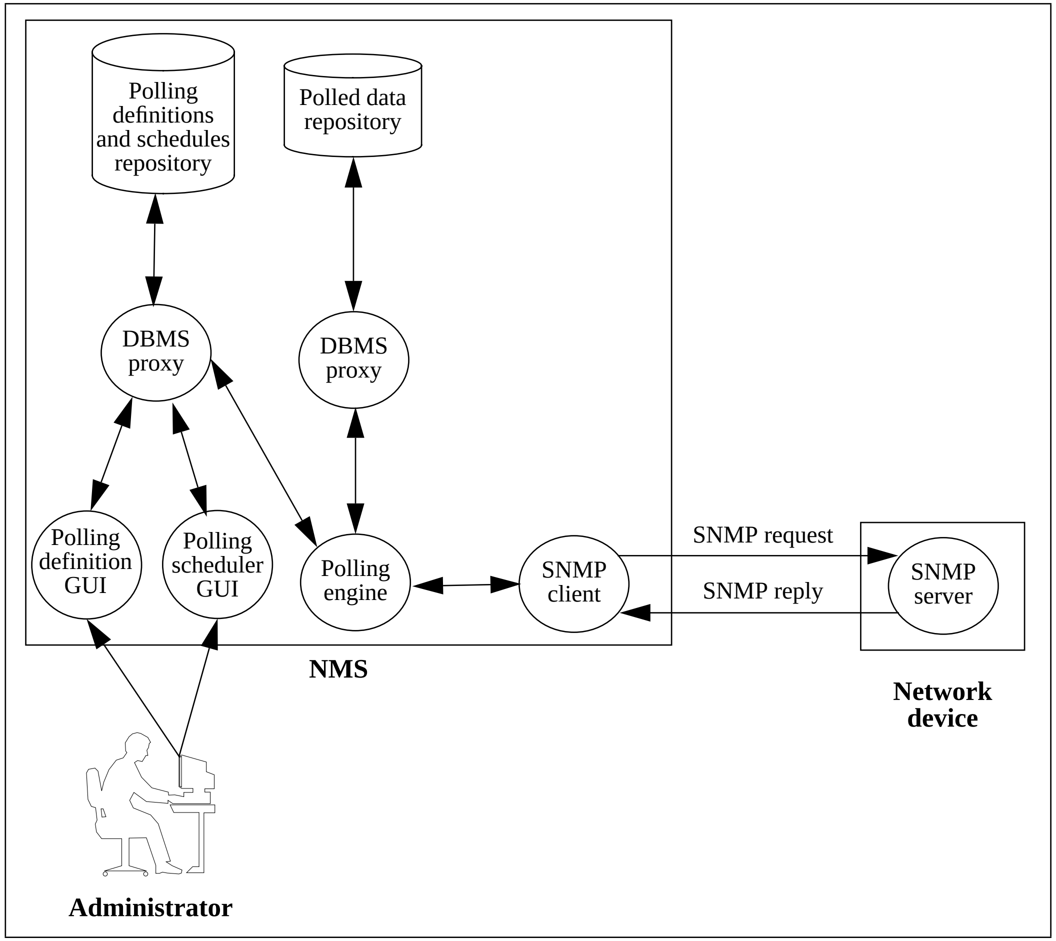
Pre-Web NMPs: Market Evolution

- Once upon a time, there were open systems [...]:
  - generic network equipment
  - generic management
- Market segmentation
- From generic MIBs to vendor-specific MIBs
- From generic management GUIs to vendor-specific management GUIs (add-ons)
- For customers, openness guaranteed by:
  - SNMP management framework
  - SNMP protocol

