Using the Script MIB for Policy-based Configuration Management

T. Klie, S. Mertens,
J. Schönwälder, F. Strauß
Computer Science Department
Technical University Braunschweig
Mühlenpfordtstr. 23
38106 Braunschweig
Germany
{schoenw,strauss}@ibr.cs.tu-bs.de

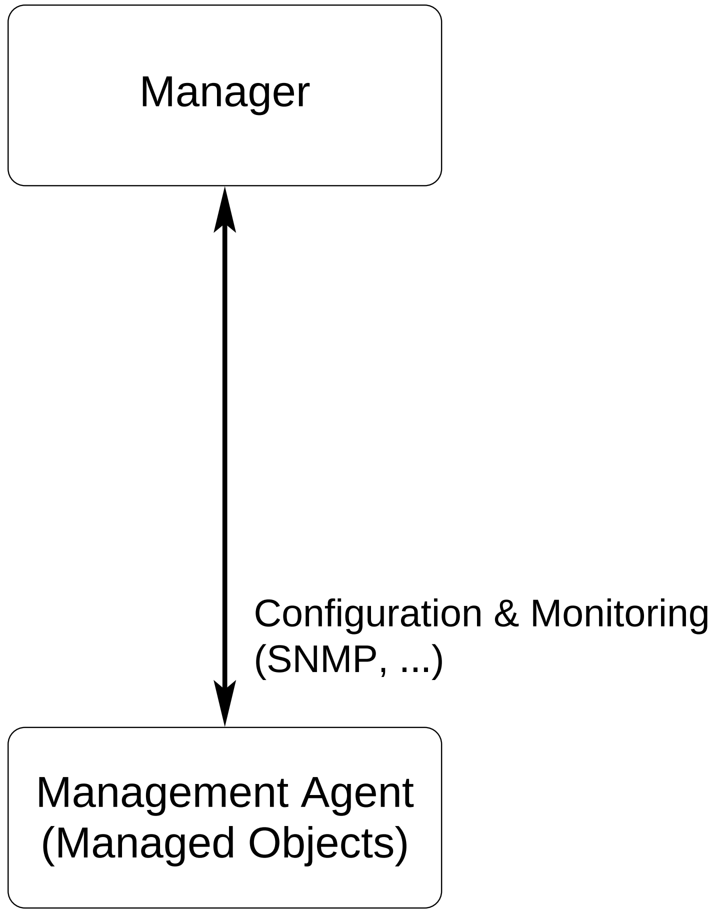
M. Brunner, P. Martinez,
J. Quittek
Network Laboratories
NEC Europe Ltd.
Adenauerplatz 6
69115 Heidelberg
Germany
{brunner,quittek}@ccrle.nec.de

jasmin-team@ibr.cs.tu-bs.de

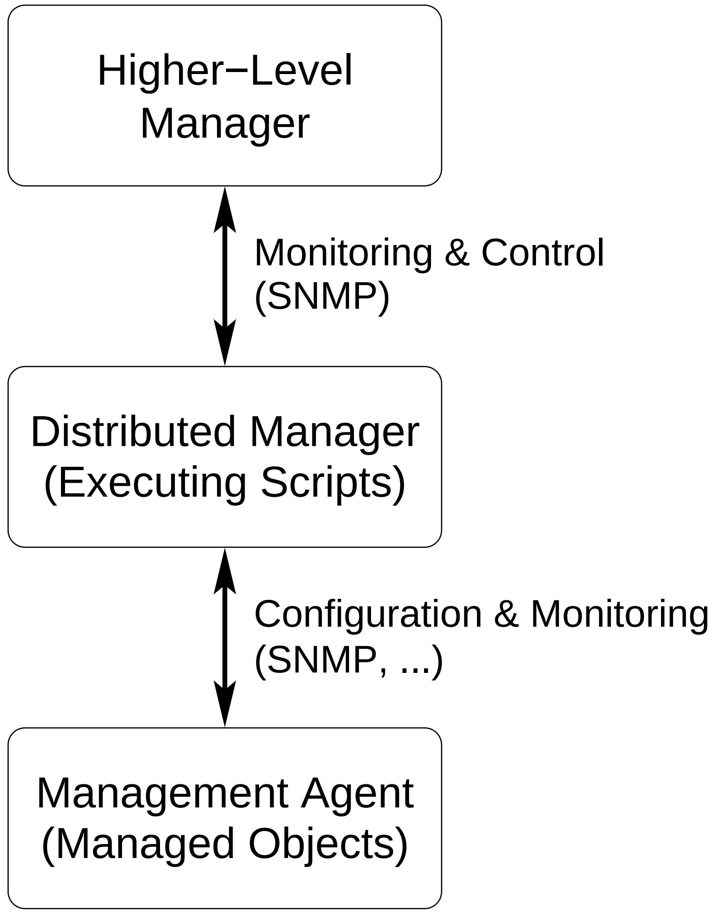
Outline

1. IETF Management-by-Delegation Architecture
   - The Script MIB
   - *Jasmin*: A Script MIB Implementation
2. IETF Policy Framework
3. Script MIB-based Policy Management
   - Policies as Programs
   - Policies as Objects
4. Conclusion

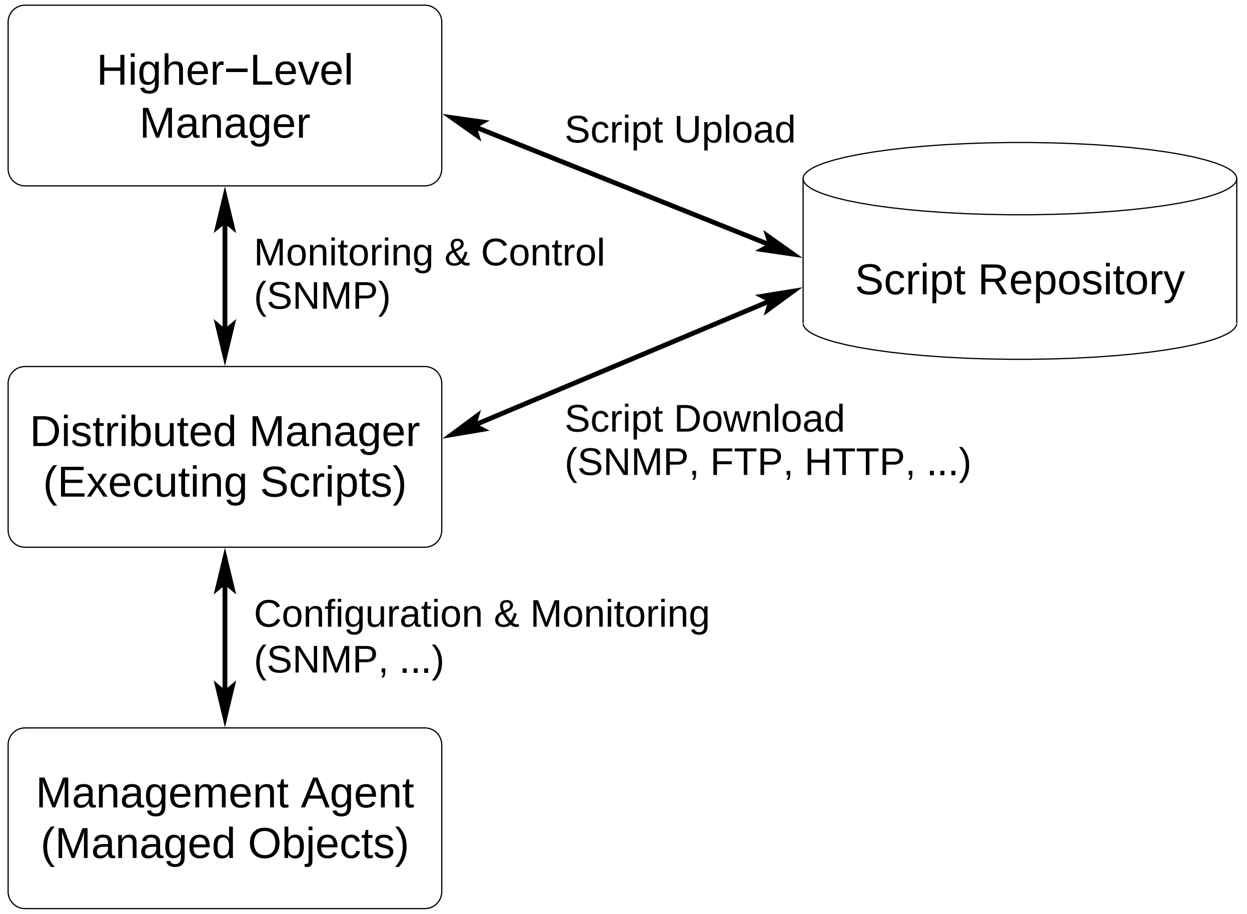
The Traditional Manager/Agent Architecture



