An Extension of XACML to Improve the Performance of Decision Making Processes when Dealing with Stable Conditions

Romain Laborde, Thierry Desprats

21-22 October, 2008

Outline

- Introduction to XACML
  - Policy language
  - Architecture
- Scenario
- Definition of Stable Conditions
- Improvement of the XACML architecture
- Experiments
- Conclusion & Future works

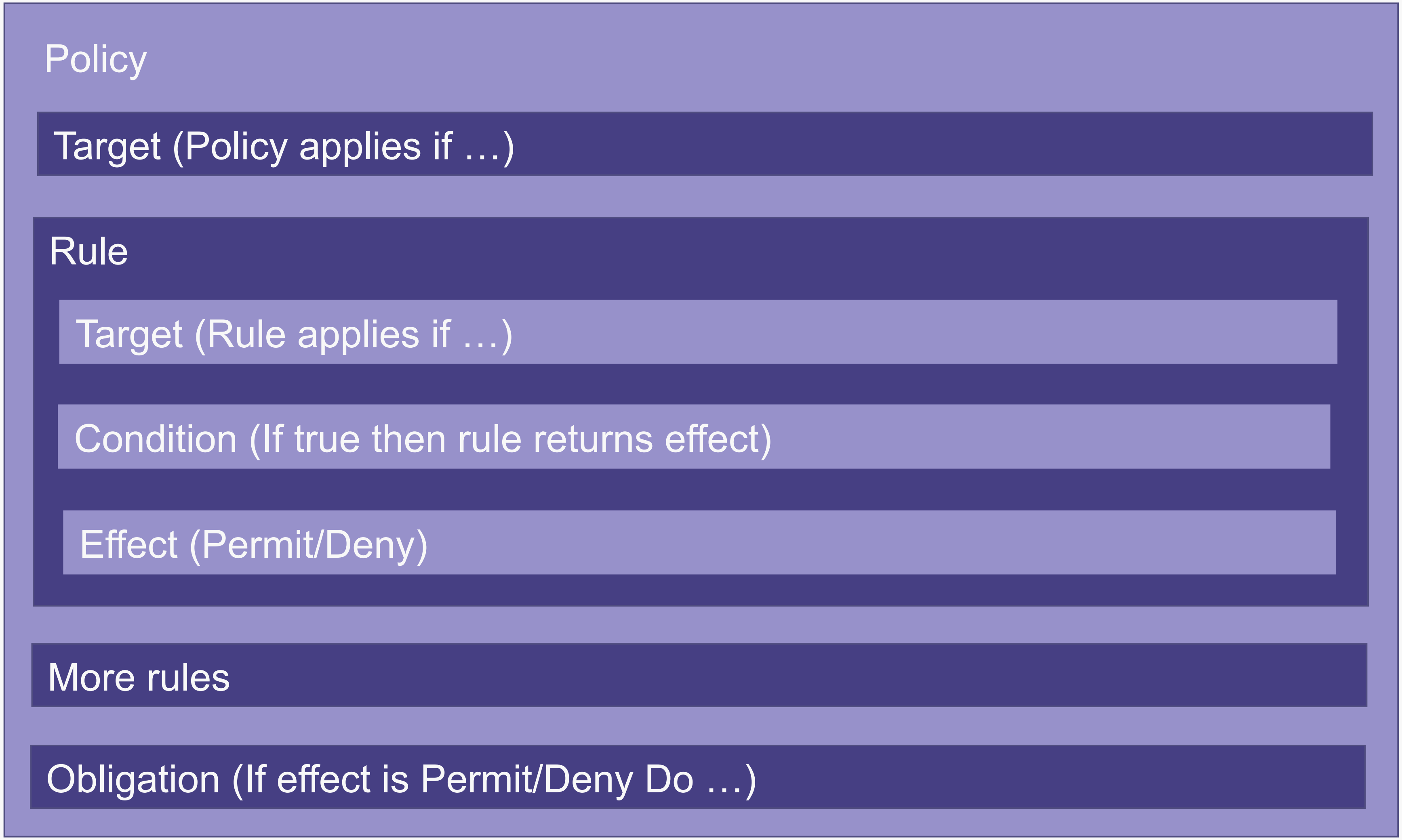
XACML

- OASIS Standard (Organization for the Advancement of Structured Information Standards)
    - eXtensible Access Control Markup Language
    - Based on XML
- Access control policy language
    - Attribute based access control
- Access control management architecture
    - Policy Based Management
- Protocol (Request/Decision)

XACML Policies

- Attribute Based Access Control
  - Four objects:
    - Subject
    - Resource
    - Action
    - Environment
  - Attribute
    - any security relevant characteristics of requestors, actions, resources, and environment
  - Example
    - role of the subject, name of the action, type of resource, etc.