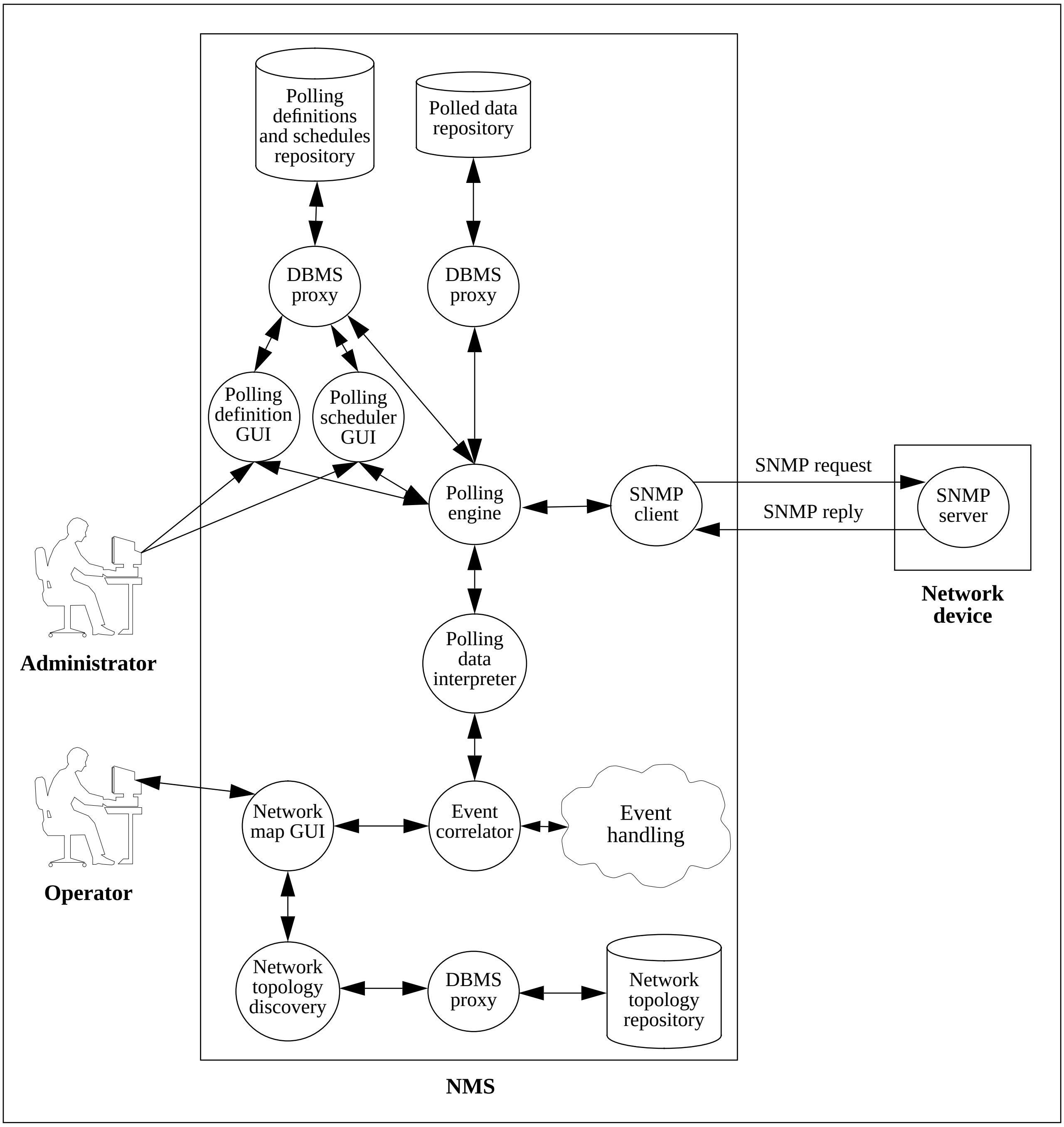
A Simple Model of Pre-Web NMPs



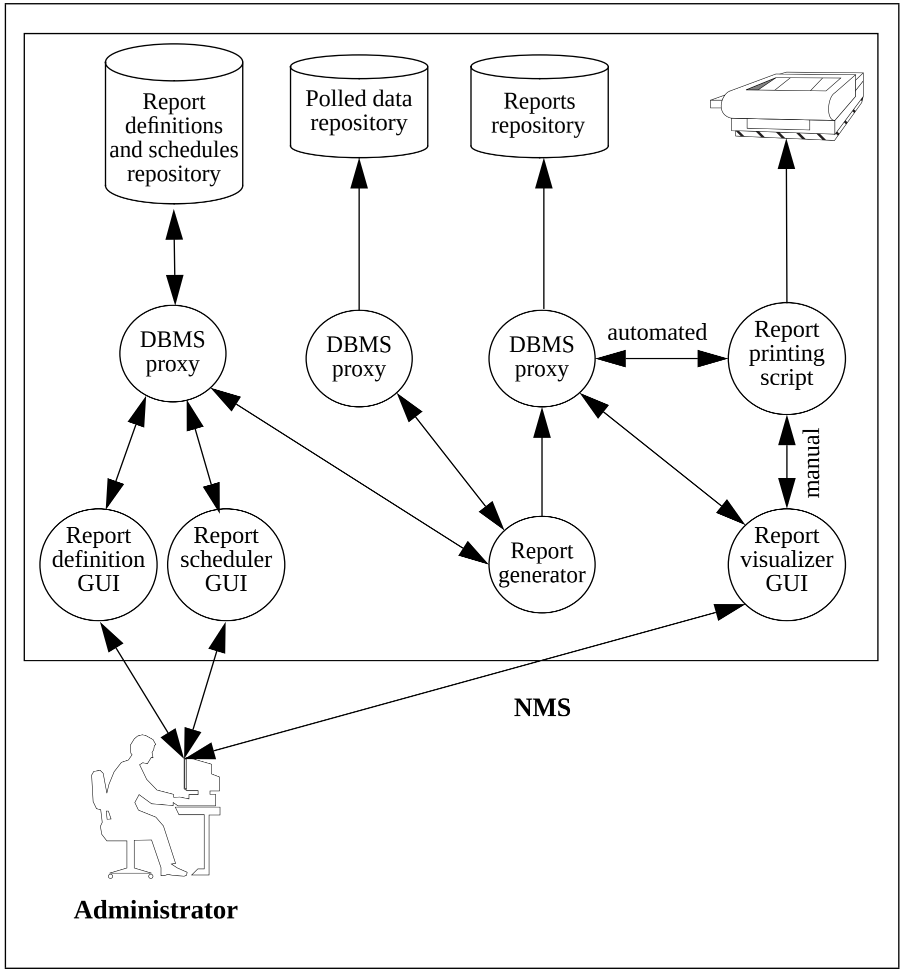
Data Collection



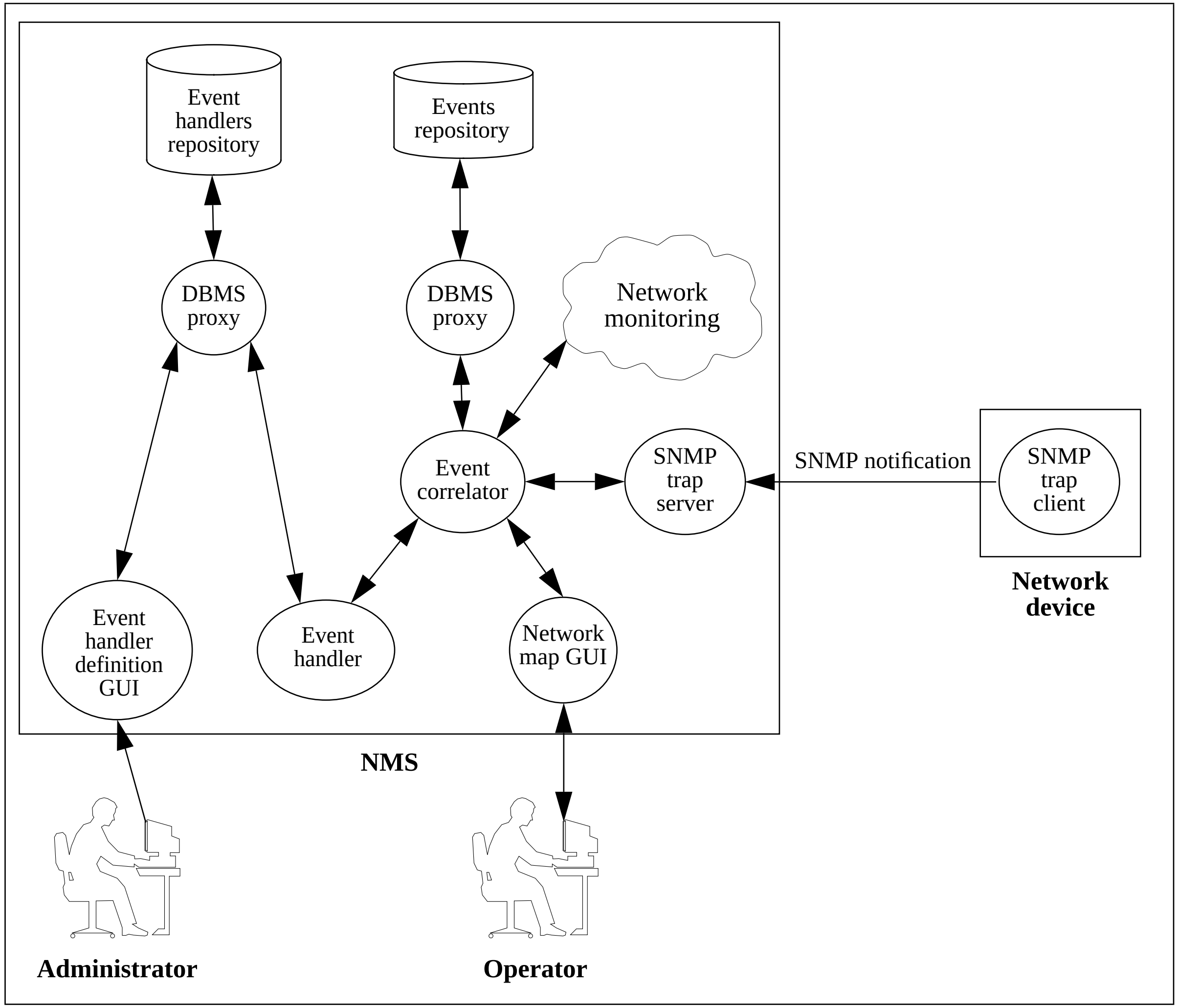
Network Monitoring



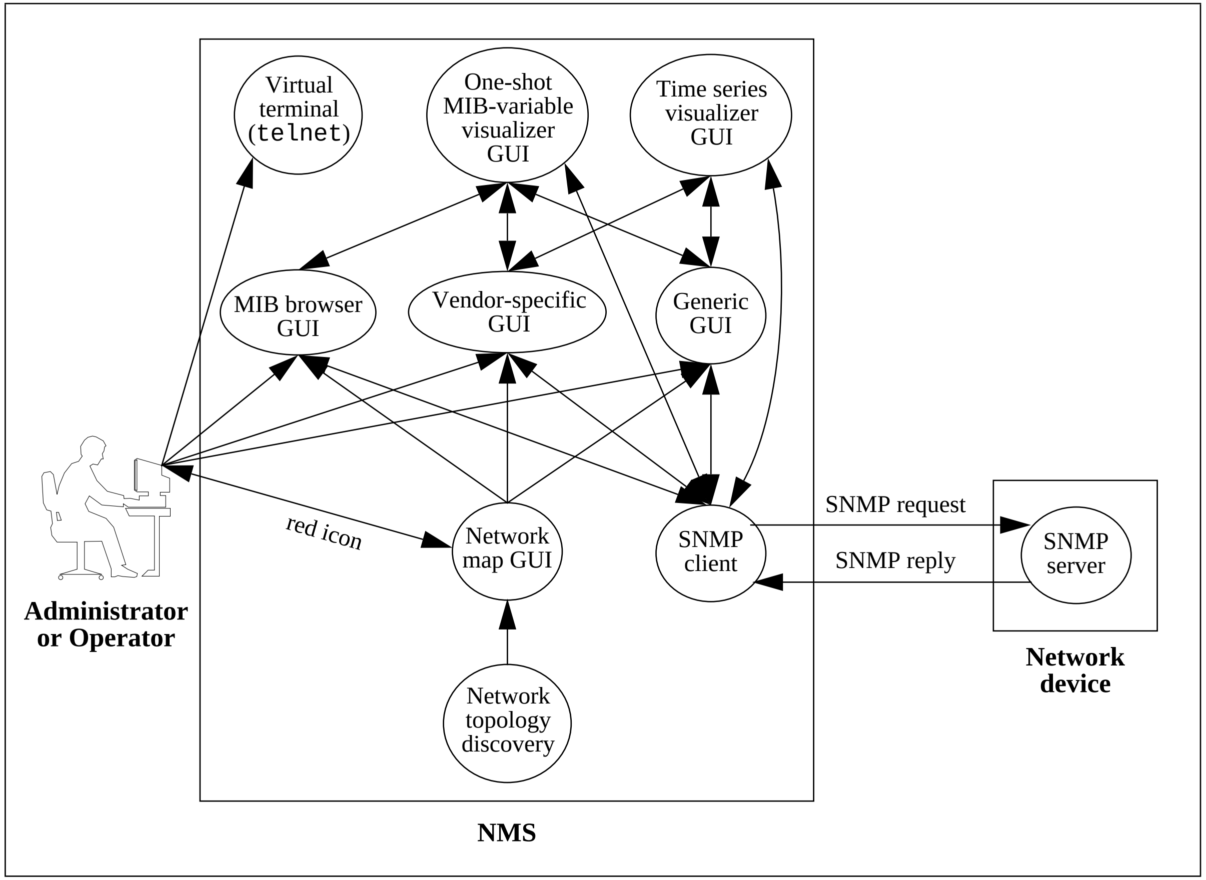
Report Generation



Notification Delivery and Event Handling



Ad Hoc Management


Virtual terminal (telnet)
One-shot MIB-variable visualizer GUI
Time series visualizer GUI
MIB browser GUI
Vendor-specific GUI
Generic GUI
red icon
Network map GUI
SNMP client
SNMP request
SNMP reply
SNMP server
Network device
Network topology discovery
Administrator or Operator
NMS

Problems with Pre-Web NMPs (1/2)

- NMP vendors:
  - no problem, it’s big bucks time!
- Customers:
  - NMPs are too expensive (hardware and software):
    - ➠ dedicated hardware for network management
  - limited support for third-party RDBMSs
  - need for Unix expertise to maintain existing platforms:
    - ➠ cost to migrate to Windows is too high
- Network equipment vendors:
  - the support of many device-specific add-ons for many NMPs and many OSs is too expensive

Problems with Pre-Web NMPs (2/2)

- Customers and network equipment vendors:
  - poor time-to-market for add-ons:
    - ➠ several months after hardware release if large market share
    - ➠ never if small market share:
      - startup companies need to resort to separate NMPs
  - versioning:
    - ➠ version mismatch between the add-on (NMP) and the MIBs (devices) while a vendor-specific MIB is gradually upgraded in a network:
      - update the NMP manually, device by device (no MIB-discovery protocol)
      - do not use new features of a MIB until all devices have been upgraded

Software Engineering Problems with Pre-Web Network Management (1/2)

- Protocol efficiency:
  - poor efficiency of BER encoding [Mitra 1994] [Neufeld and Vuong 1992]:
    - addressed by PER (Packed Encoding Rules) in OSI
    - SMIv1 and SMIv2 mandate BER encoding for all SNMP frameworks
  - poor efficiency of SNMP:
    - no efficient table retrieval mechanism --> repeated message exchanges
    - in varbind lists, OIDs take much more space than values