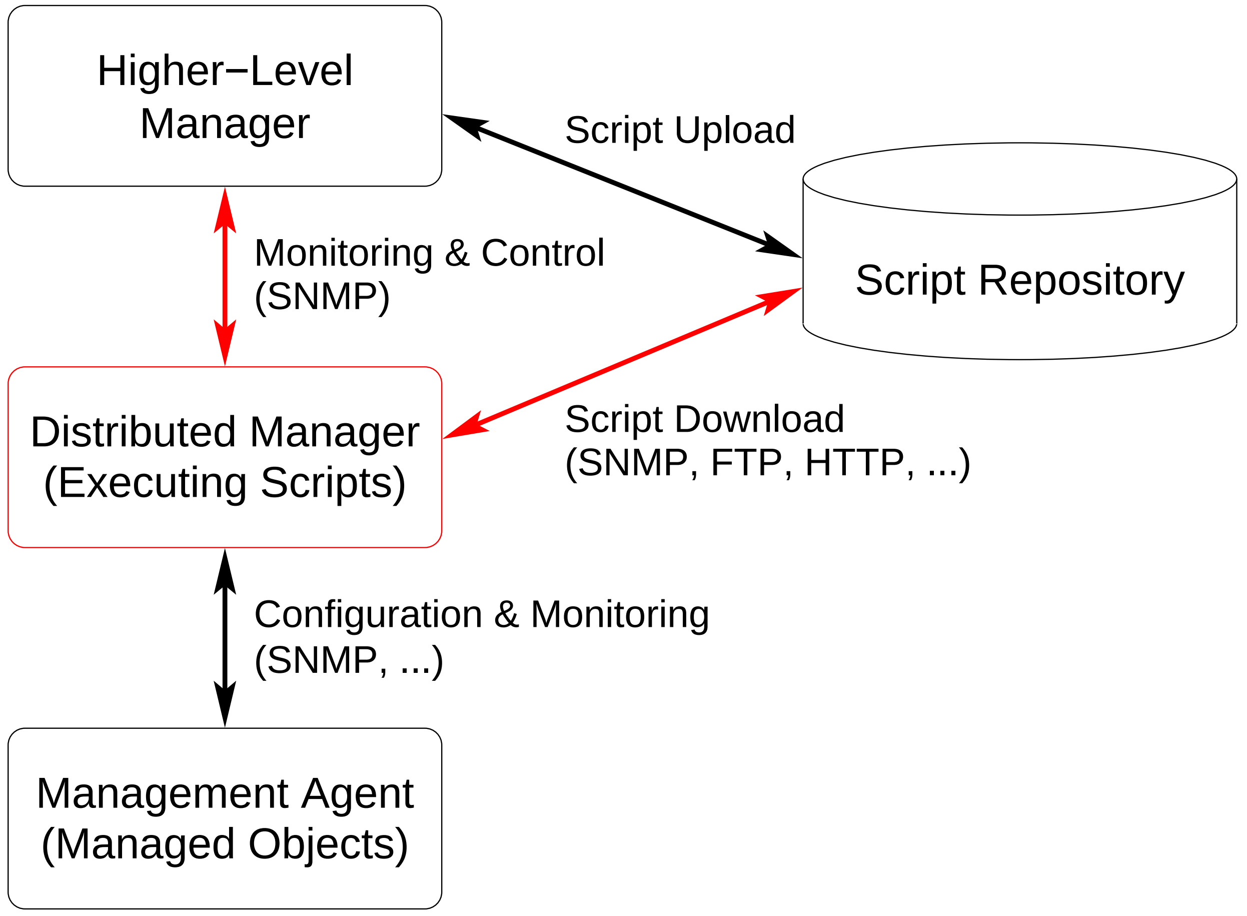
The IETF Management by Delegation (MbD) Architecture (I)


Higher−Level Manager
Monitoring & Control (SNMP)
Distributed Manager (Executing Scripts)
Configuration & Monitoring (SNMP, ...)
Management Agent (Managed Objects)

The IETF Management by Delegation (MbD) Architecture (II)


Higher−Level Manager
Script Upload
Script Repository
Monitoring & Control (SNMP)
Distributed Manager (Executing Scripts)
Script Download (SNMP, FTP, HTTP, ...)
Configuration & Monitoring (SNMP, ...)
Management Agent (Managed Objects)

What the IETF Script MIB Specifies


Higher−Level Manager
Script Upload
Script Repository
Monitoring & Control (SNMP)
Distributed Manager (Executing Scripts)
Script Download (SNMP, FTP, HTTP, ...)
Configuration & Monitoring (SNMP, ...)
Management Agent (Managed Objects)

The IETF DISMAN Script MIB

- Designed and standardized by the IETF Distributed Management (DISMAN) Working Group
- First Proposed Standard: RFC 2592, May 1999
- Updated Proposed Standard: RFC 3165, August 2001
- Supported functions:
  - Information on supported script languages and extensions
  - Transfer of scripts to a distributed manager
  - Control execution of management scripts
  - Retrieve results from management scripts
- Security based on:
  - SNMPv3 security (USM and VACM)
  - Script runtime engine security models (sandbox)

The Jasmin Project

- Joint project (1998 – 2001):
  - Technical University of Braunschweig
  - Network Laboratories, NEC Europe Ltd.
- Goals of the project:
  - Evaluate and enhance the Script MIB Standard
  - Provide a proto-type implementation
  - Study use-cases and develop supporting tools
- Primary outcome of the project:
  - a flexible open source Script MIB agent implementation
  - supporting various runtime engines (currently Java, Tcl, Perl) via the *Script MIB Extensibility Protocol* (SMX), RFC 3179
- In 2000 demand for policy-based configuration management increased

  → How could the Script MIB support this?

