ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Web-Based Management of IP Networks and Systems

Imperial College, London, UK
June 2, 2000

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

Executive Summary: A New Mgmt Architecture for the IP World

- A new problem demands a new solution:
  - SNMP focused on simplicity, interoperability, and network mgmt
  - SNMP is good at managing small data networks
  - the market now demands integrated mgmt = integration of network, systems, application, service, a nd policy mgmt
  - vendors are now working on WBEM/CIM:
    - ➠ we ought to deal with several info. models
- Previous proposals focused on the organizational model (MbD, Script MIB) or the information model (CIM)
- WIMA (Web-based Integrated Mgmt Architecture) proposes:
  - a new organizational model
  - a new communication model

Outline

- Background
- Problems with SNMP-based mgmt
- Web-based mgmt
- Push model
- New communication model
- XML
- JAMAP: research prototype
- Conclusion

IP Management Platforms: Mandatory Tasks

- Monitoring:
  - detect faults in network devices, network links, and systems:
    - reactive w.r.t. faults
    - proactive w.r.t. short-term complaints from users
- Data collection:
  - gather data to build daily, weekly, and monthly reports:
    - proactive w.r.t. long-term complaints from users
- Notification handling:
  - pseudo real-time (no hard real-time constraints)
  - react to events generated by the agents (SNMP notifications)
  - react to events generated by the manager (rule-based data interpreter)
- Configuration mgmt

Regular Management

- Ongoing monitoring and data collection
- Automated
- 2 modes:
  - attended mode: operators gazing at GUIs (red-icon angst)
  - unattended mode:
    - automated correlation
    - alarms trigger pager, email, telephone, siren, etc.
- Midsize and large networks

Ad Hoc Management

- Troubleshooting, configuration mgmt, and temporary monitoring
- Not automated
- Single mode: attended (administrators or operators)
- All networks
- Replaces regular mgmt in small networks

Outline

- Background
- Problems with SNMP-based mgmt
- Web-based mgmt
- Push model
- New communication model
- XML
- JAMAP: research prototype
- Conclusion

Problems with SNMP (1/2)

- Scalability, network overhead, and latency are adversely affected by old protocol design decisions:
  - BER encoding is inefficient [Mitra 1994]
  - SNMP table retrieval mechanism is poor (“holes”, many messages)
  - max. message size is too low (484 bytes guaranteed, up to 1472 bytes)
  - OIDs take much more space than values
  - mgmt data cannot be compressed
- Security:
  - SNMPv1 and SNMPv2c: community string (simplistic)
  - SNMPv3: better, still simple, but not used
  - Next step: expensive encryption hardware (e.g., VPNs)
  - firewalls: complex and costly UDP relays [Chapman & Zwicky 1995]

Problems with SNMP (2/2)

- Unreliable transport protocol:
  - important SNMP notifications (unacknowledged) are lost for silly reasons (e.g., buffer overflow)
  - SNMPv3 informs (acknowledged) are not used yet
  - important mgmt data requires retransmissions at the application level
- Distribution:
  - still no framework to distribute mgmt across a hierarchy of managers
    - ➠ mgmt platforms resort to proprietary extensions

Problems with SNMP MIBs

- Low-level semantics:
  - only instrumentation MIBs
  - no standard high-level APIs
  - site-specific network applications developed from scratch:
    - ➠ bound to the API of a specific mgmt platform, not to a standard technology

Problems with SNMP-Based Mgmt Platforms

- Too expensive for customers (hardware and software)
- Limited support for third-party RDBMSs
- The support for device-specific mgmt GUIs is too expensive for equipment vendors:
  - many mgmt platforms
  - many operating systems
  - many GUIs
- Poor time-to-market for mgmt GUIs
- MIB versioning
- Investment bound to a specific operating system
- (These problems are due to the way the SNMP market evolved)

Nontechnical Problems with SNMP-Based Mgmt

- SNMP expertise is domain specific --> rare and expensive
- SNMP was devised for network mgmt in the late 1980s:
  - myth of the dumb agent [Wellens & Auerbach 1996]
  - myth of the collapsing backbone [Wellens & Auerbach 1996]
  - myth of the collapsing manager [Ph.D. thesis]
  - SNMP is not adequate for integrated mgmt in the 2000s
- Evolution of SNMP hampered by legacy systems:
  - “better replace than repair”

Outline

- Background
- Problems with SNMP-based mgmt
- Web-based mgmt
- Push model
- New communication model
- XML
- JAMAP: research prototype
- Conclusion

Web-Based Management

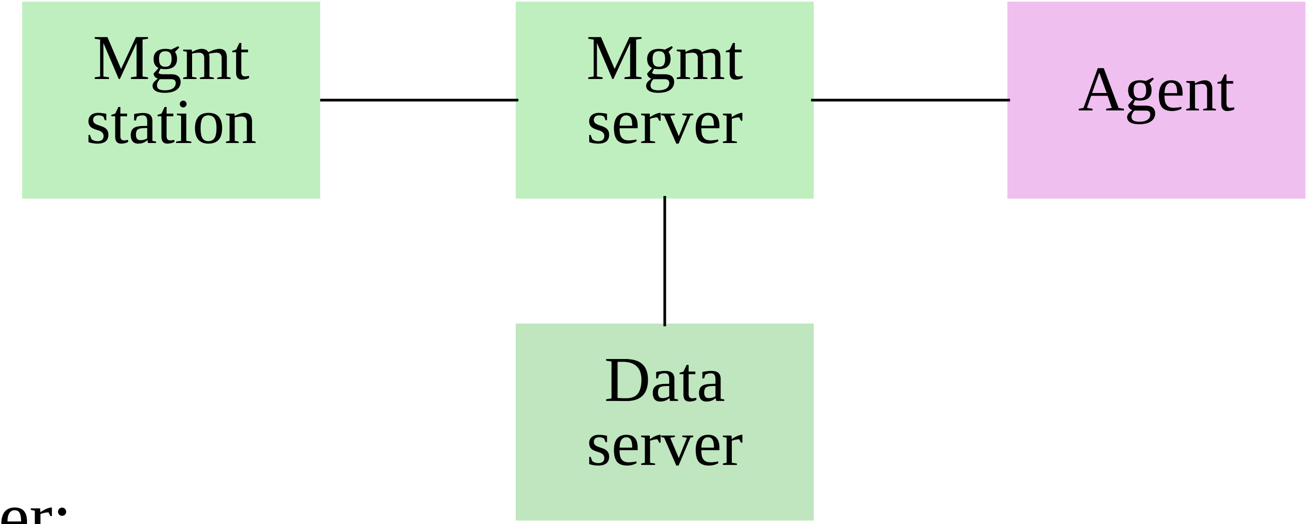
- Definition: integrated mgmt based on Web technologies
- Large choice of Web technologies:
  - HTML forms
  - CGI (Perl scripts, Tcl/Tk scripts, shell scripts, binaries)
  - Java applets, servlets, and applications
  - Java Object Serialization
  - Java RMI (distributed objects)
  - Java IDL (CORBA)
  - JDBC (databases)
  - XML
  - ...

Why Use Web Technologies?