Software Engineering Problems with Pre-Web Network Management (2/2)

- Security:
  - lack of secure SNMP `get` or `set` in SNMPv1 and SNMPv2c
  - SNMPv3: just released, still to show its acceptance in the field
  - VPNs: need expensive encryption hardware to manage remote subsidiaries
  - firewalls: UDP relays are complex to set up and maintain [Chapman and Zwicky 1995]
- TCP vs. UDP
  - in theory, both OK to transport SNMP; in practice, only UDP
  - important SNMP notif. are lost for silly reasons (e.g. buffer overflow)
  - some mgmt data is more important than user data (e.g. heartbeats)

Web-Based Management

- What is it?
- What can Web technologies bring to pre-Web NMPs?
- Early solutions: HTML pages
- Today: pull model
- Tomorrow: push model
- Near future: mobile code (mgmt tasks delegated to agents)

What is Web-Based Management? (1/2)

- Marketing answer = WBEM
    - today, CIM schema but little Web [Thompson 1998]
- Proposed technical answer = integrated network, systems and service management based on Web technologies:
    - HTML forms
    - Java applets, servlets and applications
    - JDBC
    - Java RMI and Object Serialization
    - Java IDL
    - ...
- This talk = Web-Based Network Management

What is Web-Based Management? (2/2)

- Distribution in Java:
  - HTTP
  - sockets
  - RMI
  - Java IDL (CORBA)
    - ➠ telecoms = yes
    - ➠ Internet = no
- Typically:
  - applet to servlet: HTTP or sockets
  - applet to Java application: sockets or RMI

What can Web technologies Bring to Pre-Web NMPs?

- Get rid of the NMP
- Reduce network overhead of management data
- Reduce development costs of add-ons
- Reduce time-to-market of add-ons
- Put small and large equipment vendors in fair competition
- Simplify management of remote subsidiaries across a firewall
- Improve support for third-party RDBMSs
- ...

HTML Pages

- Secondary tasks:
  - automate and standardize problem reporting (helpdesk)
  - put daily, weekly and monthly reports online (paper-free office policy)
  - online help for network troubleshooting:
    - administrators write symptom-driven HTML pages for operators, with pointers to online vendors’ documentation
    - user-friendly access to management scripts (Perl, Tcl/Tk) and programs (ping, traceroute, netstat)

HTML-Based CLI

- Mapping between URLs and command line interface:
  - e.g. on Cisco routers [Bruins 1996]:
    - `http://routername/exec/show/interface/ethernet0/`
    - `show interface ethernet0`
  - generated via HTML forms, or embedded in symptom-driven HTML pages

Pull vs. Push

- Newspaper metaphor:
  - buy it everyday from the same newsdealer
  - receive it everyday by postal mail
- Pull model:
  - request/response paradigm
  - data transfer initiated by the manager
  - e.g., data polling in pre-Web NMPs (network monitoring and data collection)
- Push model:
  - publish/subscribe/distribute paradigm
  - parallel and independent data transfers initiated by agents
  - e.g., SNMP notifications in pre-Web NMPs

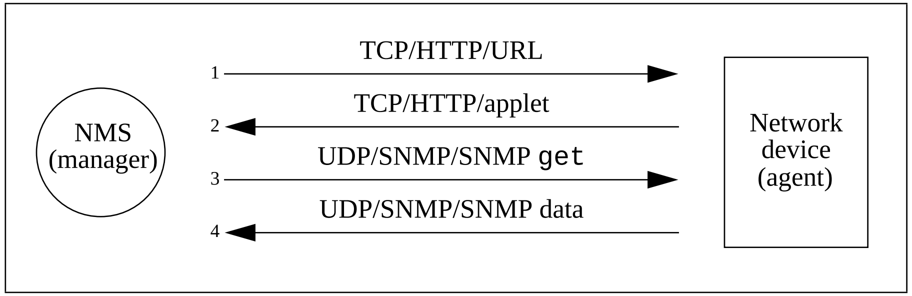
Pull Model

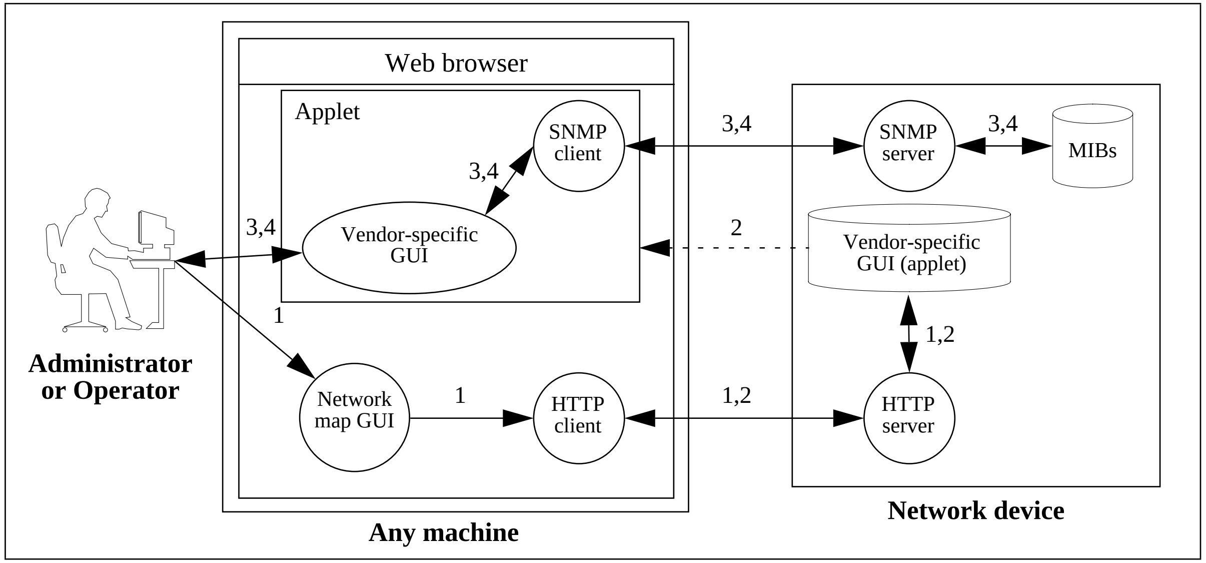
- Ad hoc management:
  - vendor-specific management GUIs coded as applets
    - ➠ HTTP together with SNMP
    - ➠ HTTP instead of SNMP
  - generic management GUIs coded as applets
- Regular management:
  - all GUIs coded as applets
  - data polling based on HTTP

HTTP Together With SNMP (1/3)

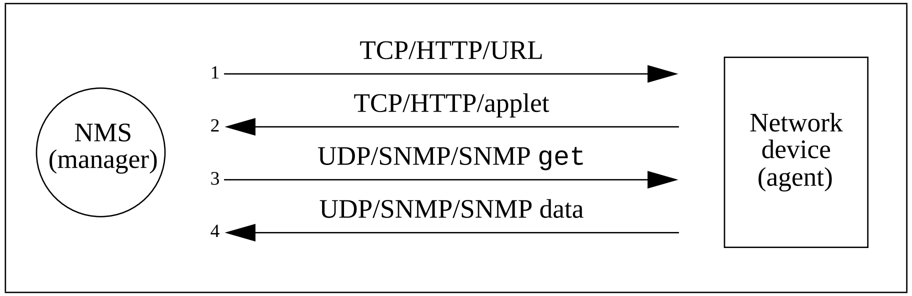
- [Bruins 1996]
- Device-specific management GUI (add-on) coded as an applet
- Management data transferred via SNMP
- Ad hoc management

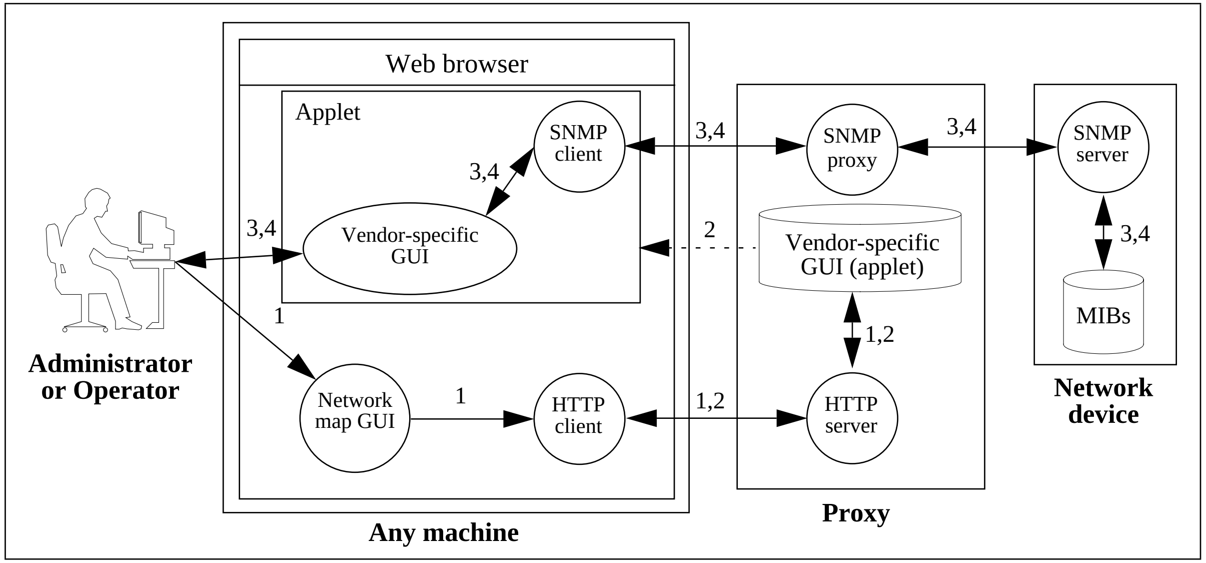
HTTP Together With SNMP (2/3)