Policy-based Configuration Management

- Motivation:
  - Traditional management of individual device-specific configurations is
    - ∗ complex and error-prone (different vendors means different ways)
    - ∗ too static (state configuration, no behavior configuration)
  - The general policies behind those configurations are often simple
- Consequence:
  - Let the administrator configure just those policies
  - PBMS supports automated enforcement of the policies

General Concept of Policies

- A *Policy* is represented by a number of *Rules*
- Each rule consists of a *Condition* and an *Action*
- The evaluation of a rule is triggered by an *Event*

`on` <*event(s)*> `if` <*condition*> `do` <*action(s)*>

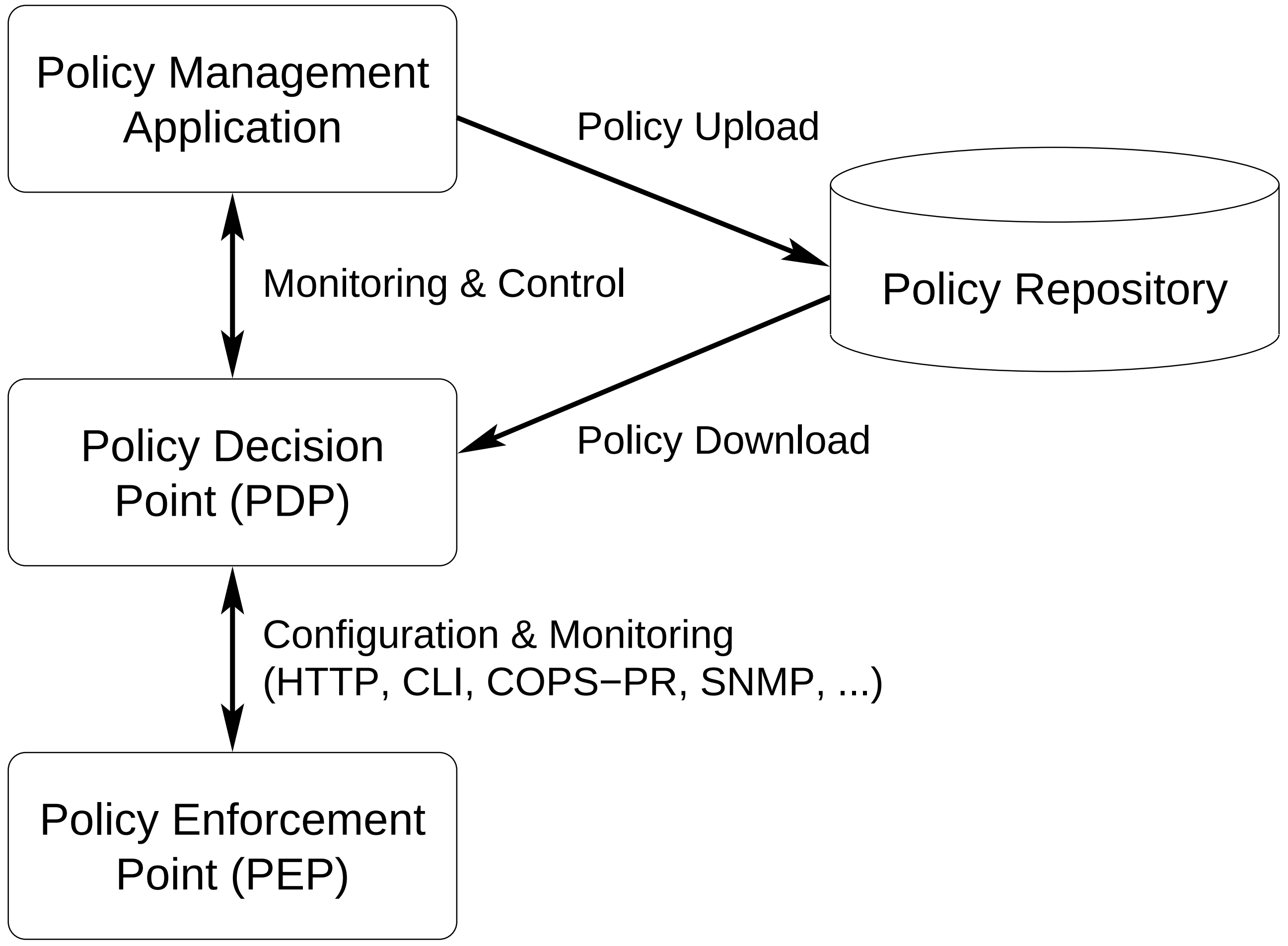
Approaches to express policies:

- Specific policy definition language, e.g. PONDER
- Traditional programming language & language extension for policies
- Policy Core Information Model (PCIM)