XACMLv2 policies



Policy

Target (Policy applies if …)

Rule

Target (Rule applies if …)

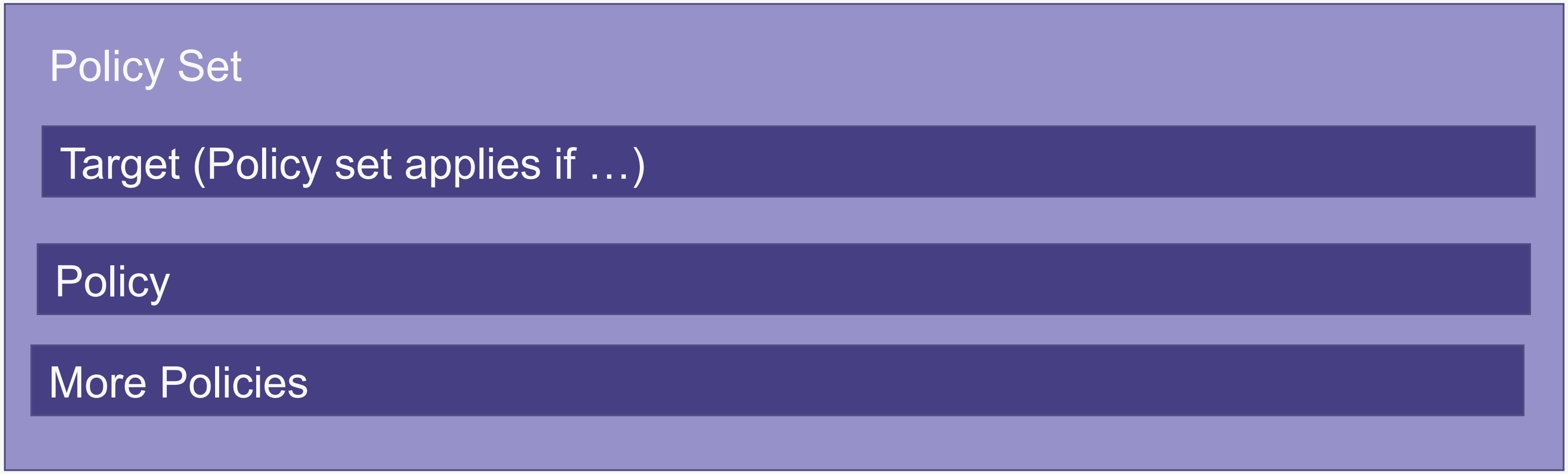
Condition (If true then rule returns effect)

Effect (Permit/Deny)

More rules

Obligation (If effect is Permit/Deny Do …)