HTTP Together With SNMP (3/3)

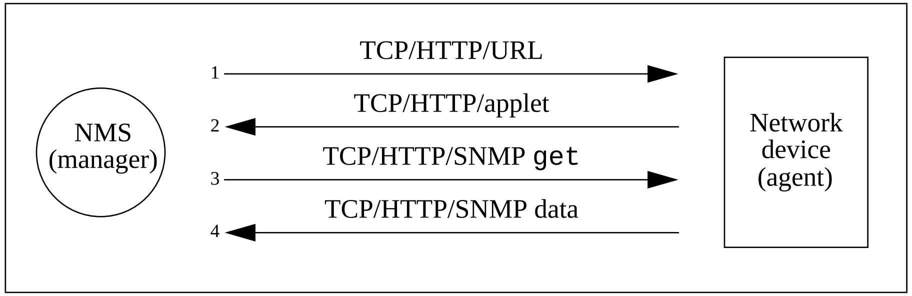


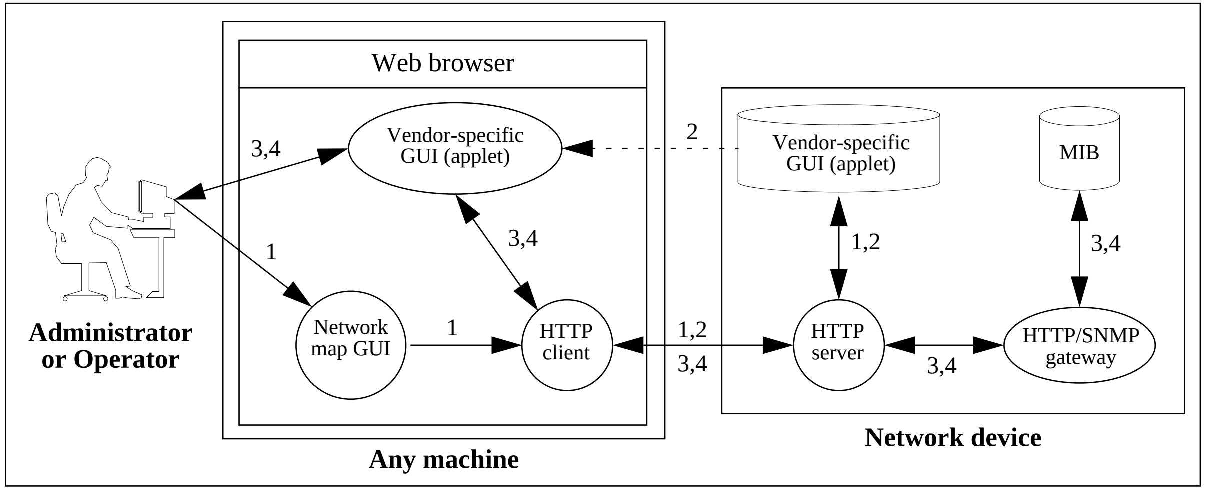


HTTP Instead of SNMP (1/3)

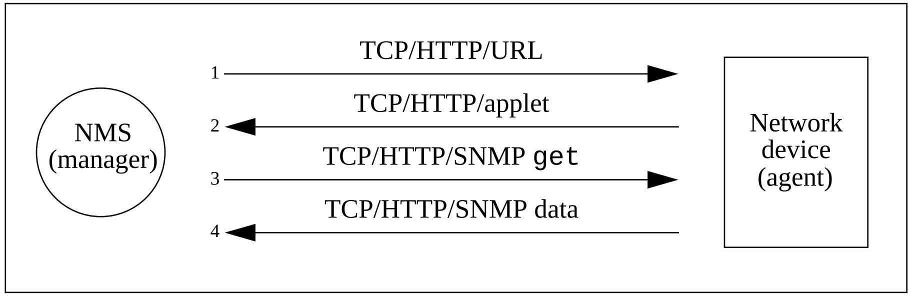
- [Wellens and Auerbach 1996]
- Device-specific management GUI (add-on) coded as an applet
- Management data transferred via HTTP
- Ad hoc management

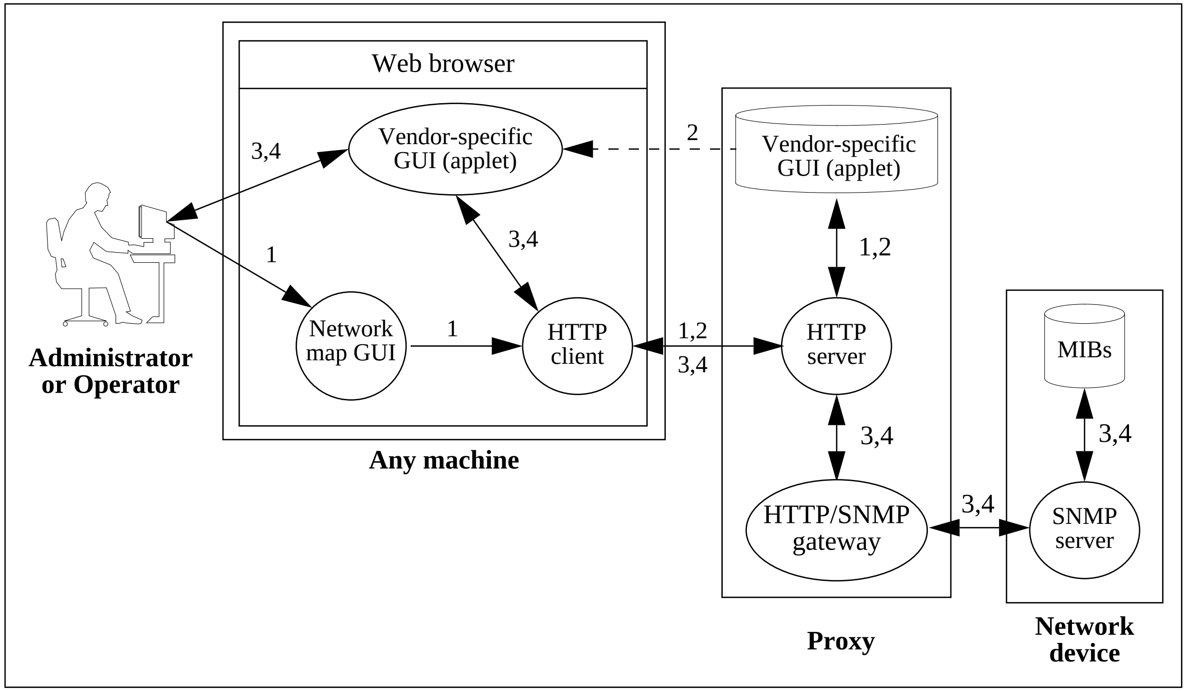
HTTP Instead of SNMP (2/3)



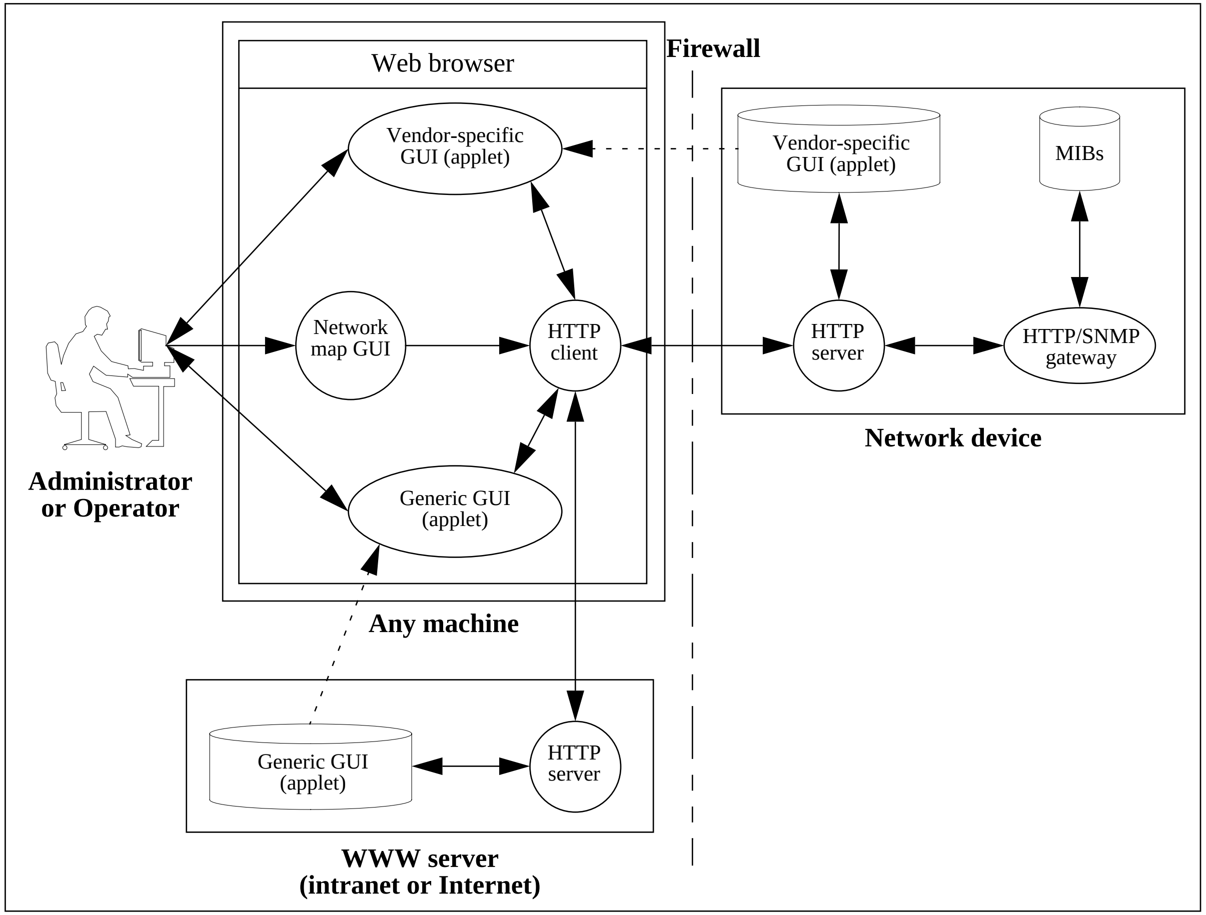


HTTP Instead of SNMP (3/3)