- Reduce development costs of mgmt GUIs (applets):
  - less expensive for customers
- Zero the time-to-market of mgmt GUIs (embedded)
- Suppress the need for separate mgmt platforms:
  - integrated mgmt
  - put small and large equipment vendors in fair competition
- Simplify mgmt of remote subsidiaries across firewalls
- Reduce network overhead by compressing mgmt data
- Make mgmt platforms more open, more modular, and less costly
- Improve the support for 3rd-party databases

Better Design of the Mgmt Platform (1/2)

- Split manager:



- Split mgmt server:
  - was: big, monolithic, opaque, and proprietary code
  - now:
    - integration of COTS components and OO frameworks
    - fine-grained competition between vendors (e.g., buy an event correlator):
      - less expensive
      - manager to manager: more interoperable
      - no longer enchained by big investment

Better Design of the Mgmt Platform (2/2)

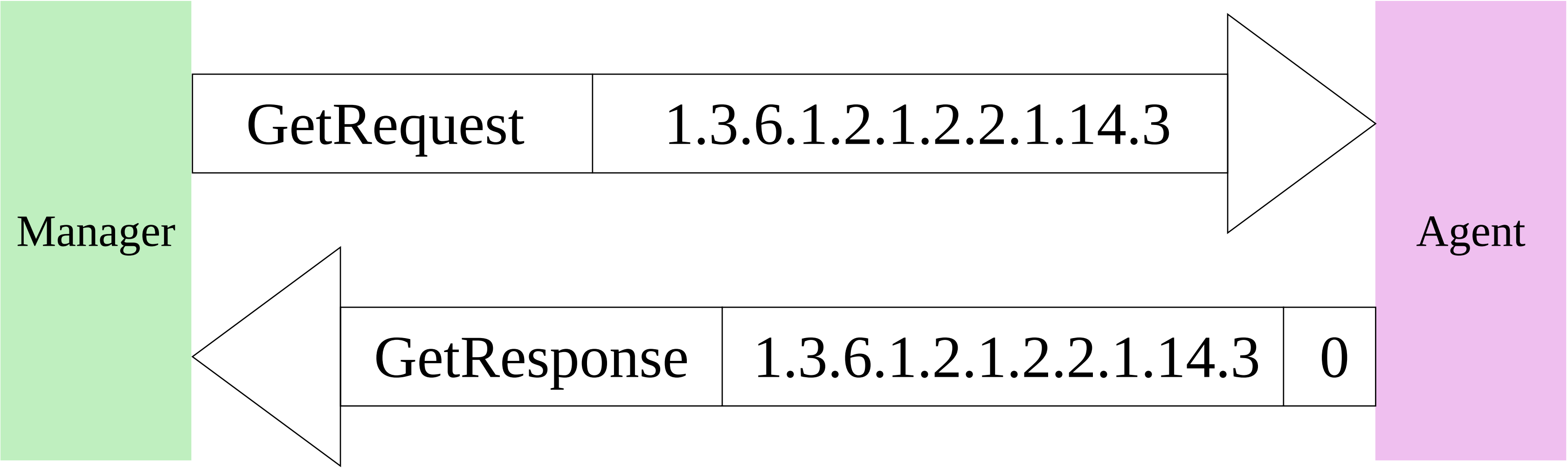
- Generic hooks for accessing the data server:
  - virtually all databases support JDBC or XML
  - customers are no longer dependent on peer-to-peer agreements between mgmt-platform and database vendors
  - customers need not buy a new database for integrated mgmt

Outline

- Background
- Problems with SNMP-based mgmt
- Web-based mgmt
- Push model
- New communication model
- XML
- JAMAP: research prototype
- Conclusion

The Push Model

- Why use the push model?
  - reduce network overhead of mgmt data --> save network bandwidth
  - move some workload from the manager to the agents
  - e.g., error rate for inbound traffic through interface #3:



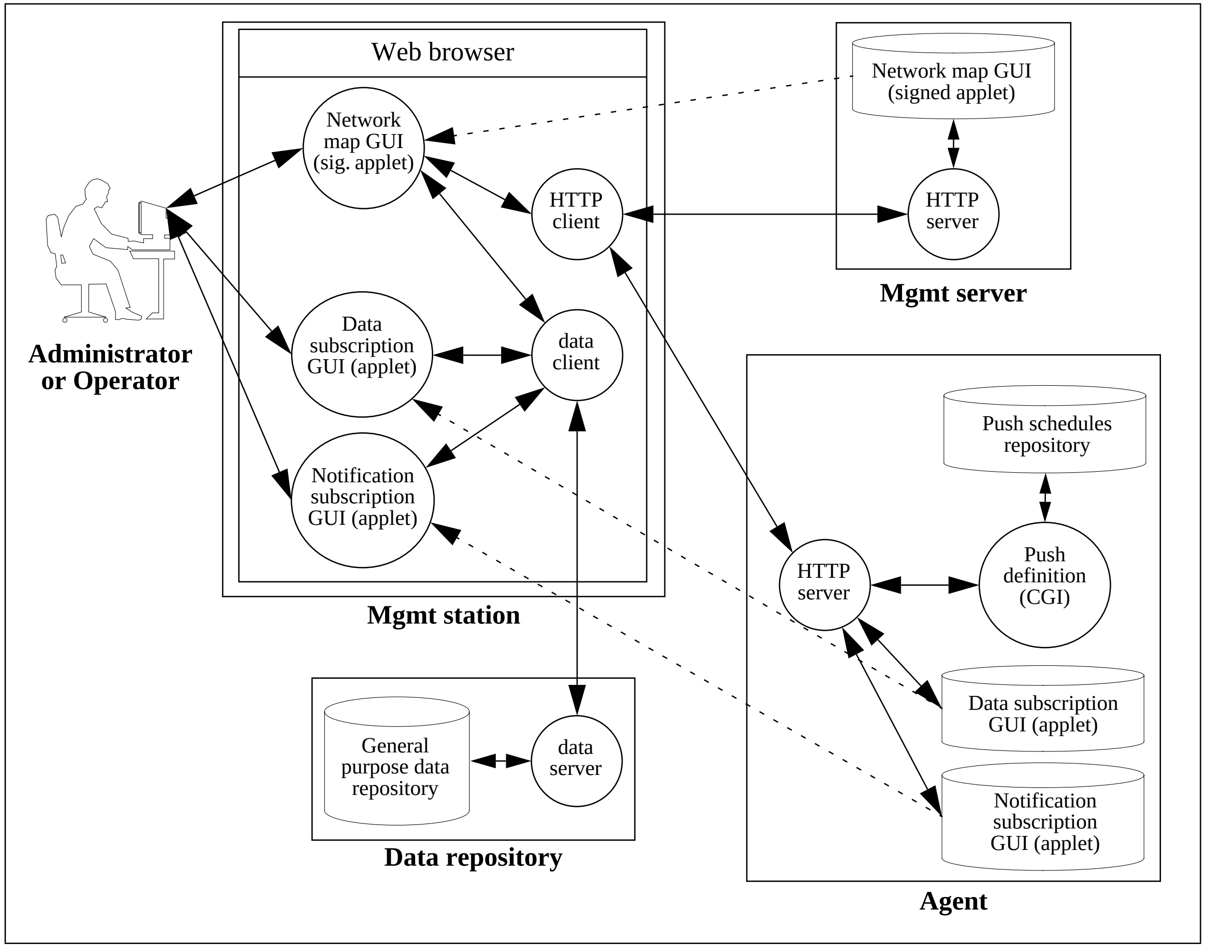
`get`: (2xOID) + value

`get-next`: (3xOID) + value

Characterization of the Push Model

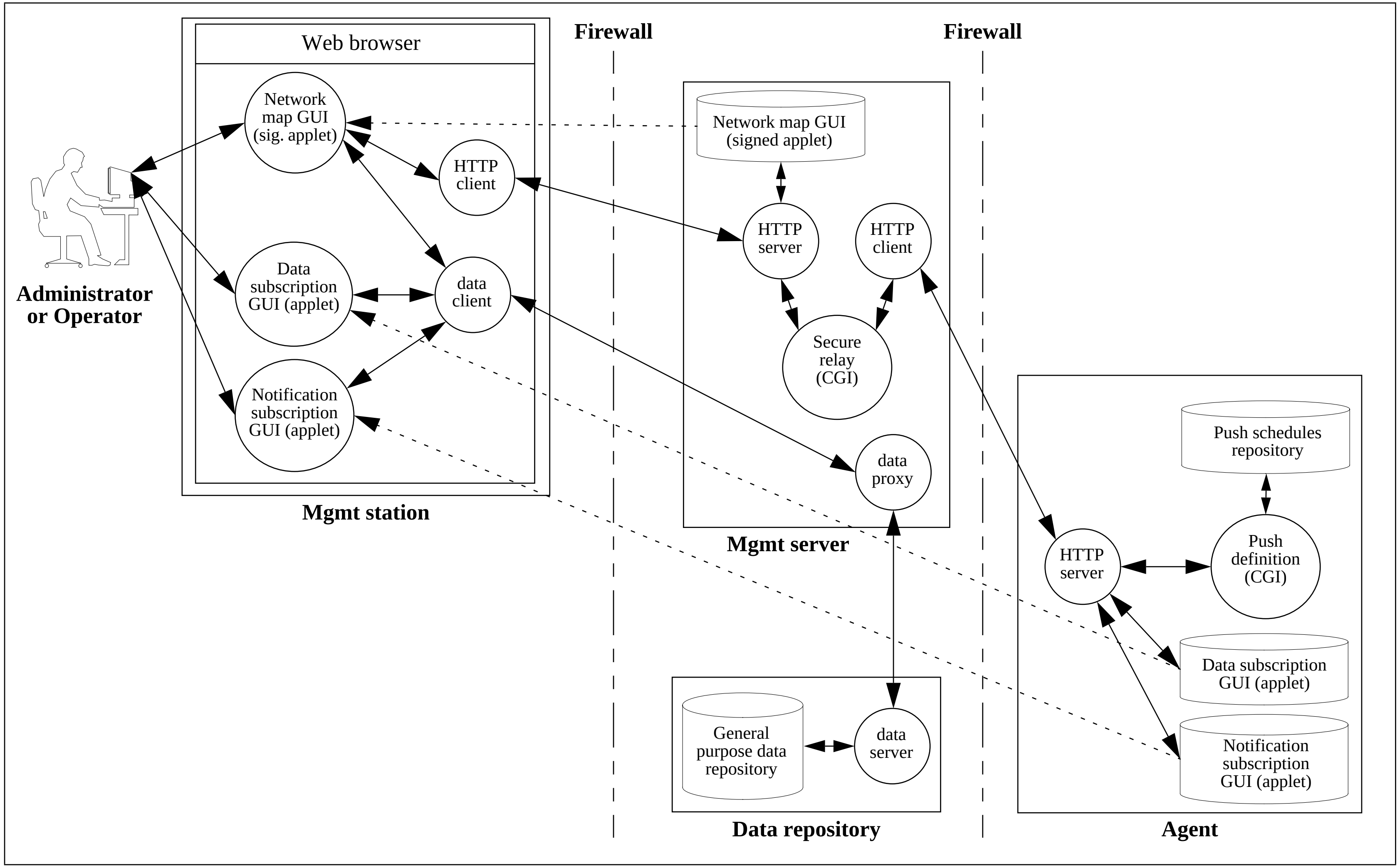
- Variant of the Publish-Subscribe design pattern (Observer in [Gamma *et al*. 1995]):
    - one subscriber (manager), many publishers (agents)
    - 3 phases: publication, subscription, and distribution
- Pseudo client-server communication model:
    - client sends data to server
    - server may or may not acknowledge receipt of data
- Client = agent
- Server = manager
- Parallel and independent data transfers initiated by the clients