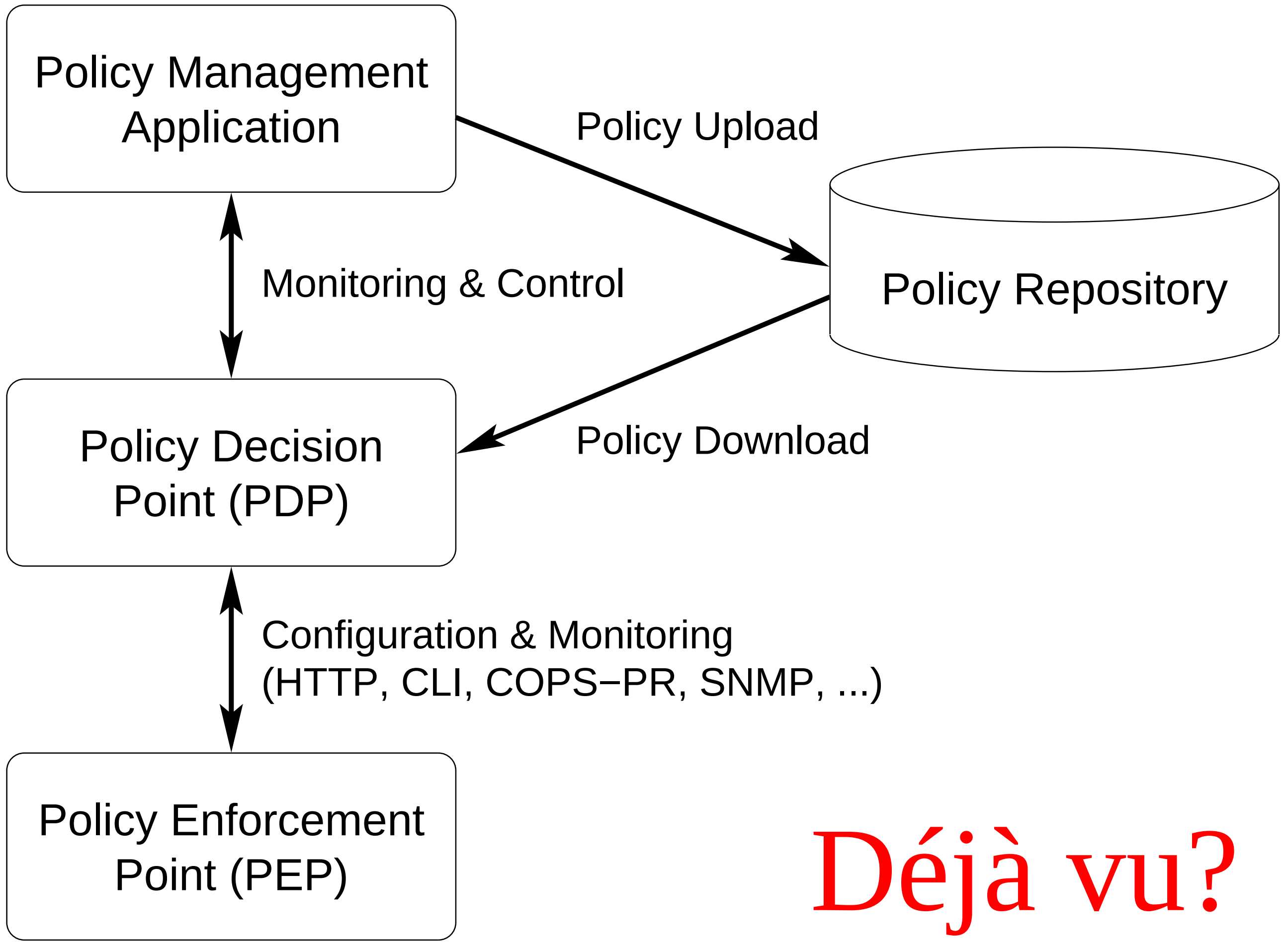
An infrastructure is required:

- Policies must be distributed over the network
- Policies must be interpreted
- Managed devices must be configurable

The IETF Policy-based Management Framework

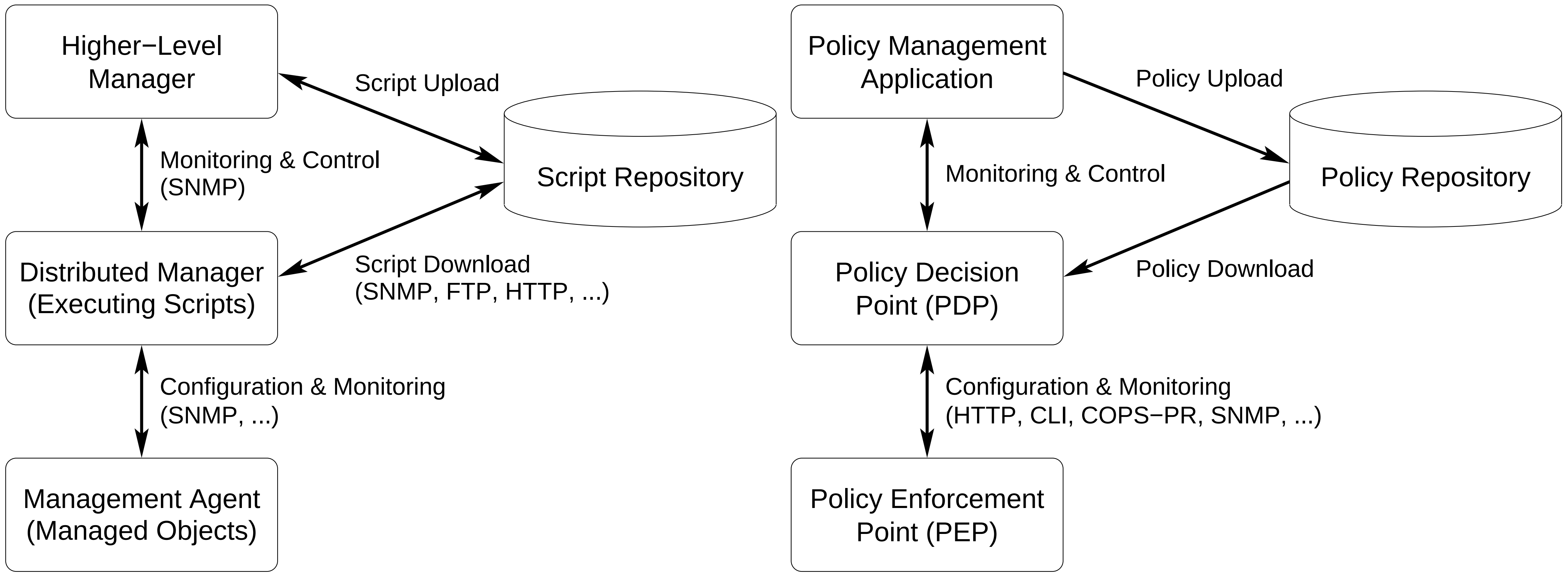

Policy Management Application
Policy Upload
Policy Repository
Monitoring & Control
Policy Decision Point (PDP)
Policy Download
Configuration & Monitoring (HTTP, CLI, COPS−PR, SNMP, ...)
Policy Enforcement Point (PEP)