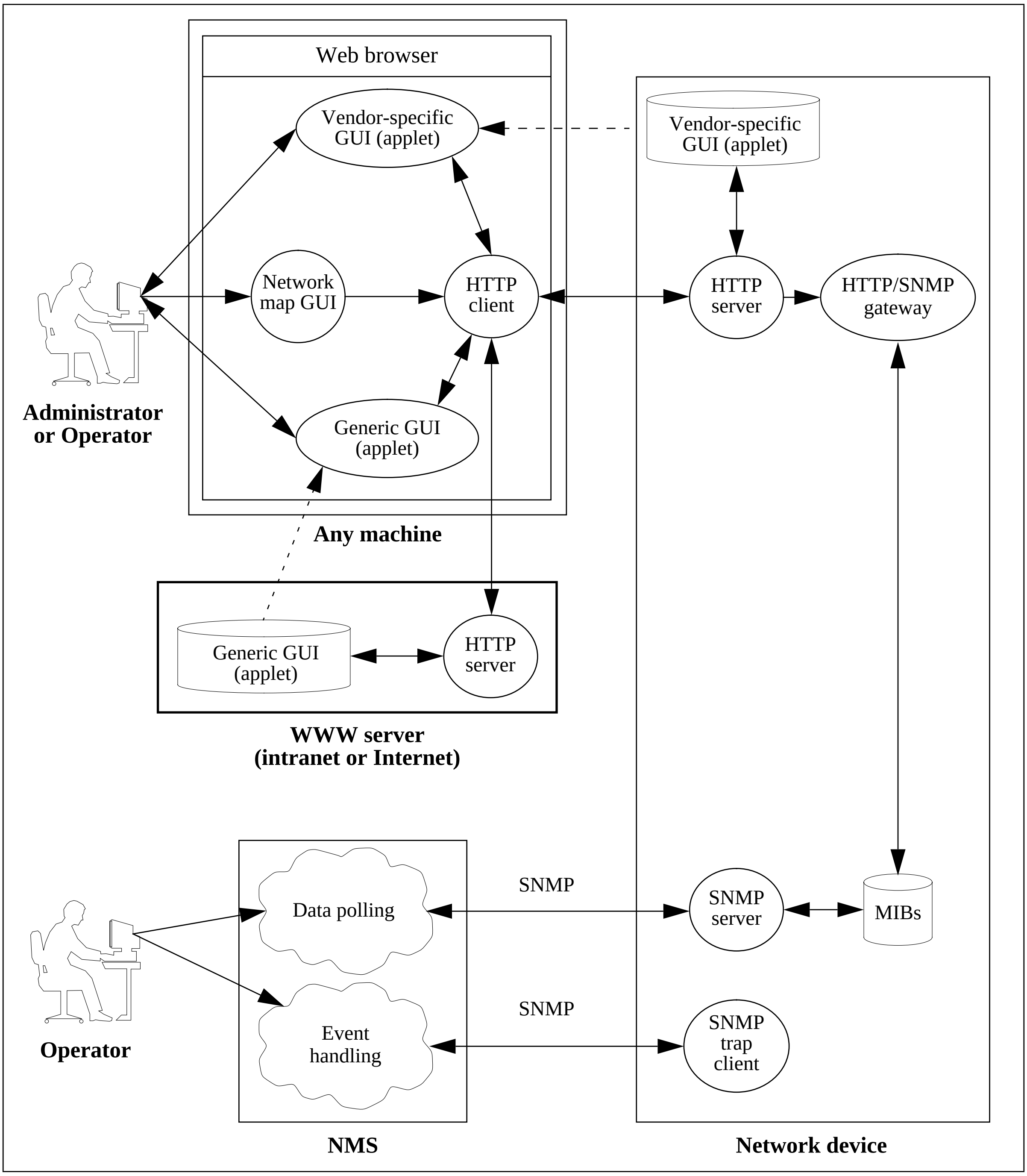
Generic GUIs Coded as Applets



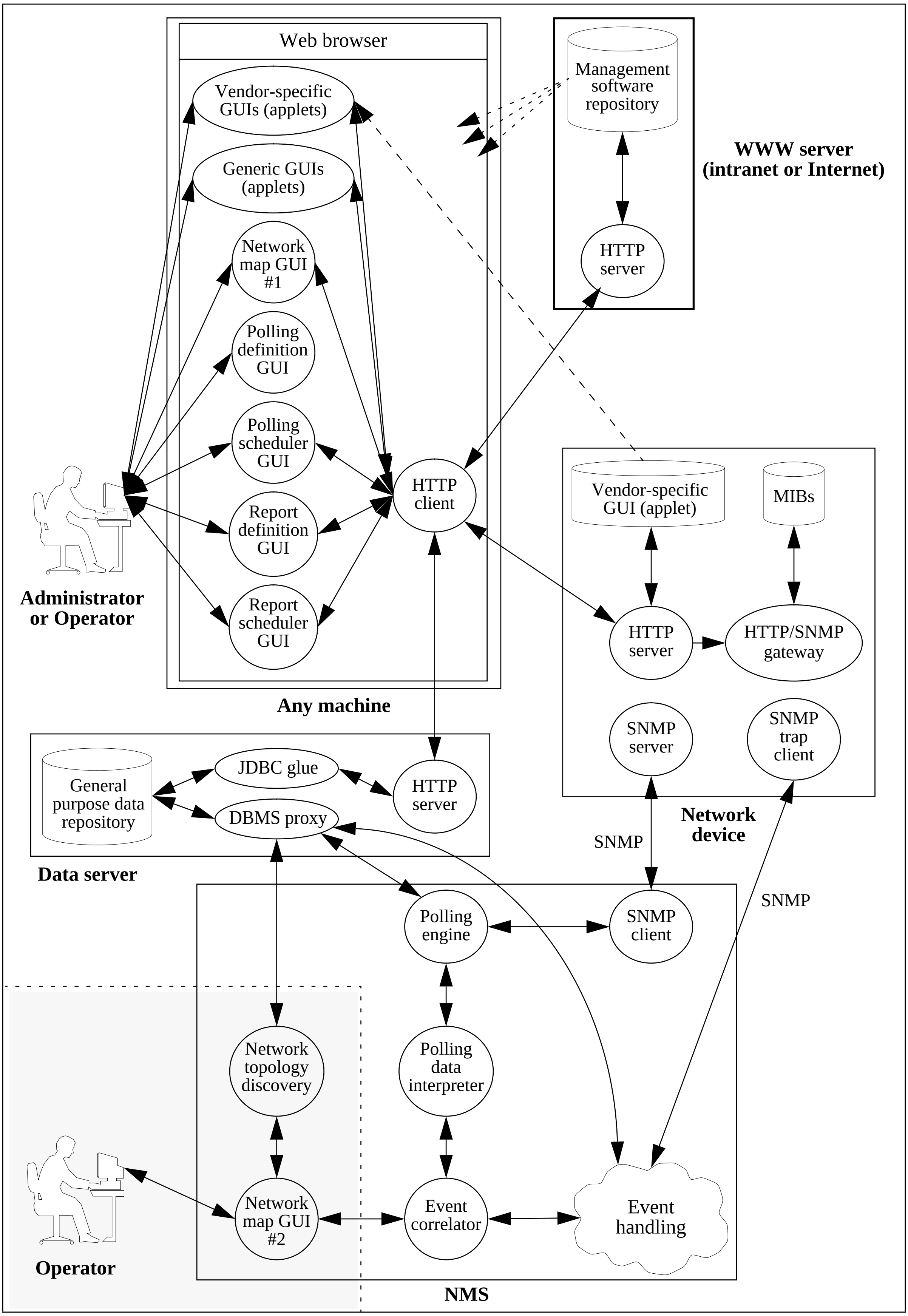
Pull Model

- Ad hoc management:
  - vendor-specific management GUIs coded as applets
    - HTTP together with SNMP
    - HTTP instead of SNMP
  - generic management GUIs coded as applets
- Regular management:
  - all GUIs coded as applets
  - data polling based on HTTP