WIMA: Publication and Subscription Phases

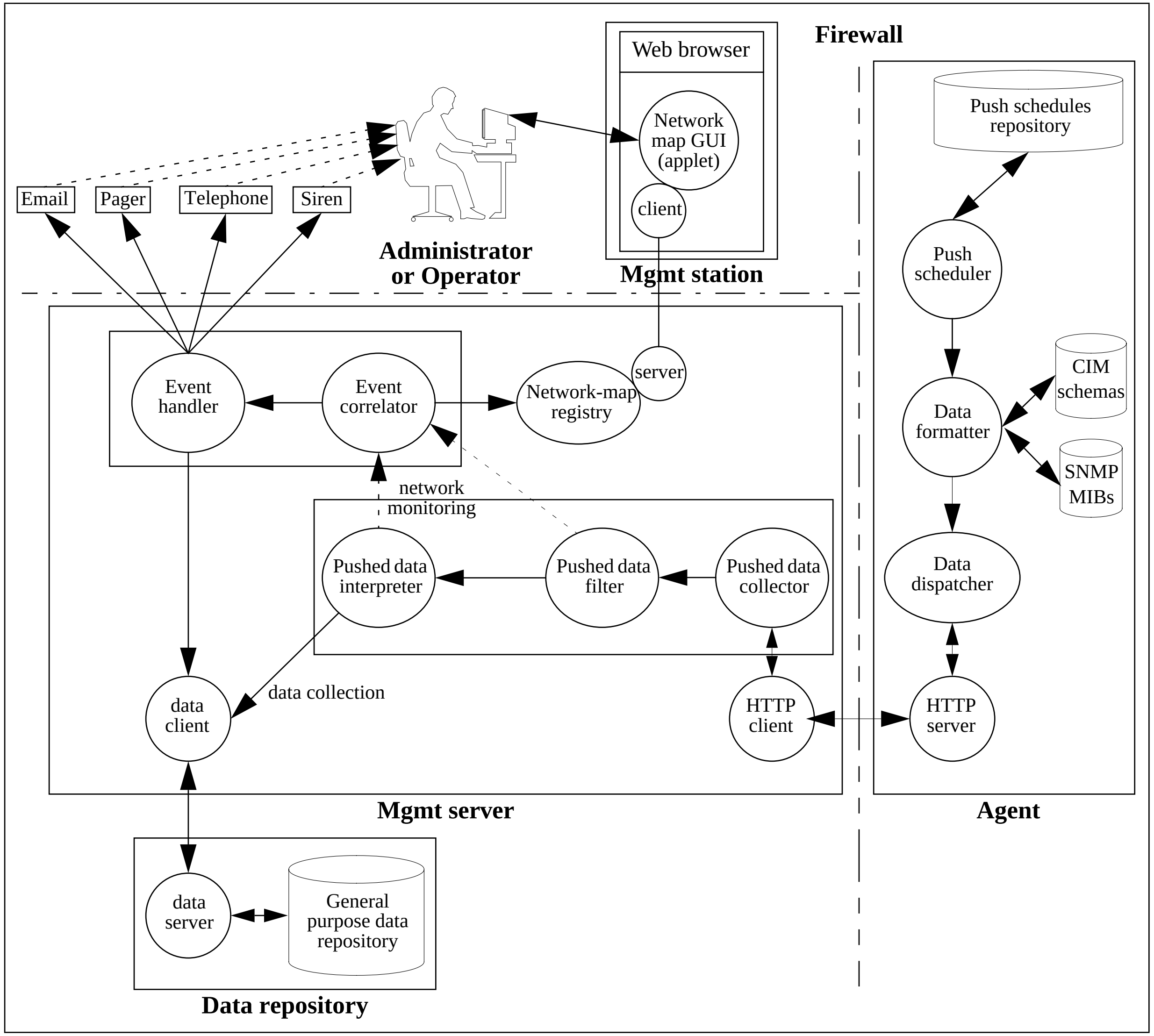

Web browser
Network map GUI (sig. applet)
HTTP client
Data subscription GUI (applet)
data client
Notification subscription GUI (applet)
Mgmt station
Administrator or Operator
Network map GUI (signed applet)
HTTP server
Mgmt server
Push schedules repository
HTTP server
Push definition (CGI)
Data subscription GUI (applet)
Notification subscription GUI (applet)
Agent
General purpose data repository
data server
Data repository

CGI = CGI script, CGI binary, Java servlet, etc.

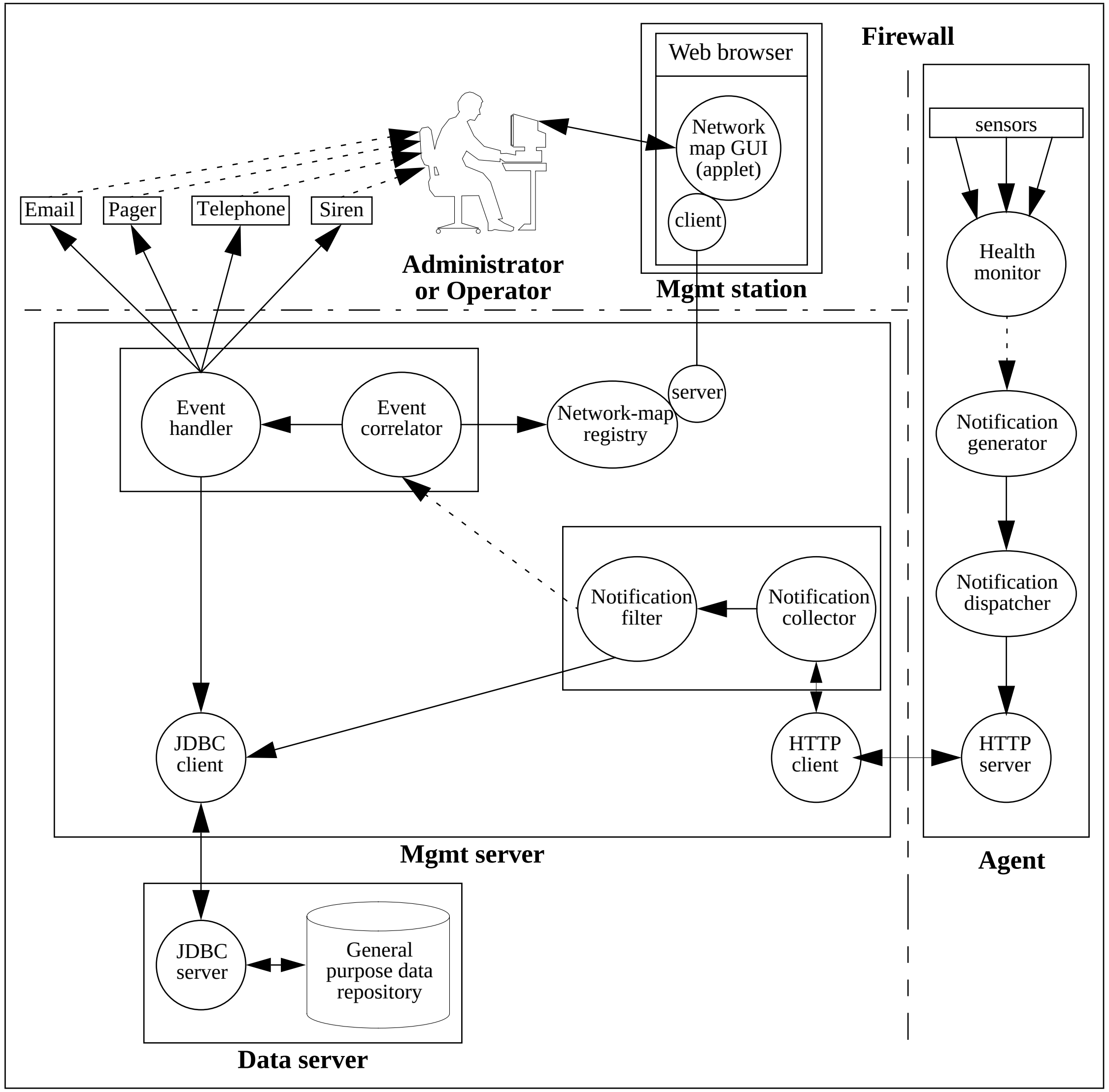
WIMA: Publication and Subscription Phases (Firewall)


Web browser
Firewall
Firewall
Network map GUI (sig. applet)
HTTP client
Network map GUI (signed applet)
Administrator or Operator
Data subscription GUI (applet)
data client
HTTP server
HTTP client
Secure relay (CGI)
Notification subscription GUI (applet)
data proxy
Push schedules repository
Mgmt station
Mgmt server
HTTP server
Push definition (CGI)
Data subscription GUI (applet)
General purpose data repository
data server
Notification subscription GUI (applet)
Data repository
Agent

WIMA: Distribution Phase for Monitoring and Data Collection



WIMA: Distribution Phase for Notifications



Outline

- Background
- Problems with SNMP-based mgmt
- Web-based mgmt
- Push model
- New communication model
- XML
- JAMAP: research prototype
- Conclusion

New Communication Model: WIMA-CM

- HTTP
- UDP --> TCP
- Persistent TCP connections
- Persistent HTTP connections with MIME multipart
- Two connections per agent: urgent vs. nonurgent data
- Compress mgmt data
- Cope with firewalls
- Timeouts and reconnections by the manager

Communication based on HTTP (1/2)

- Four APIs to communicate between agents and managers:
  - HTTP
  - sockets API
  - Java RMI
  - Java IDL (CORBA)
- Distributed objects (Java RMI or CORBA):
  - telecoms world = yes,   IP world = no
  - the *my-middleware-is-better-than-yours* syndrome
  - cost
  - footprint on agents