The IETF Policy-based Management Framework

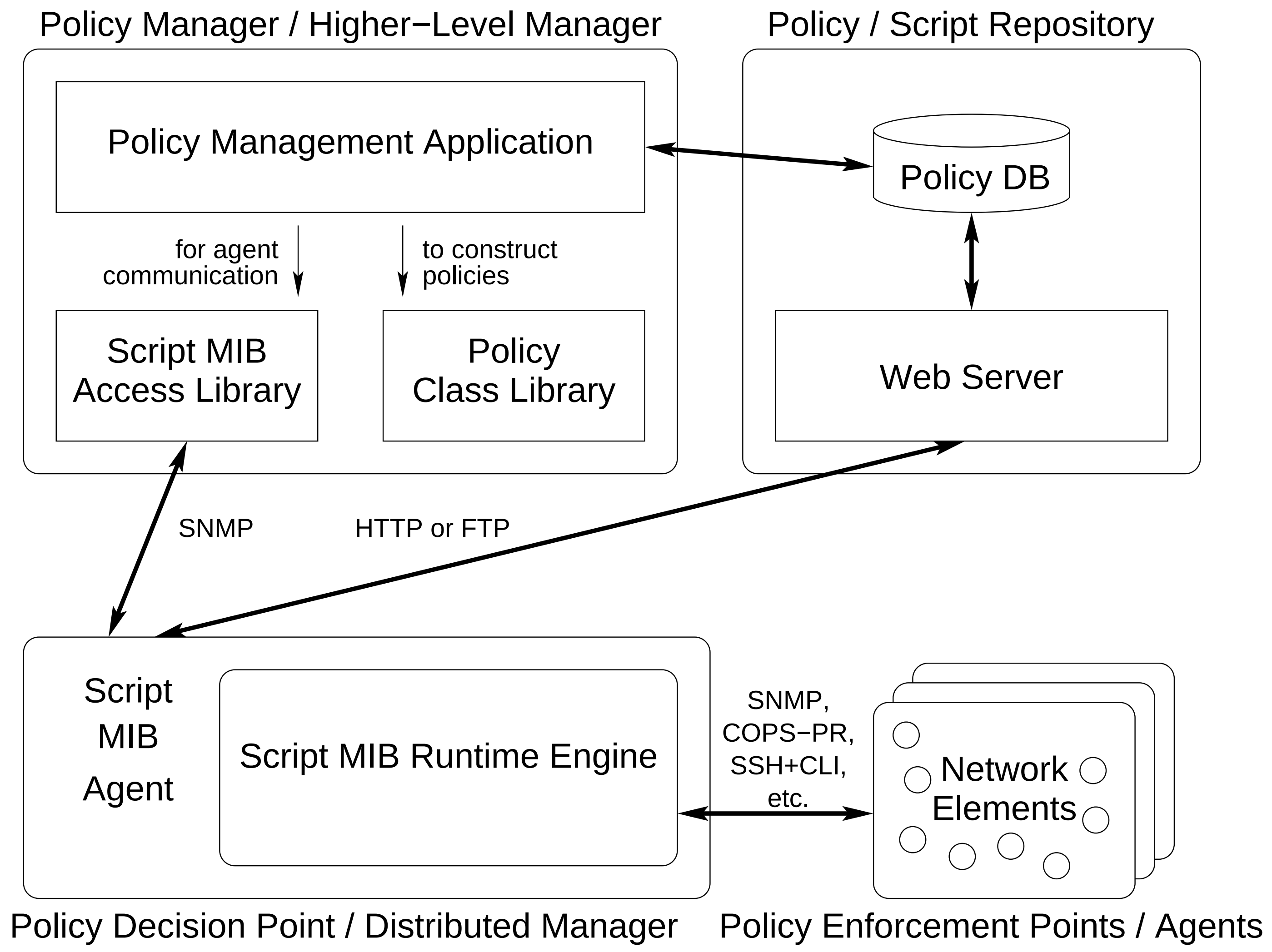

Policy Management Application
Policy Upload
Policy Repository
Monitoring & Control
Policy Decision Point (PDP)
Policy Download
Configuration & Monitoring (HTTP, CLI, COPS−PR, SNMP, ...)
Policy Enforcement Point (PEP)

Déjà vu?

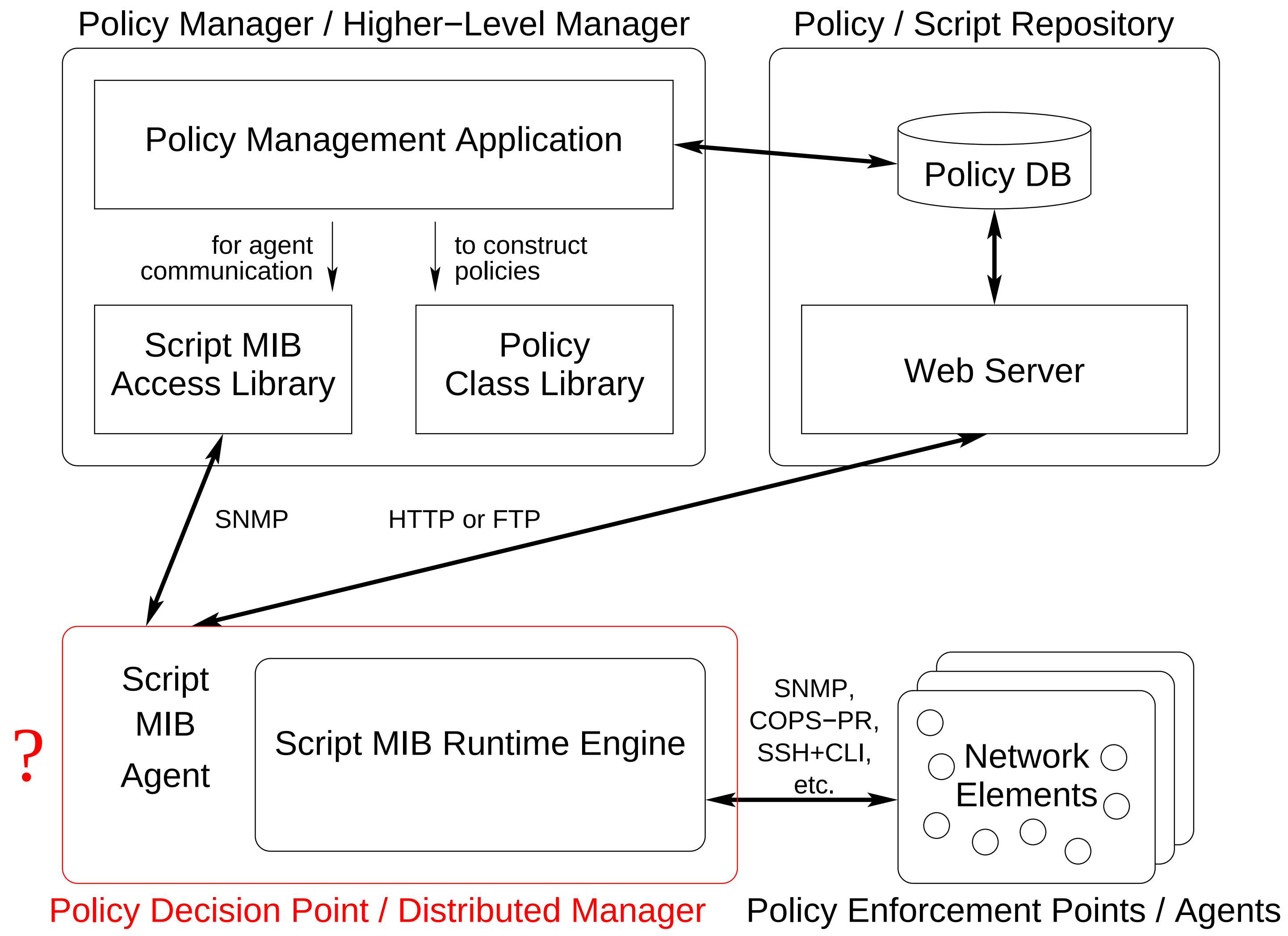
Management-by-Delegation vs. Policy-based Management


Higher−Level Manager
Script Upload
Script Repository
Monitoring & Control (SNMP)
Distributed Manager (Executing Scripts)
Script Download (SNMP, FTP, HTTP, ...)
Configuration & Monitoring (SNMP, ...)
Management Agent (Managed Objects)
Policy Management Application
Policy Upload
Policy Repository
Monitoring & Control
Policy Decision Point (PDP)
Policy Download
Configuration & Monitoring (HTTP, CLI, COPS−PR, SNMP, ...)
Policy Enforcement Point (PEP)