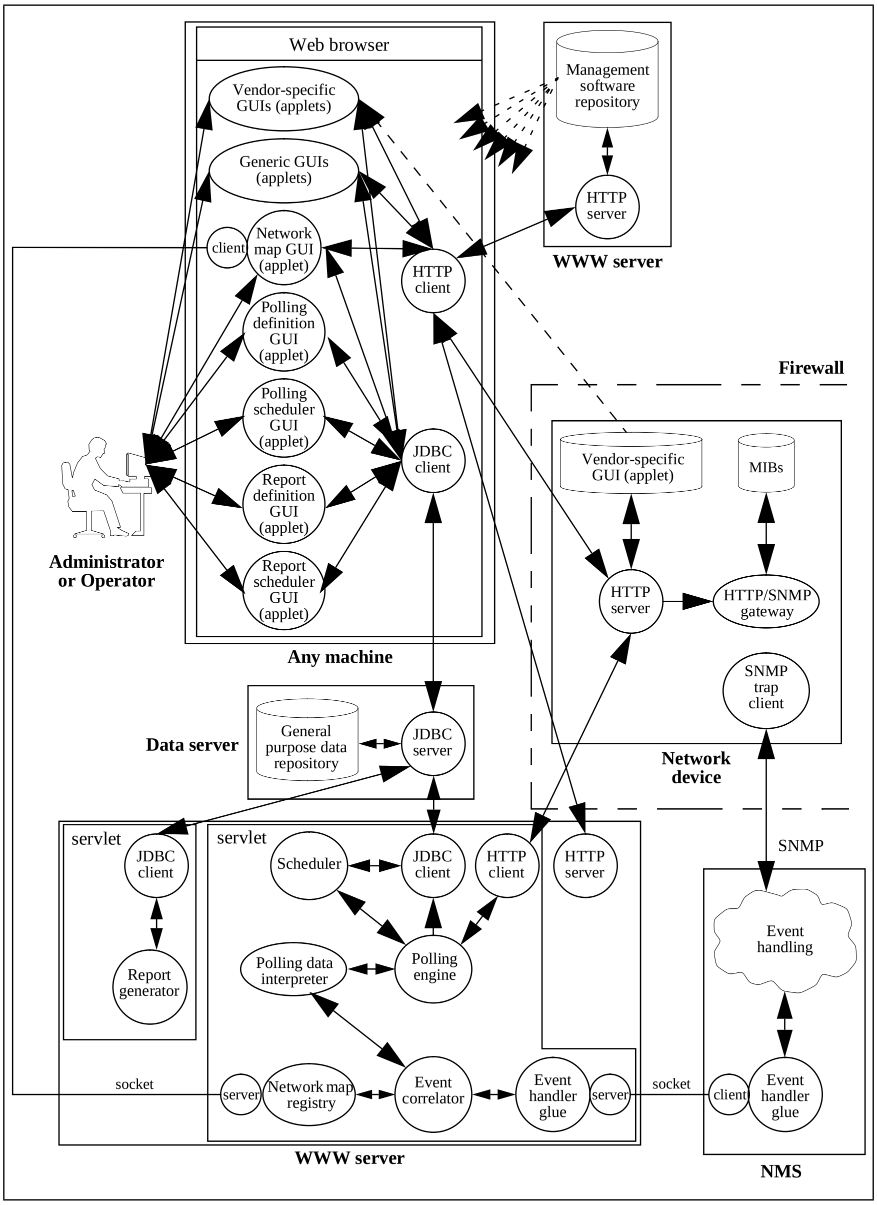
Web-Based Ad Hoc Mgmt & Pre-Web Regular Management