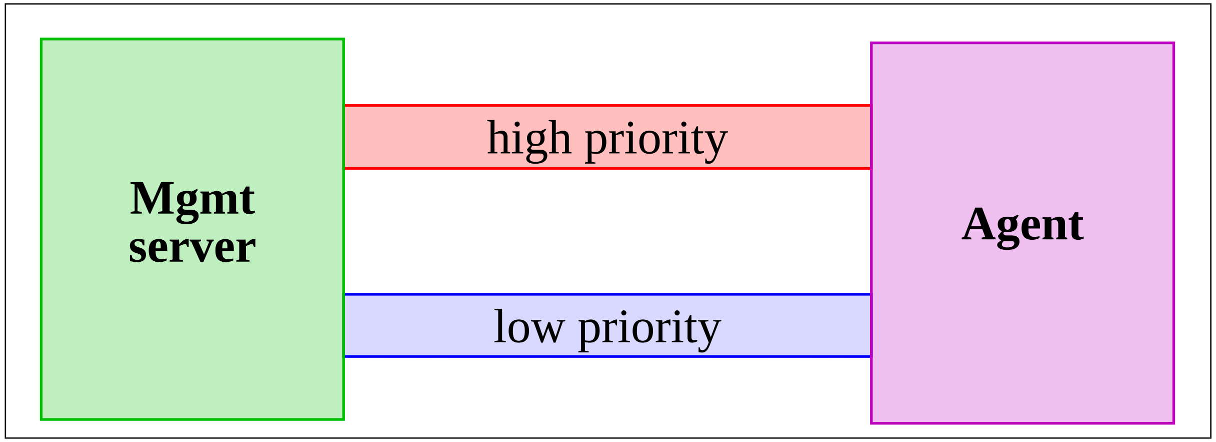
Communication based on HTTP (2/2)

- HTTP > sockets API:
  - avoid a domain-specific transfer protocol
  - firewall setup easier for nonexperts:
    - important for small and midsize companies
  - if Java servlets:
    - manager: natural communication between Java servlets
    - same technology:
      - between agents and manager
      - within the manager

Persistent TCP Connections

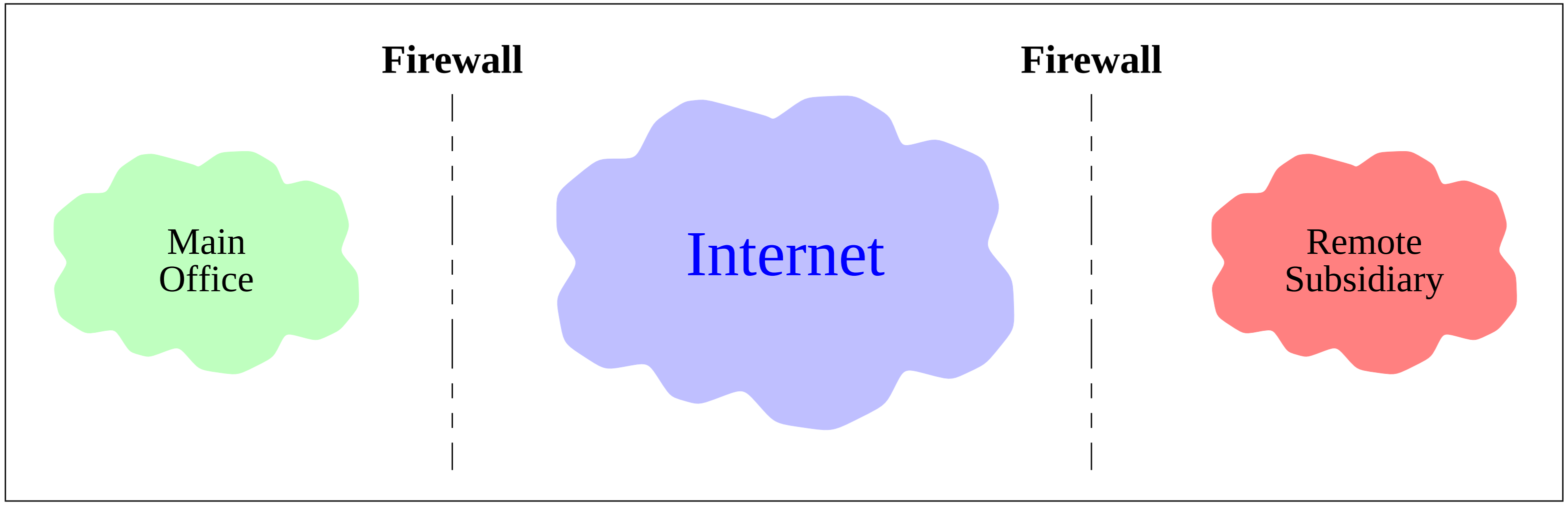
- TCP vs. UDP:
  - decrease losses of mgmt data:
    - ➠ still no guarantee of delivery
  - retransmissions and ACKs need not be performed at the app. level:
    - ➠ better interoperability
    - ➠ simpler application
- Persistent TCP connections:
  - avoid overhead of frequently setting up and tearing down connections
  - necessary for security reasons: the agent pushes mgmt data in a pre-existing connection

Two Persistent Connections Per Agent



- High priority: e.g., urgent SNMP notifications
- Memory overhead for the manager:
  - less that 8 MBytes to manage 400 of agents
  - requires special tuning of the kernel:
    - ➠ drawback: we still need a dedicated mgmt platform