XACMLv2 policies set



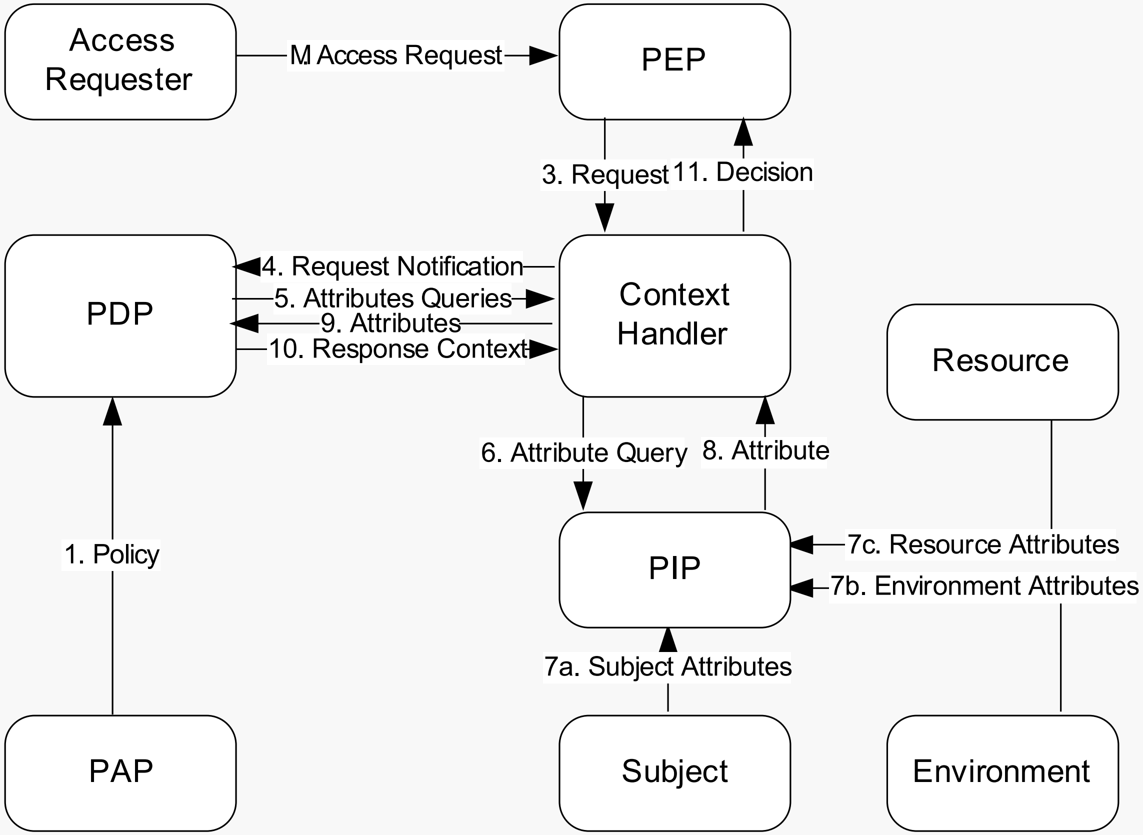
Policy Set

Target (Policy set applies if …)