All GUIs Coded As Applets



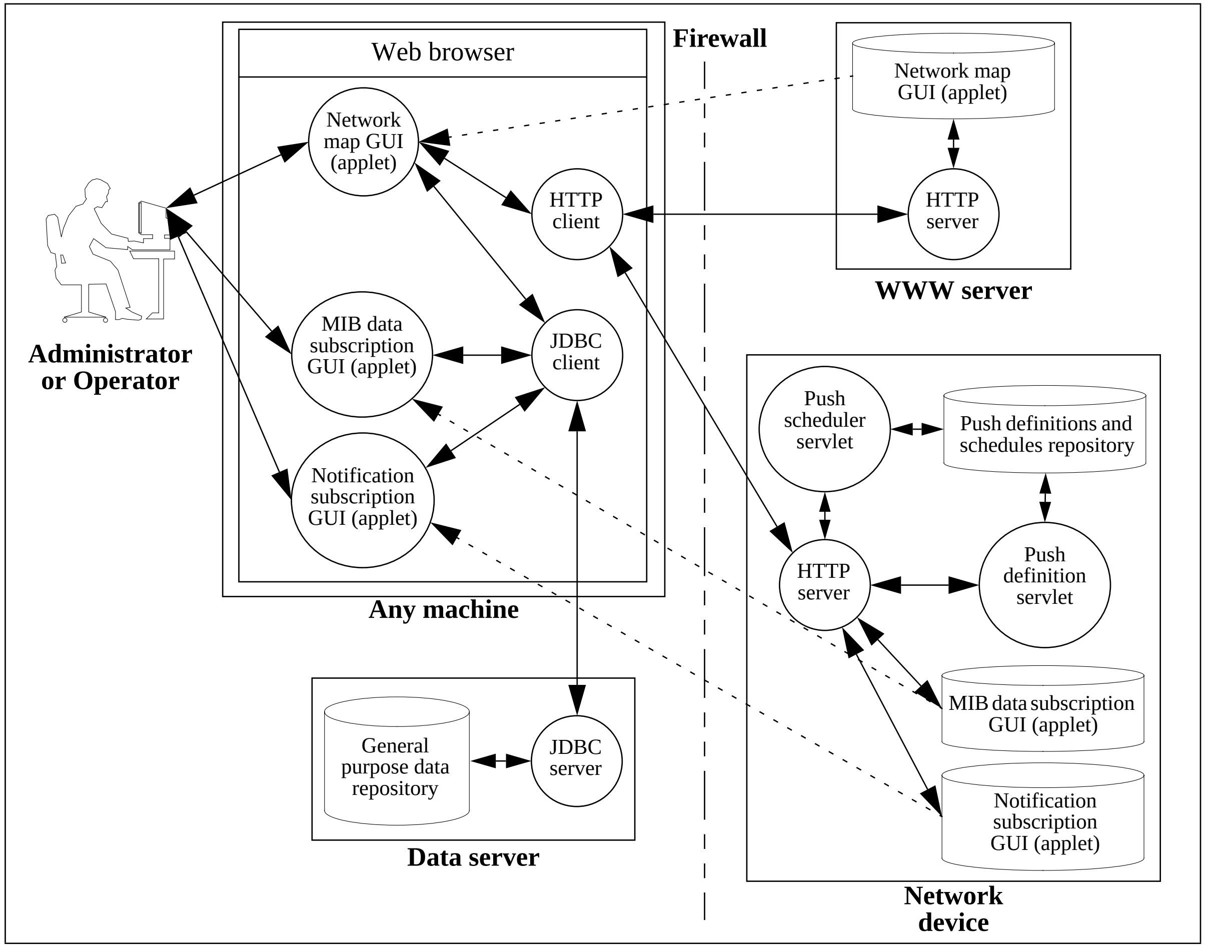
Data Polling Based on HTTP



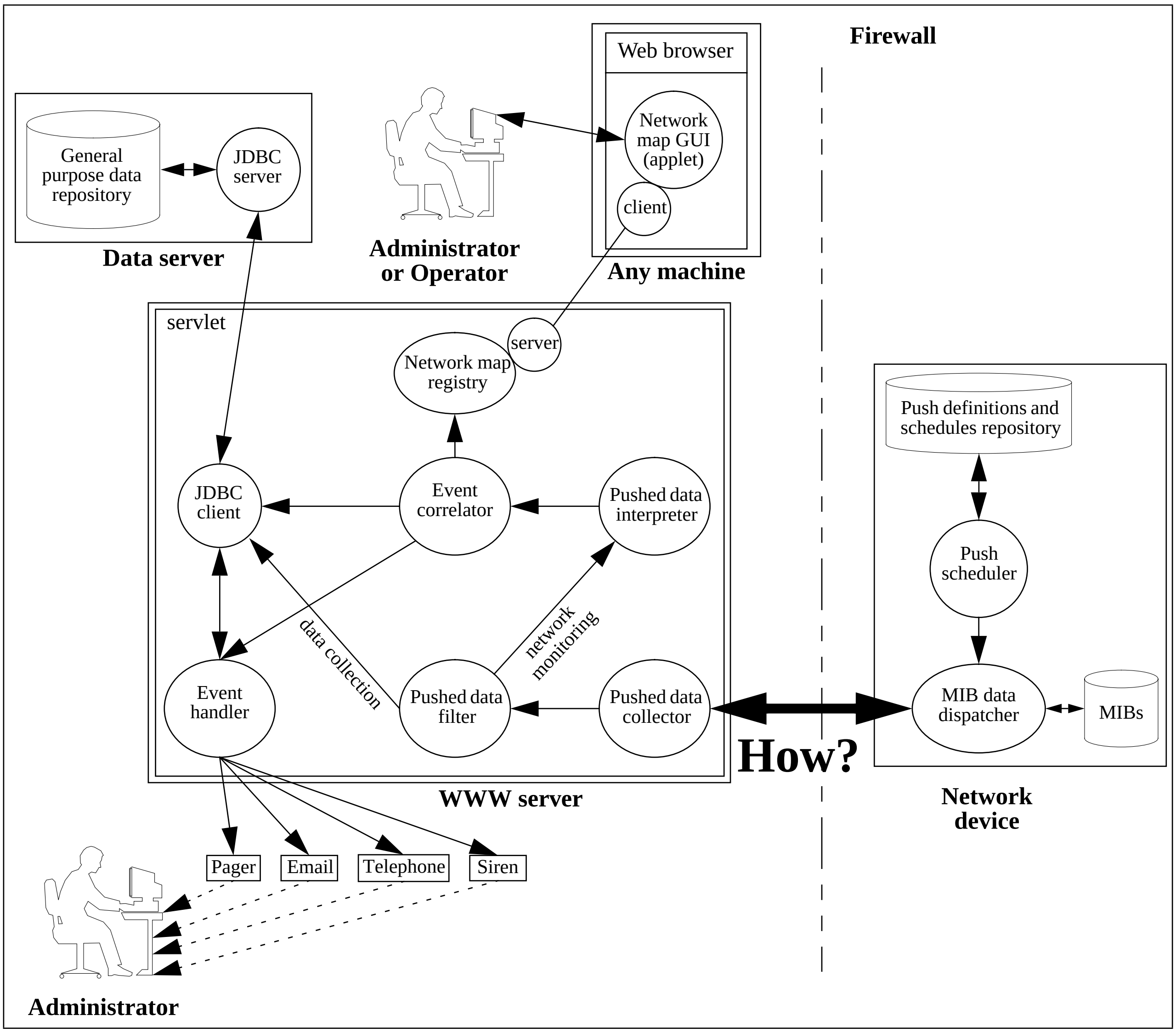
The Push Model

- 3 phases:
  - publish
  - subscribe
  - distribute

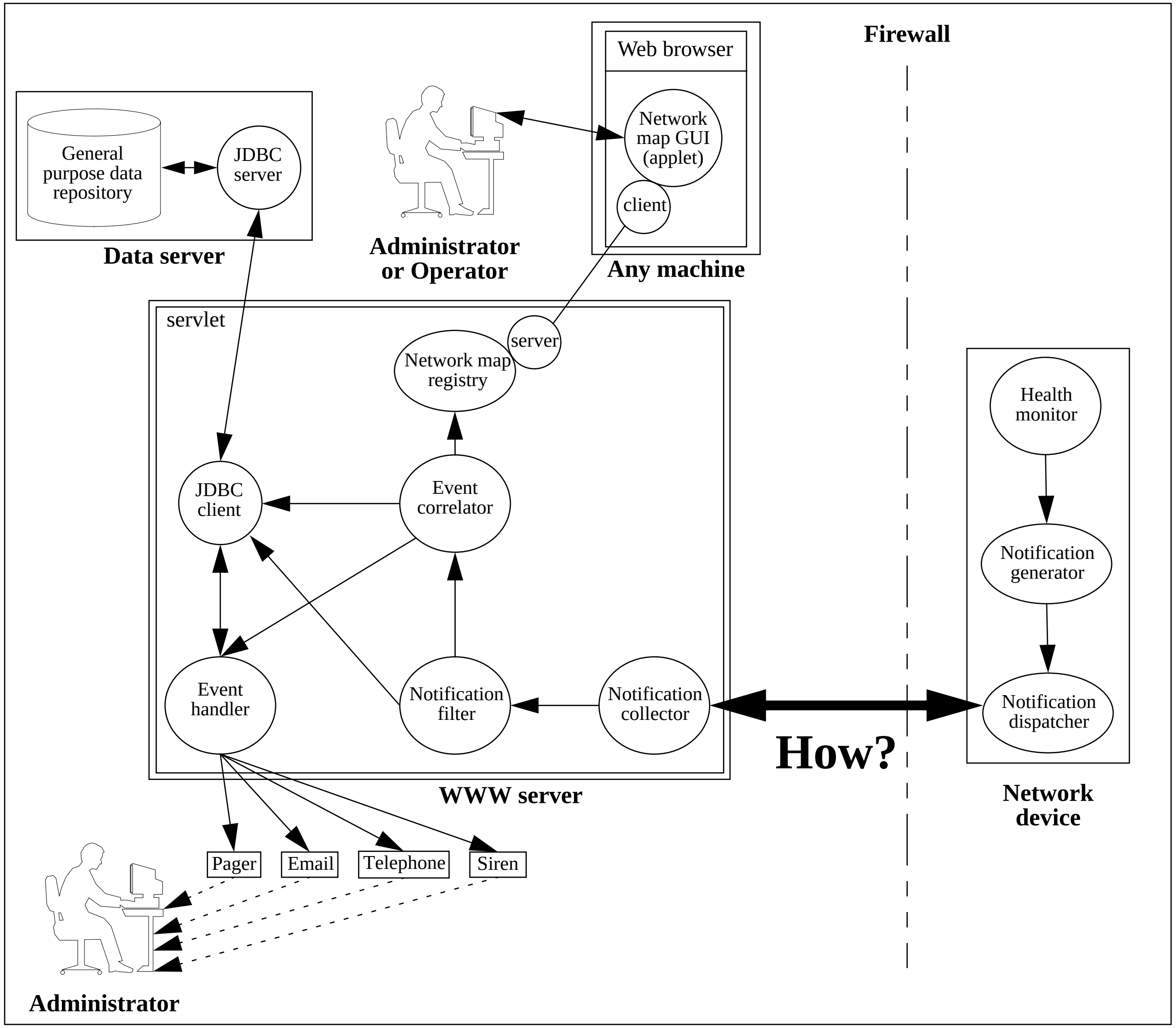
Publication and Subscription Phases


Web browser
Firewall
Network map GUI (applet)
HTTP client
HTTP server
WWW server
Administrator or Operator
MIB data subscription GUI (applet)
JDBC client
Notification subscription GUI (applet)
Any machine
Push scheduler servlet
Push definitions and schedules repository
HTTP server
Push definition servlet
MIB data subscription GUI (applet)
Notification subscription GUI (applet)
Network device
General purpose data repository
JDBC server
Data server

Distribution Phase for Network Monitoring and Data Collection



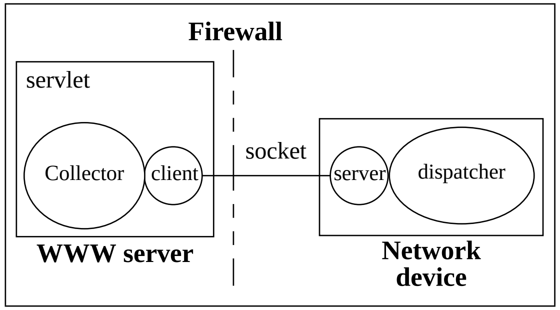
Distribution Phase for Notifications



Issues

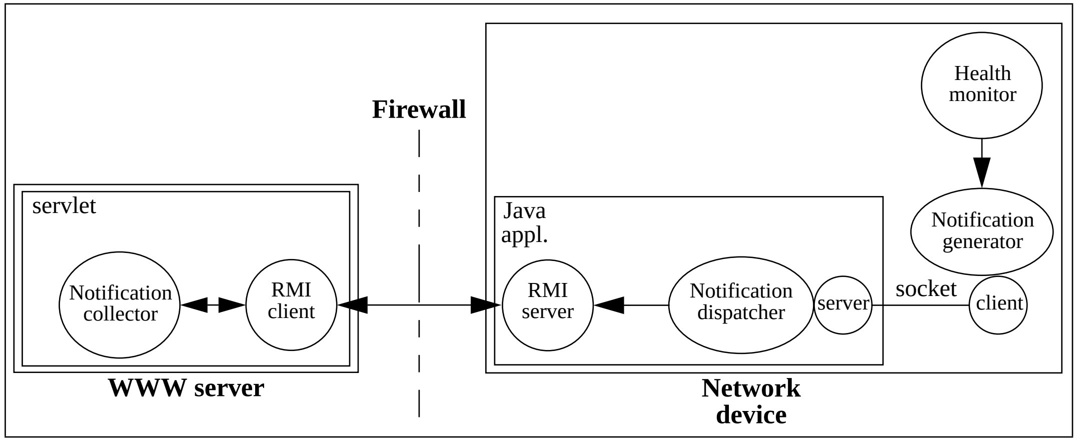
- Positions of client and server now reversed:
  - transfer of management data initiated by the agent
  - client side of the persistent connection still on the manager side
  - we want the server to initiate a transfer in a client/server architecture!
- Firewalls: HTTP vs. sockets vs. RMI
- Timeout of the persistent connection

Sockets

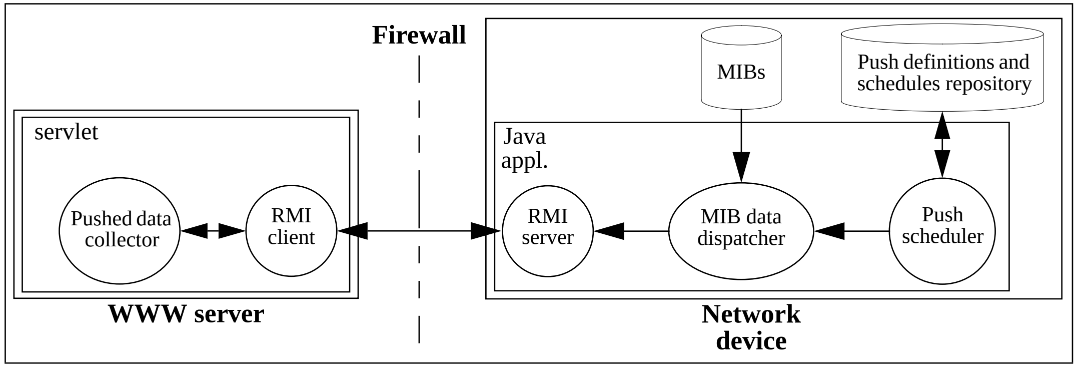


- Advantages:
  - bi-directional
  - simple to implement
- Drawbacks:
  - persistent connection is instable if socket timeout < push period
  - robustness: notifications delivery by the agent depends on a persistent connection created by another entity (the manager)
  - firewalls: require specific settings (UDP or TCP)

Java RMI (1/2)



Distribution via Java RMI for notification delivery

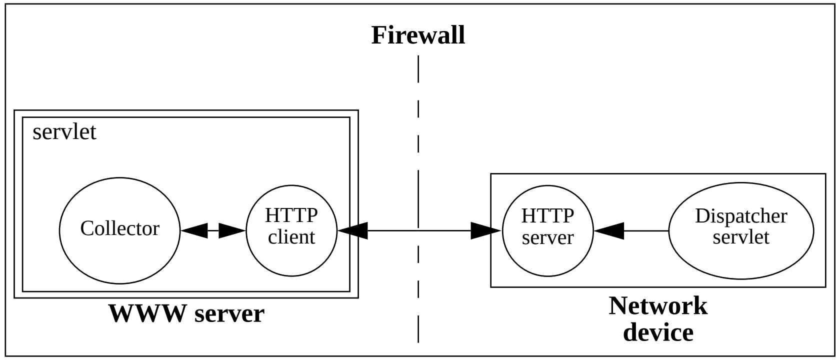


Distribution via Java RMI for data collection and network monitoring

Java RMI (2/2)