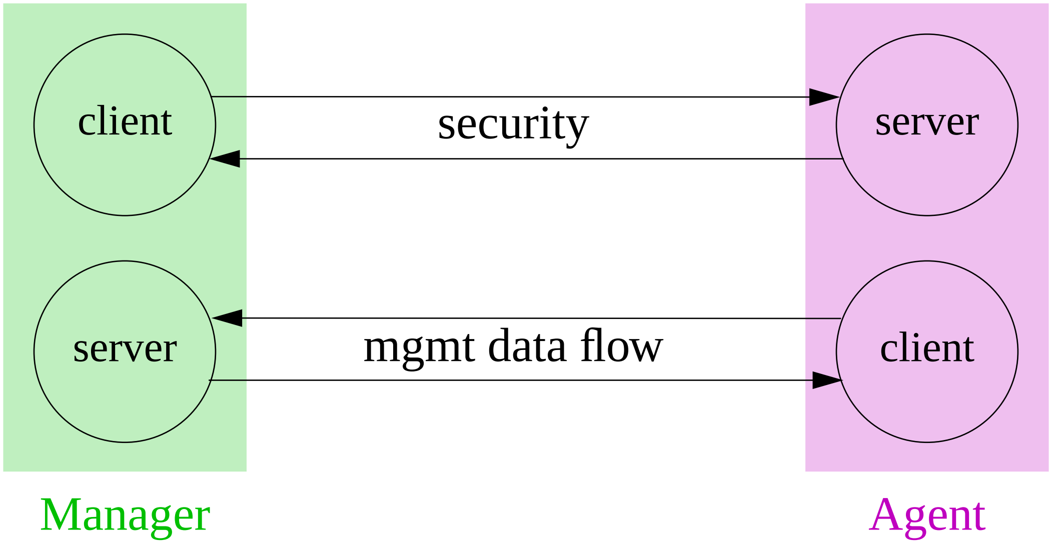
Firewalls



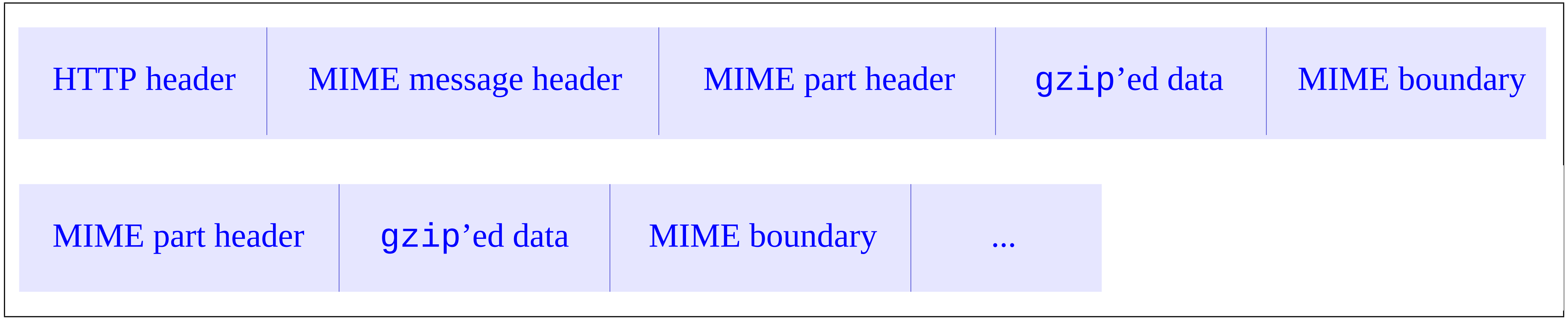
- Robustness principle: TCP connections should be created by internal trusted manager, not external untrusted agent:
  - avoid TCP ports probing by external intruders
  - avoid certain DoS attacks (e.g., TCP SYN flooding)

Reversed Client and Server

- Firewalls --> positions of client and server now reversed:
  - transfer of mgmt 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!



Persistent HTTP Connections with MIME Multipart



HTTP header | MIME message header | MIME part header | `gzip`’ed data | MIME boundary

MIME part header | `gzip`’ed data | MIME boundary | ...

MIME = Multipurpose Internet Mail Extensions

- Advantages:
  - simple to implement
  - MIME makes it easy to compress mgmt data transparently
  - firewalls: minor change (assuming Web access already)
- Drawback:
  - how does the manager detect that a connection was broken?

New MIME Types for Part Headers

- General format: <InformationModel>-to-<Encoding>
- Three levels of granularity:
  - *information model* (e.g., CIM2.2-to-string, SNMPv1-to-string)
  - *RFC*: (e.g., RFC2271-to-BER, RFC2571-to-BER)
  - *XML mapping*: (e.g., CIM2.2-to-XML-v2.0, CIM2.3-to-XML-v1.0)
- Potential combinatory explosion of MIME types:
  - poor scalability (constant flow of registrations with IANA)
- We define just one MIME type:
  - Content-Type=”application/mgmt”; mapping=”CIM2.2-to-XML” version=”2.0”

Timeouts and Reconnections

- Issues with persistent HTTP/TCP connections:
  - COTS agents: no control over timeouts --> manager
  - how does the manager know that a persistent TCP conn. was broken?
  - timeouts by the operating system or the application?
- Three solutions:
  - by the kernel: per-socket keepalives (`SO_KEEPALIVE`):
    - ➠ Linux kernel 2.3.99-pre6: `tcp_keepalive_time` (540 s), `tcp_keepalive_intvl` (10 s), `tcp_keepalive_probes` (6)
  - by the kernel: per-socket receive timer (`SO_RCVTIMEO`)
  - by the application: per-socket receive timer (`select`, `poll`, `/dev/poll`)
- We can bind the per-socket timeout value with (i) the push period of a critical data (heartbeat) or (ii) the lowest push period

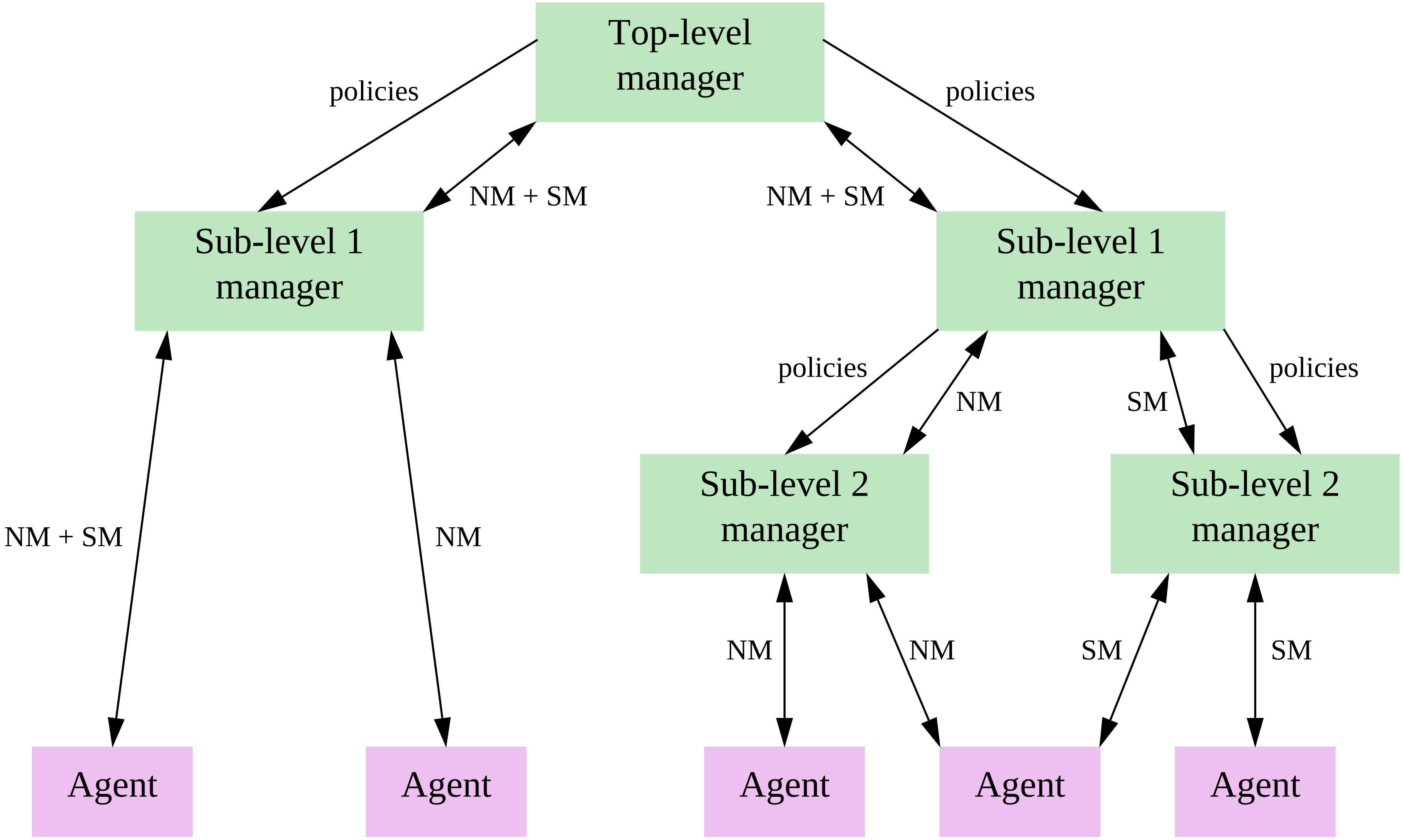
Outline

- Background
- Problems with SNMP-based mgmt
- Web-based mgmt
- Push model
- New communication model
- XML
- JAMAP: research prototype
- Conclusion

Why Use XML?