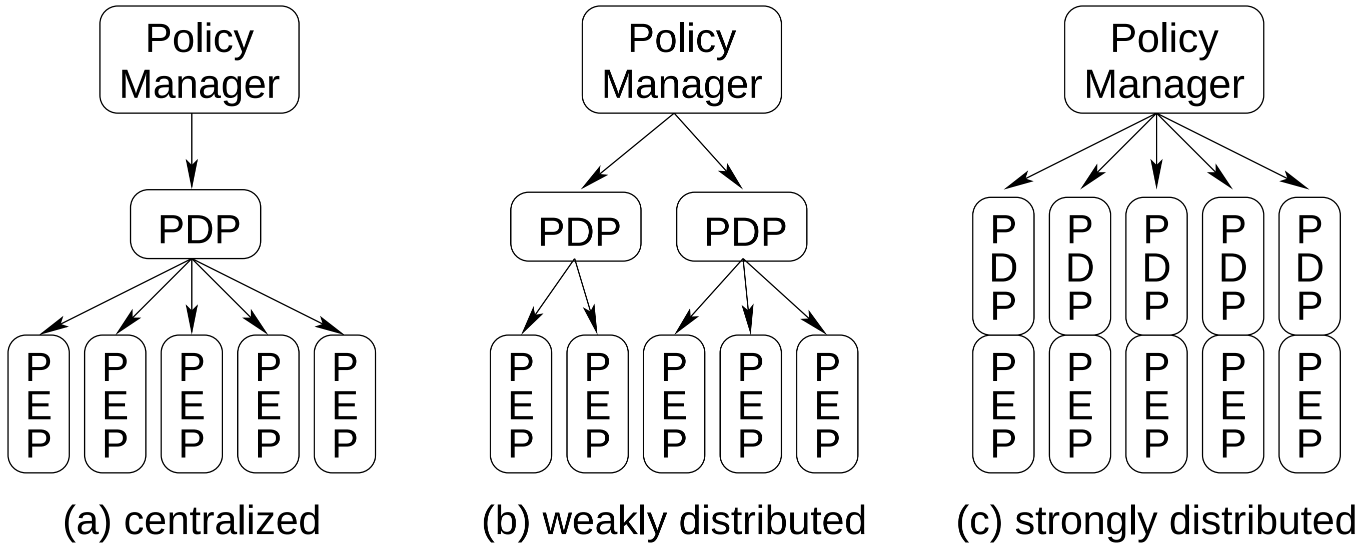
Architecture of the Jasmin Policy-based Management System


Policy Manager / Higher−Level Manager
Policy / Script Repository
Policy Management Application
Policy DB
for agent communication
to construct policies
Script MIB Access Library
Policy Class Library
Web Server
SNMP
HTTP or FTP
Script MIB Agent
Script MIB Runtime Engine
SNMP, COPS−PR, SSH+CLI, etc.
Network Elements
Policy Decision Point / Distributed Manager
Policy Enforcement Points / Agents

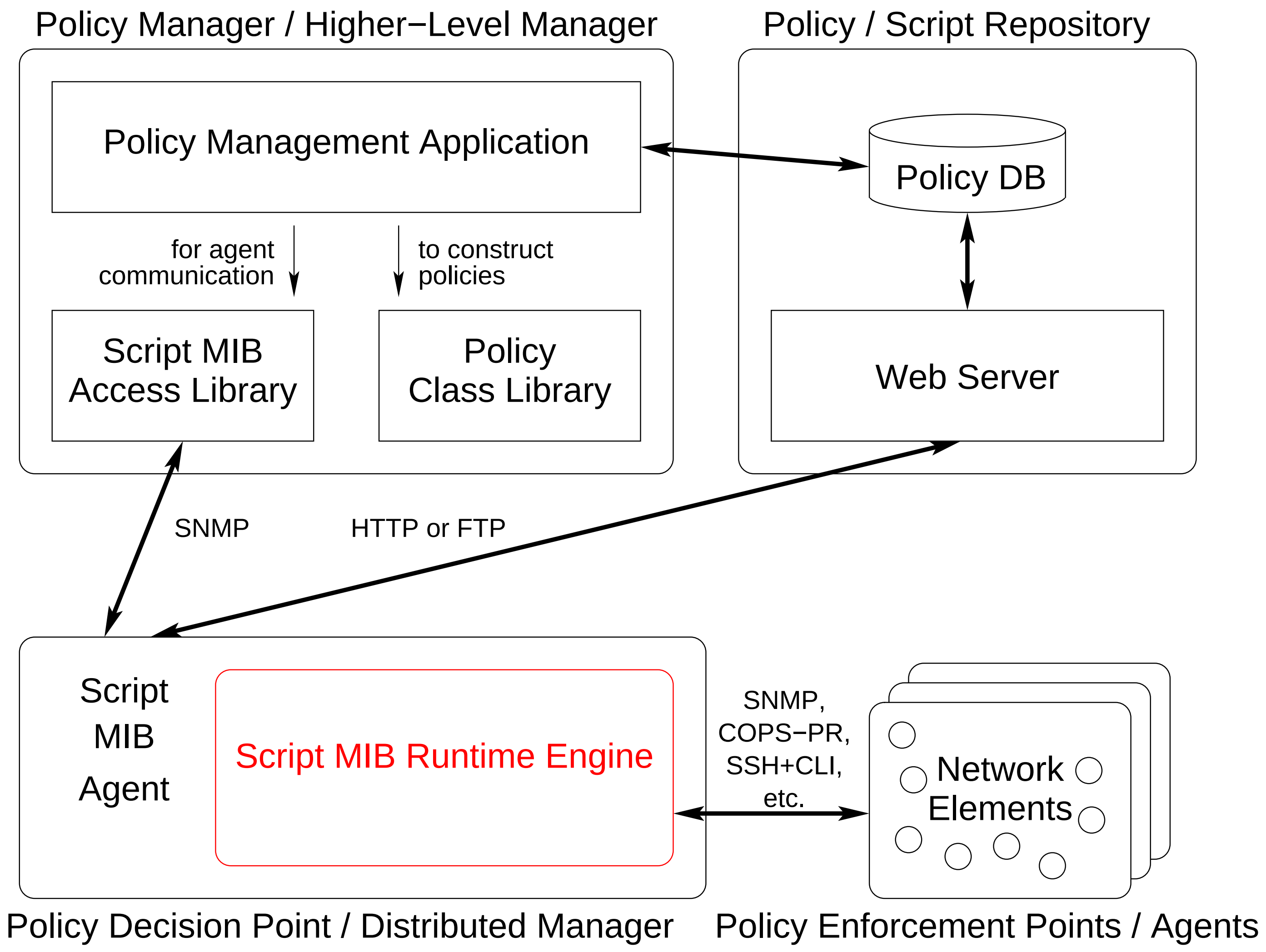
Architecture of the Jasmin Policy-based Management System


Policy Manager / Higher−Level Manager
Policy / Script Repository
Policy Management Application
Policy DB
for agent communication
to construct policies
Script MIB Access Library
Policy Class Library
Web Server
SNMP
HTTP or FTP
?
Script MIB Agent
Script MIB Runtime Engine
SNMP, COPS−PR, SSH+CLI, etc.
Network Elements
Policy Decision Point / Distributed Manager
Policy Enforcement Points / Agents