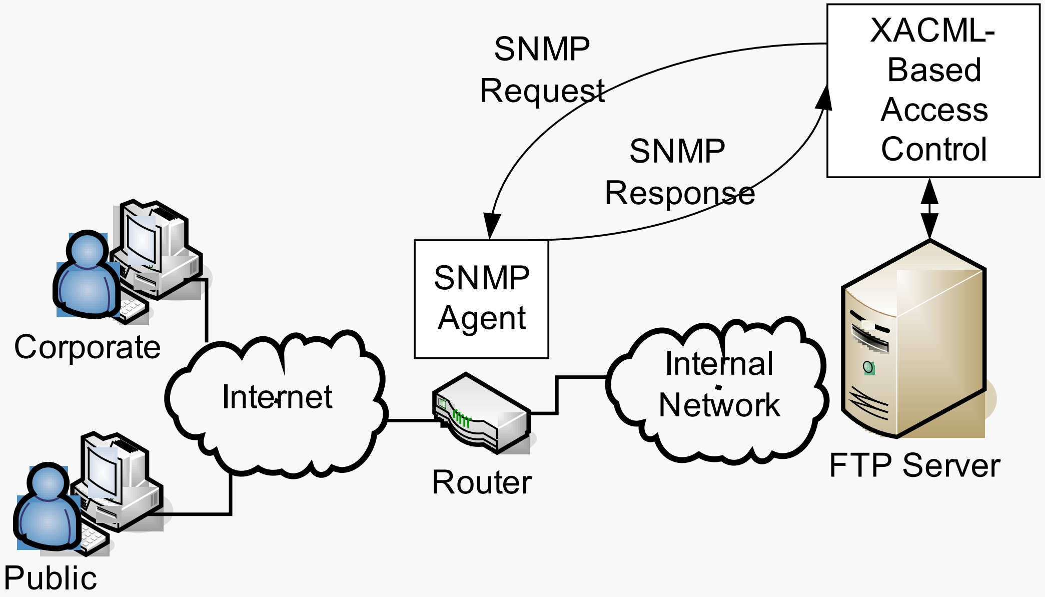
Policy

More Policies

XACML Architecture

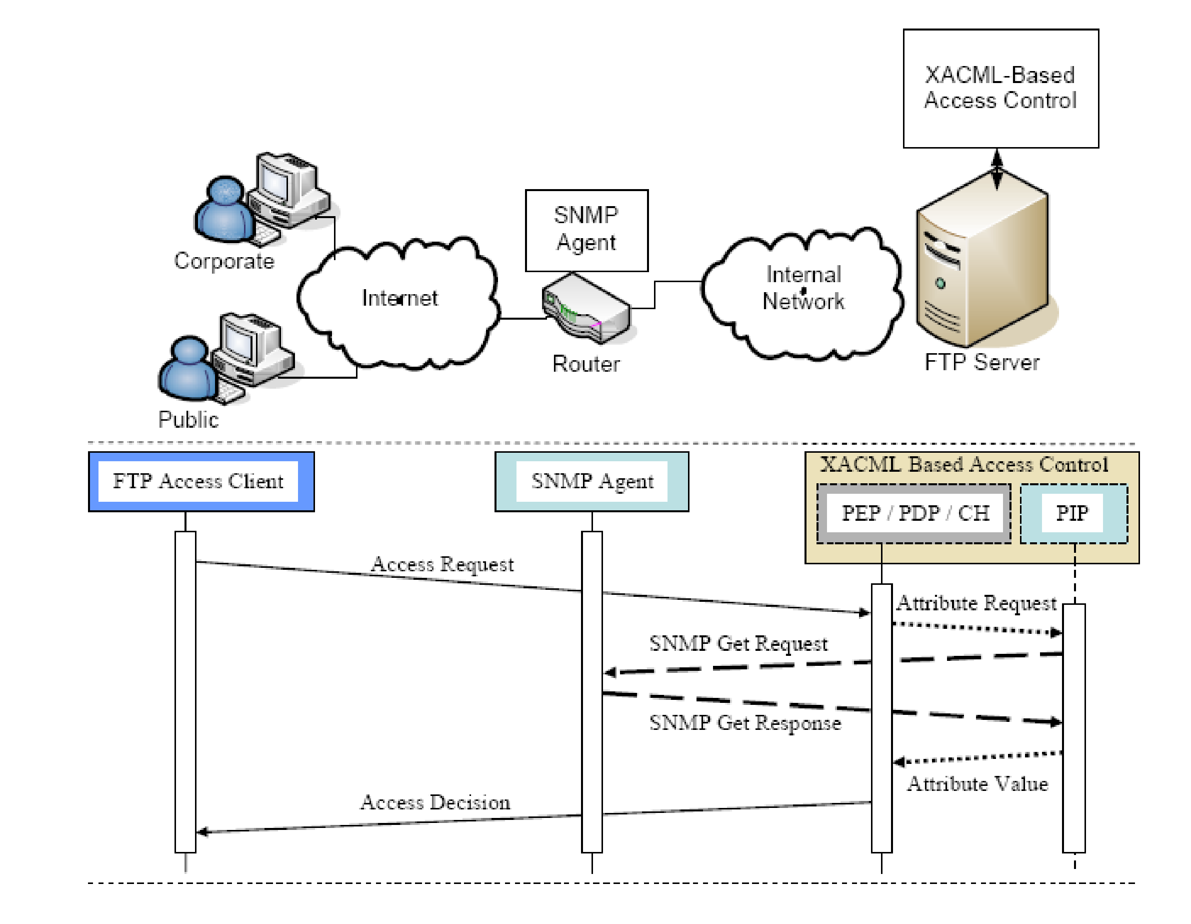


Scenario



Policy:

1) role(S) = corporate ∧ name(R) = ftp://ftp.example.com/private => Permit
2) name(R) = ftp://ftp.example.com/public ∧ BW(E) < 60% => Permit
3) Else => Deny



Policy:

1) role(S) = corporate ∧ name(R) = ftp://ftp.example.com/private => Permit
2) name(R) = ftp://ftp.example.com/public ∧ BW(E) < 60% => Permit
3) Else => Deny

Stable Conditions

- Descriptive definition
  - A stable condition can be viewed as an expression that always returns the same result during a given period considered to be long.

- Characterization (*eligible stable condition*)
  - A stable condition is an expression where every argument does not directly or indirectly depend on the value of one of the intrinsic attributes of the request.

- Request intrinsic attributes
  - the attributes sent by the PEP to the Context Handler in an authorization request
  - Examples: Subject’s role, name of the resource, etc.
- Request extrinsic attributes
  - Attributes which do not depend on the request itself
  - Examples: Bandwidth, time, network intrusion

Stable conditions processing

- Our idea:
  - Remove stable conditions from policies
  - Notify when the value returned by a stable condition has changed
  - Modify the policy according this changing

- Example:

  1) role(S) = corporate ∧ name(R) = ftp://ftp.example.com/private => Permit

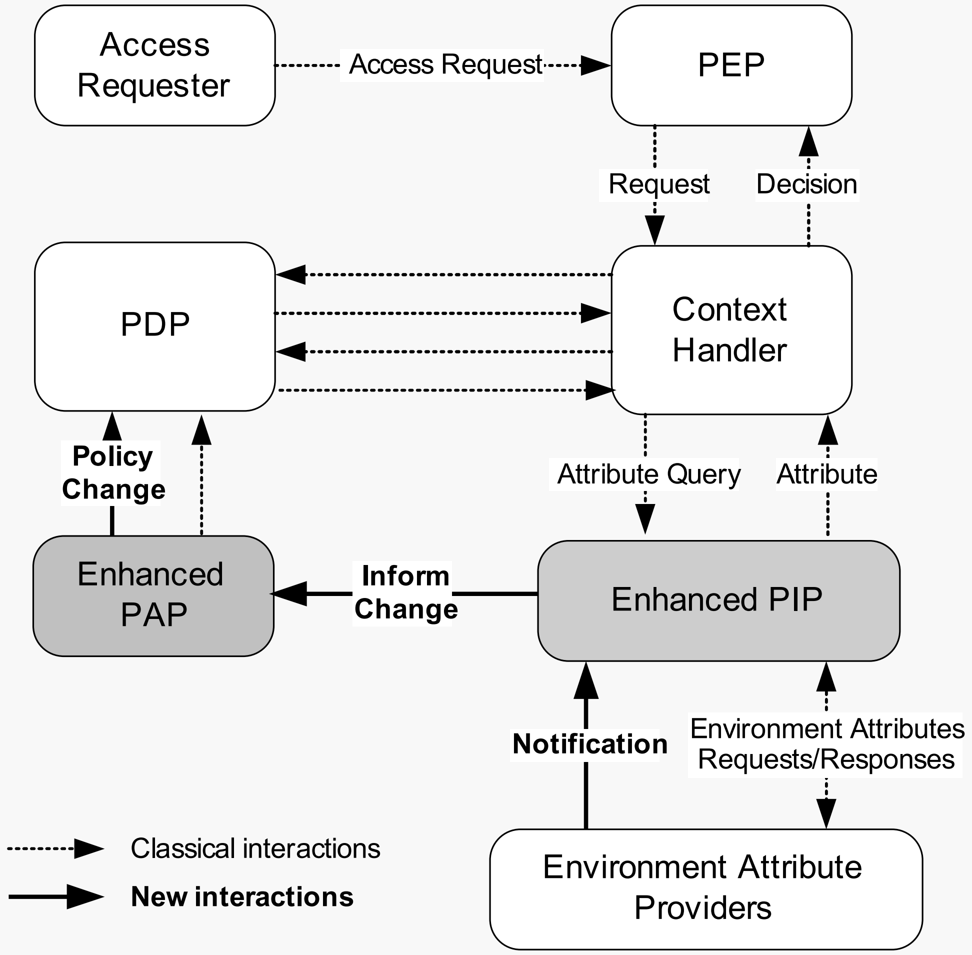
  2) name(R) = ftp://ftp.example.com/public ∧ => Permit