- A truce in the middleware war
- More generic than IIOP and JRMP
- Low footprint on agents and managers
- Cost =~ zero:
  - a lot of freeware available
- Demanded by customers:
  - becoming ubiquitous in software eng.
- Feature rich:
  - state: transfer data
  - behavior: invoke remote methods

XML for Hierarchically Distributed Management

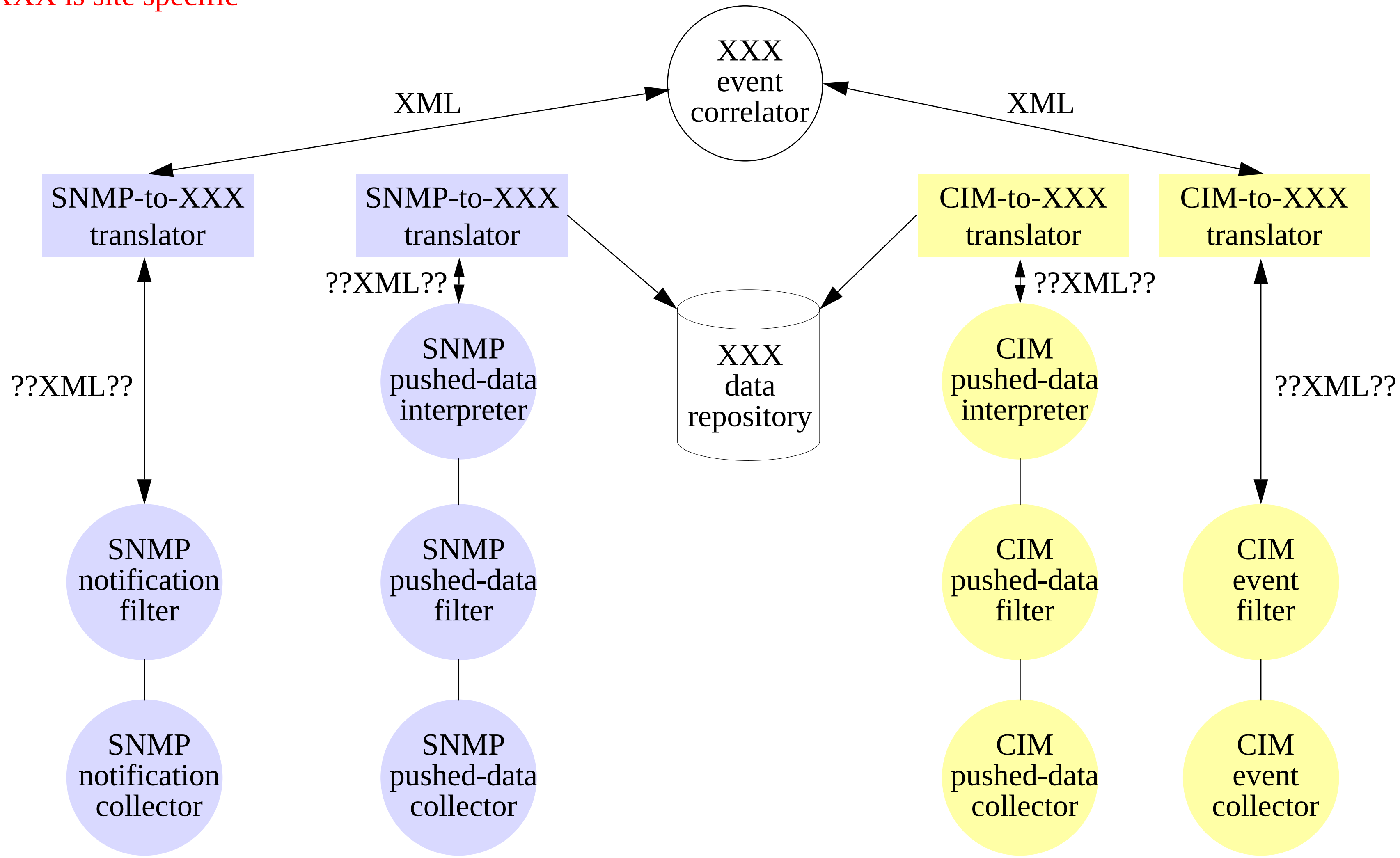


XML for High-Level Semantics

- Clean invocation of remote methods:
  - no need to resort to SNMP’s programming by side effect
- The DMTF learned from the IETF’s mistakes:
  - working on instrumentation MIBs *and* high-level MIBs
- XML renders easy many tasks that are not with SNMP:
  - transfer SNMP MIB table in one bulk (no more “holes”)
  - transfer entire time series for 24h in one bulk
  - ...
- XML interfaces nicely with OO info. models (e.g., CIM), which offer high-level semantics to mgmt applications designers