Different Levels of PDP Distribution



Architecture of the Jasmin Policy-based Management System


Policy Manager / Higher−Level Manager
Policy / Script Repository
Policy Management Application
Policy DB
for agent communication
to construct policies
Script MIB Access Library
Policy Class Library
Web Server
SNMP
HTTP or FTP
Script MIB Agent
Script MIB Runtime Engine
SNMP, COPS−PR, SSH+CLI, etc.
Network Elements
Policy Decision Point / Distributed Manager
Policy Enforcement Points / Agents

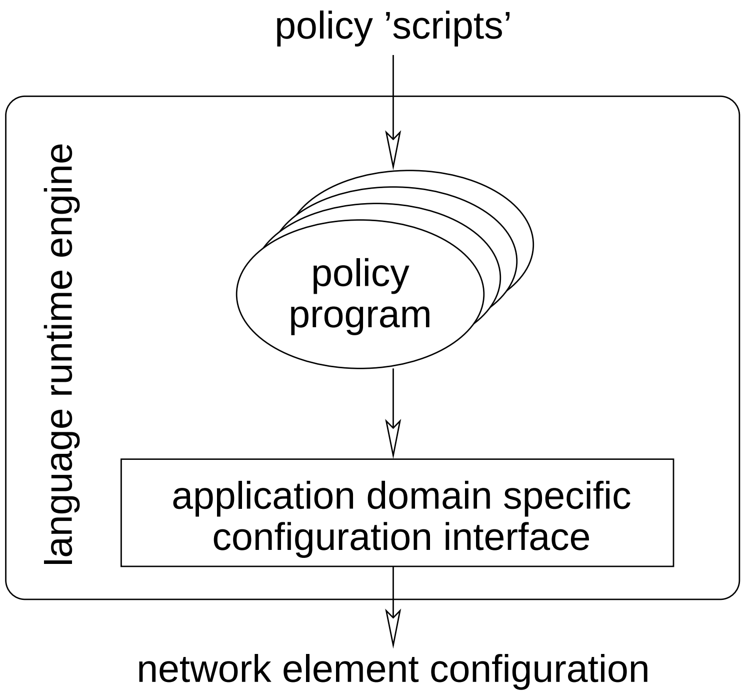
Two Approaches

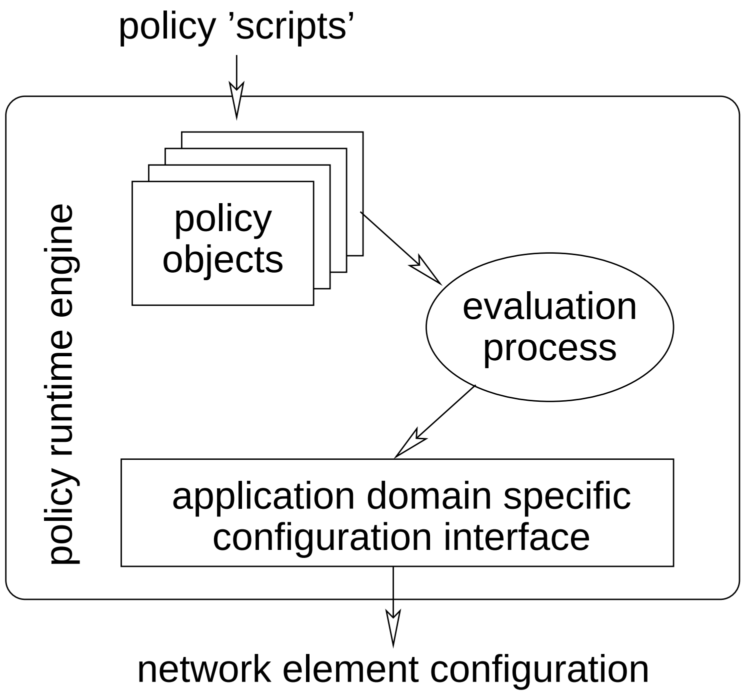
Policies as Programs

– Each policy is implemented as a Java program
– Programs run independently and concurrently

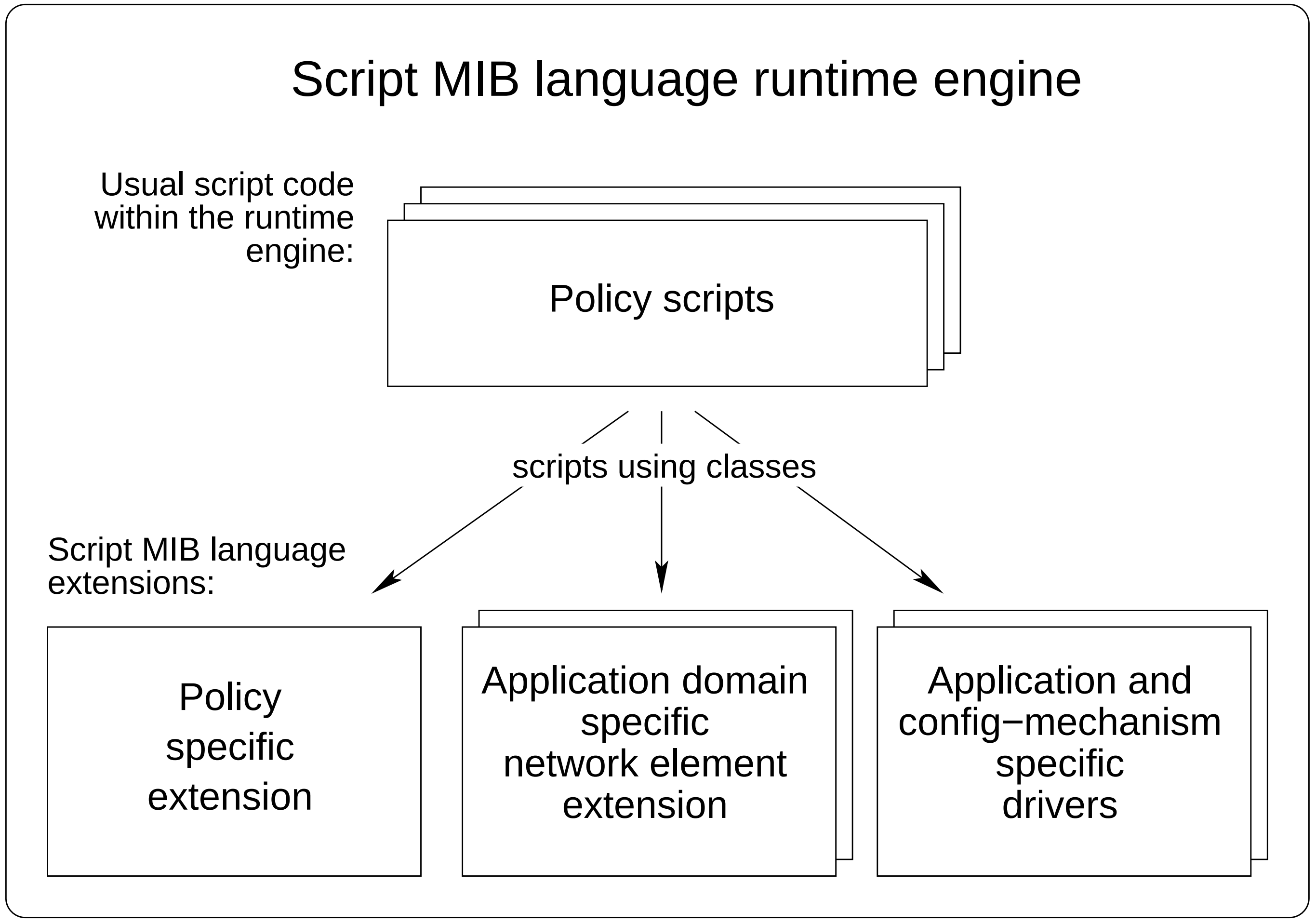
Policies as Objects

– Each policy is implemented by a set of PCIM objects
– All policy objects are evaluated by the same process