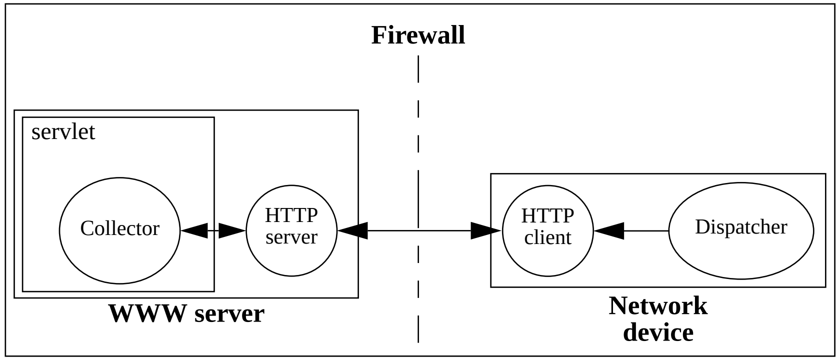
- Advantages:
  - bi-directional association between RMI client and RMI server (sockets underneath)
  - elegant design (fully OO network management)
- Drawbacks:
  - requires a full JVM in agents
  - RMI implementations are slow, and thus not scalable
  - firewalls: how to control ports used by RMI clients? (supposedly transparent to the application)

HTTP (1/1)



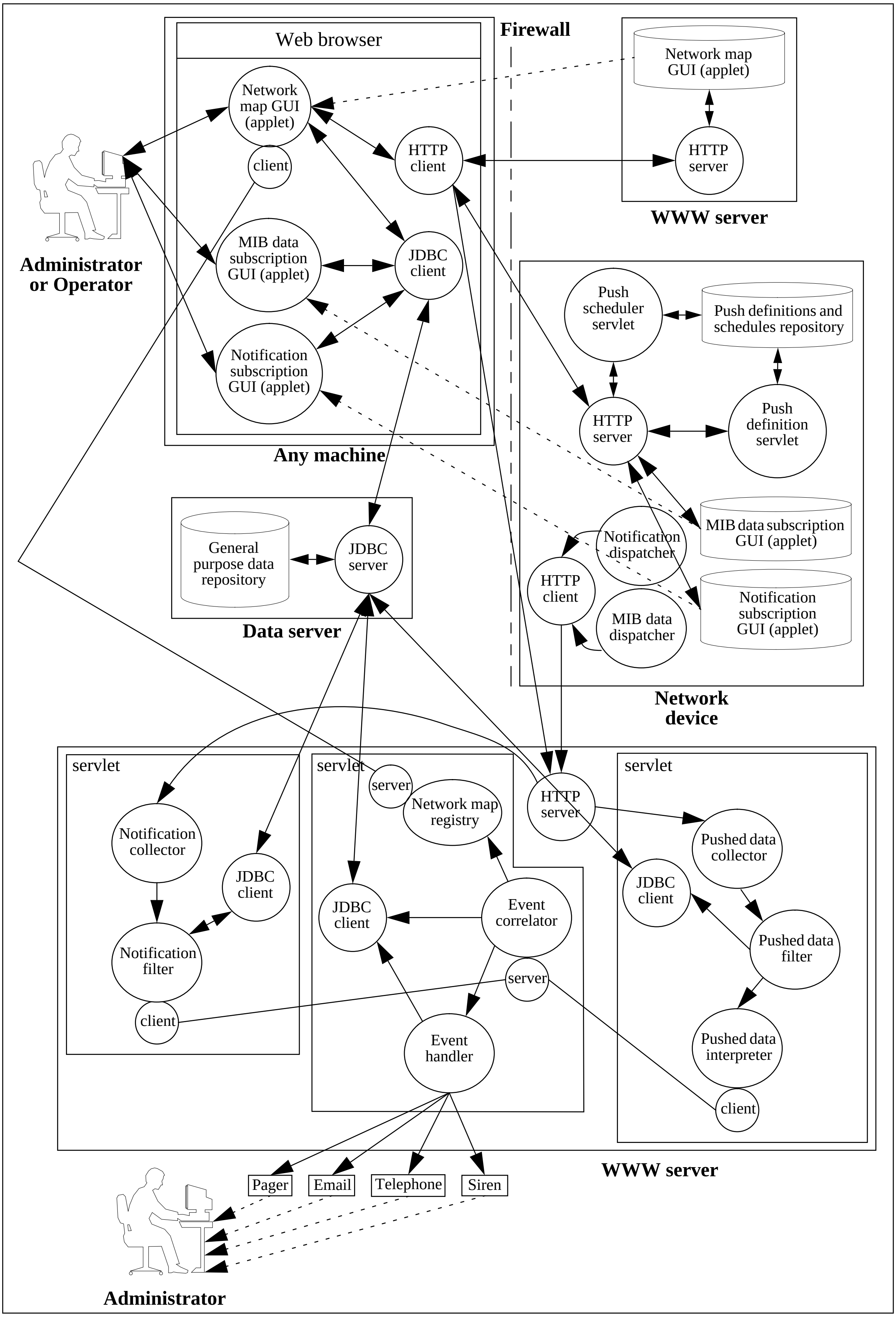
- Infinite number of MIME parts [Netscape 1995]
- Advantages:
  - simple to implement
  - firewalls: no change or minor change (assuming Web access)
- Drawbacks:
  - need to control the HTTP server timeout (Apache: OK)
  - need to control the operating system timeout (patch kernel: not OK!)

HTTP (2/2)



- Advantages:
  - simple design: client and server on the good side
  - robustness: the agent can reconnect immediately in case of timeout, it does not have to count on the manager
  - firewalls: no change or minor change (assuming Web access)
- Drawbacks:
  - security: connection created by external agent, not internal manager

The Collapsed NMP



Many Problems Have Been Solved (1/4)

- Customers:
  - platforms are too expensive (hardware and software)
    - no need for dedicated hardware
    - less expensive software (small collection of applets and scripts)
    - capitalize on previous investment (e.g., use existing RDBMS)
  - limited support of third-party RDBMS vendors
    - no need for peer-to-peer agreement, use JDBC instead
  - need for Unix expertise to maintain existing platforms:
    - cost to migrate to Windows is too high:
      - migration cost from/to any platform is minimal
      - GUIs and Java applications are platform independent

Many Problems Have Been Solved (2/4)

- Network equipment vendors:
  - the support of many device-specific add-ons for many NMPs and many OSs is too expensive:
    - ➠ single applet
- Customers and network equipment vendors:
  - poor time-to-market for add-ons:
    - ➠ zero time-to-market, whatever the market share
    - ➠ access to integrated network management for startup companies
  - versioning:
    - ➠ MIBs and applets upgraded together, device by device

Many Problems Have Been Solved (3/4)

- Protocol efficiency:
  - poor efficiency of BER encoding
    - ➠ BER encoding no longer used
  - poor efficency of SNMP
    - ➠ SNMP as a communication protocol is replaced with HTTP
    - ➠ more management data per packet with push or RMI
    - ➠ management data can be compressed (gzip)
    - ➠ with RMI, no MIB variables anymore (higher level of abstraction)

Many Problems Have Been Solved (4/4)

- Security:
  - management of VPNs: HTTP security may be used instead of encryption boxes
    - ➠ still weak security
    - ➠ at least better than community string
  - firewalls: HTTP simpler than SNMP
- TCP vs. UDP:
  - HTTP makes it possible to use TCP to transfer management data
  - reliable transport layer for SNMP notifications:
    - ➠ important notifications are no longer lost for silly reasons
    - ➠ still no guarantee of delivery

Bonus

- Redundant managers are simple to support with push:
  - one step toward fault-tolerance

New Problems

- NMP vendors:
  - loss of revenue
  - is a market of generic applets sustainable?
  - need to find niche markets:
    - fault tolerance
    - large networks where scalability is stretched to the limits
    - real-time networks where responsiveness and speed are stretched to the limits
- Known problems:
  - clock synchronization
  - need to control timeout values
- Potential problem (to be investigated):
  - Java is slow (e.g. JDBC) --> impact on scalability?

Future Research: Implementation of the Push Model

- Demonstrate the feasibility of Web-based network management
- Develop a network management application in Java that implements the push model:
  - Sun’s JMAPI and Java DMK (M-beans)
  - AdventNet’s SNMP package
  - EmbeddedJava
- Compare the performances of pull- and push-based management
- Investigate the issues of scalability and performance
- Prototype: IP routers of Lightning (and others?)

Related Publications

J.P. Martin-Flatin. *The Push Model in Web-Based Network Management*. Technical Report SSC/1998/023, version 3, SSC, EPFL, Lausanne, Switzerland, November 1998. Submitted to *ACM Computer Communication Review*, December 1998.

J.P. Martin-Flatin. *Push vs. Pull in Web-Based Network Management*. Technical Report SSC/1998/022, version 3, SSC, EPFL, Lausanne, Switzerland, November 1998. Accepted for publication in *Proc. 6th IFIP/ IEEE International Symposium on Integrated Network Management (IM'99)*, Boston, MA, USA, May 1999.

J.P. Martin-Flatin. *IP Network Management Platforms Before the Web*. Technical Report SSC/1998/021, version 2, SSC, EPFL, Lausanne, Switzerland, December 1998.