XML: Dealing with Multiple Information Models

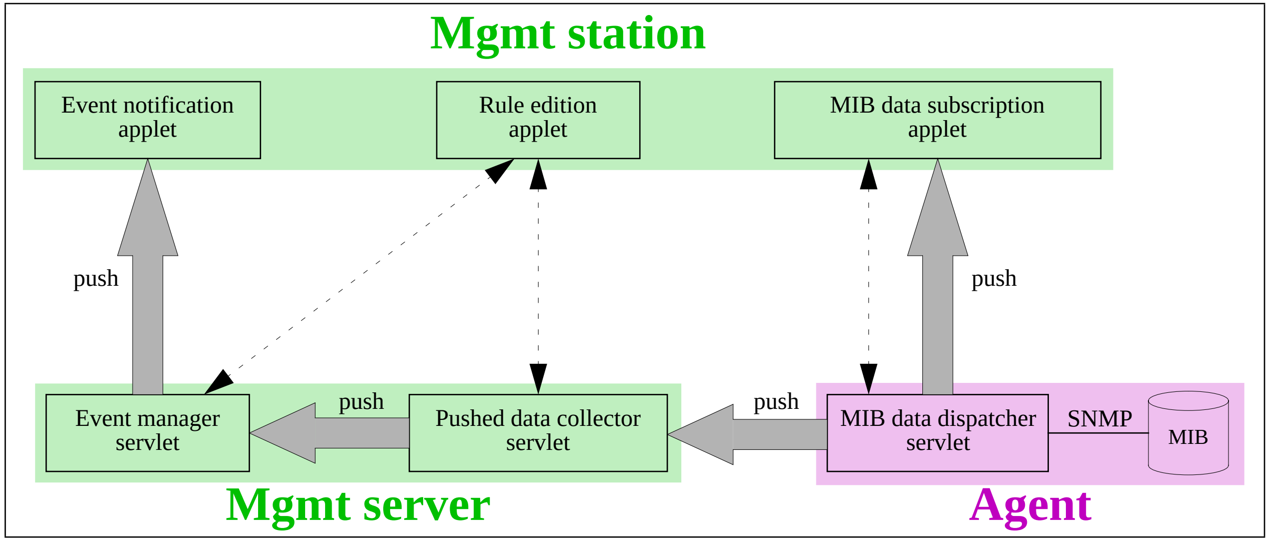
XXX is site specific

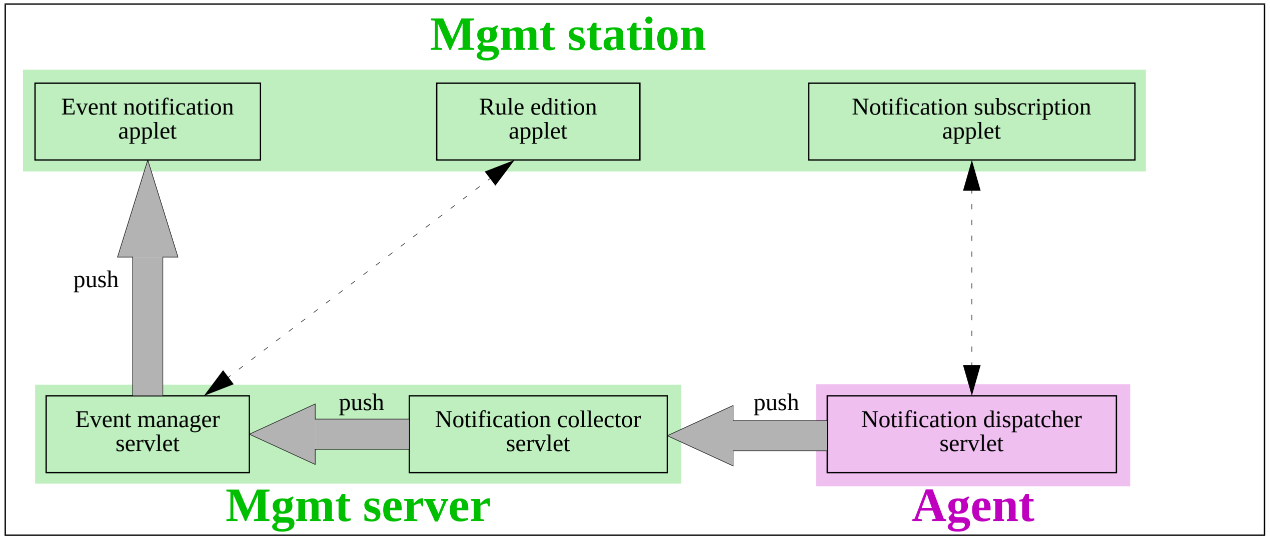


Outline

- Background
- Problems with SNMP-based mgmt
- Web-based mgmt
- Push model
- New communication model
- XML
- JAMAP: research prototype
- Conclusion

Overview of JAMAP


Mgmt station
Event notification applet
Rule edition applet
MIB data subscription applet
push
push
push
push
Event manager servlet
Pushed data collector servlet
MIB data dispatcher servlet
SNMP
MIB
Mgmt server
Agent


Mgmt station
Event notification applet
Rule edition applet
Notification subscription applet
push
push
push
Event manager servlet
Notification collector servlet
Notification dispatcher servlet
Mgmt server
Agent

JAMAP: A Research Prototype

- Purpose:
  - implement MIME multipart and MIME-based push
  - demonstrate the simplicity of the design:
    - ➠ the core of the communication was coded in only 2 man-weeks
- Main characteristics:
  - Java servlets on manager and agent sides
  - Java servlets communicate via HTTP on the manager
  - MIB data subscription applet uses AdventNet’s MIB browser
  - rule edition applet --> Java class dynamically loaded in
  - management data is compressed with `gzip`
- Many simplifications:
  - simplistic event correlator, only SNMP MIBs, no XML yet, etc.

Outline

- Background
- Problems with SNMP-based mgmt
- Web-based mgmt
- Push model
- New communication model
- XML
- JAMAP: research prototype
- Conclusion

Summary of Contributions

- A new management architecture (WIMA):
  - WIMA-push for regular management and notifications
  - WIMA-pull for *ad hoc* management (e.g., troubleshooting)
- A new communication model (WIMA-CM):
  - persistent HTTP/TCP connections
  - firewalls --> reversed manager and client roles
  - agent's infinite reply structured with MIME multipart
  - compress data with MIME content transfer encoding
  - two connections per agent: urgent vs. nonurgent data
  - timeouts and reconnections by the manager
- XML
- Proof of concept: JAMAP

Many Problems Have Been Solved

- BER no longer mandatory
- No max. message size
- Mgmt data can be compressed
- Intermediate levels of security: HTTP auth., SSL, etc.
- Important SNMP notifications less likely to be lost
- Hierarchical distributed mgmt with XML and HTTP
- Low-level and high-level semantics
- Mgmt platforms are more modular, less expensive
- Support for 3rd-party databases
- Web expertise is not domain specific
- Deal with firewalls
- Integration of SNMP and CIM

New Problems

- Reliability of new mgmt platforms based on COTS components and OO frameworks:
  - new means buggy
- Integration of components sold by multiple vendors:
  - it does not work: whose fault is it? who should fix it?
  - need integrators
- Synchronization of all clocks (managers, agents)
  - we can cope with timestamps and loose synchronization

Related Work (1/2)

- Architectures:
    - Bruins, Deri, Harrison *et al.*, Maston, Mullaney, Thompson, etc.

- Prototypes:
    - Marvel by Anerousis, Webbin by Barillaud *et al.*, CyberAgent by Burns and Quinn, EWS by Hong *et al.*, SUMO by Jocteur Monrozier *et al.*, WbASM by Kasteleijn, NetFinity by Reed *et al.*, etc.

- Compilation of commercial offerings:
    - `http://joe.lindsay.net/webbased.html`

Related Work (2/2)

- WBEM:
  - Microsoft *et al.* --> DMTF
  - HMMP --> HTTP + XML
  - new OO info. model: CIM
  - CIM-to-XML mapping at the meta level
  - extensions to HTTP: new headers for firewalls (compliance problem)
  - DMTF Working Groups are defining CIM schemas (ongoing)
- Java-based mgmt:
  - Sun Microsystems and the Java Community
  - OO mappings of existing info. models
  - communication via Java RMI (distributed OO)
  - ongoing: JMX (agent) and FMA (manager) are merging

Future Work

- Convince the DMTF and Sun Microsystems to adopt:
  - our push-based mgmt architecture
  - our comm. model based on persistent TCP conn. & MIME multipart
- Make JAMAP code freely available (GPL)
- Convince startups to develop WIMA-compliant mgmt servers
- Register the new MIME type `application/mgmt` with IANA
- SNMP-to-XML mapping: MIB level or meta level?
- Coexistence of SNMP MIBs and CIM schemas: What are the issues?
- Is WIMA suitable for application, service, and policy mgmt?