policy 'scripts'
language runtime engine
policy program
application domain specific configuration interface
network element configuration


policy 'scripts'
policy runtime engine
policy objects
evaluation process
application domain specific configuration interface
network element configuration

Policies as Programs

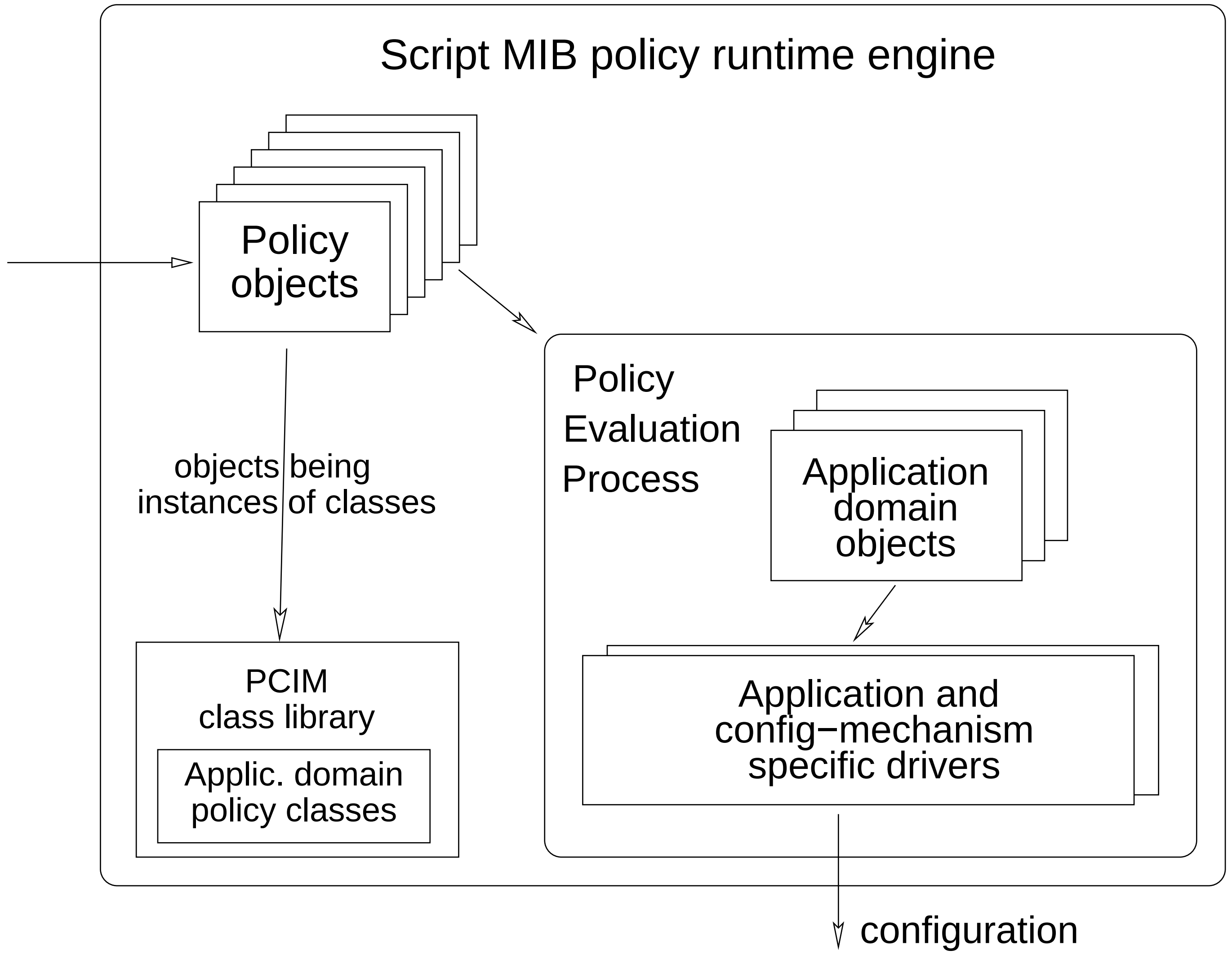

Script MIB language runtime engine
Usual script code within the runtime engine:
Policy scripts
scripts using classes
Script MIB language extensions:
Policy specific extension
Application domain specific network element extension
Application and config−mechanism specific drivers

Policies as Programs: Prototype Implementation

- `policyMgmt` Java class package
  - Classes: `Policy`, `Rule`, `Timer`, ...
  - Interfaces: `Condition`, `Action`, `EventGenerator`, `Driver`
- `diffServ` Java class package
  - Classes: `Classifier`, `Queue`, `Scheduler`, `RandomDropper`, ...
  - Driver Implementations: `JtcDriver`
- `jtc` Java class package for the Linux DiffServ implementation
  - Classes: `DSMarkQDisc`, `DSMarkClass`, `TCIndexFilter`, ...
- Simple examples

See the paper for details and the example

Policies as Objects


Script MIB policy runtime engine
Policy objects
objects being instances of classes
PCIM class library
Applic. domain policy classes
Policy Evaluation Process
Application domain objects
Application and config−mechanism specific drivers
configuration

Policies as Objects: Prototype Implementation

- Policy runtime engine
  - `PolicyEvaluator` class
  - Replaced class loader for serialized objects
- Java classes implementing PCIM objects
- Java classes implementing QPIM objects
- DiffServ class package
- Interface to the Linux DiffServ implementation from Uni Bern and NEC
- Simple examples

See the paper for details and an example

Conclusions

- Though originally designed for Distributed Management the Script MIB is well-suited as a Policy-based Configuration Management infrastructure
- No need to re-invent
  - a PDP internal architecture
  - a policy transfer protocol
  - a PDP control protocol
  - a PDP-PEP protocol
  - a security model
- Depending on the chosen approach, it can be

– *cheap*	using the existing Script MIB and runtime infrastructure, while policy scripts become more complex
– *standards based*	applying the PCIM and using a special policy runtime engine
– *user friendly*	using a policy definition language (not implemented by us)

Thank You!

Q & A