  2) name(R) = ftp://ftp.example.com/public => Deny

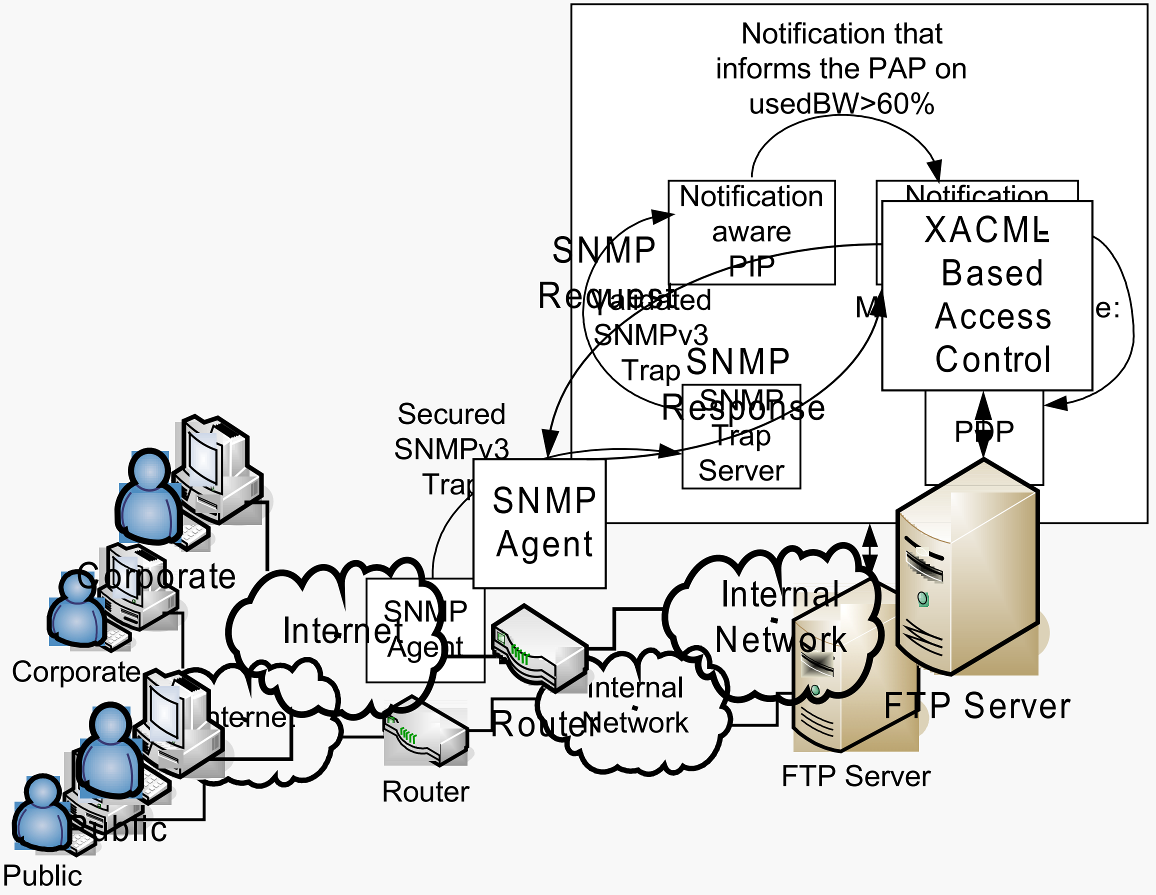
  BW(E) < 60% => Permit

  3) Deny

Modification of the XACML Architecture



Impact on our scenario

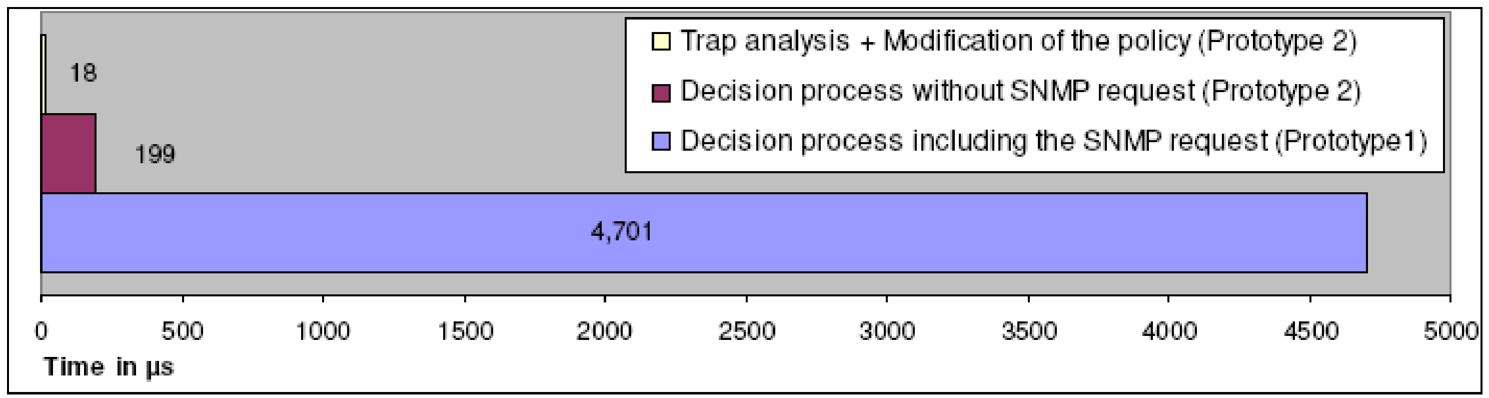


Our testing environment

- Test
  - Time to make a decision for the request “a user wants to access the public directory ftp://ftp.example.com/public”
  - 5 times 100 requests

- Router
  - PC Pentium Core 2 Duo 2.13GHz, 1Gbyte RAM
  - Linux Kubuntu DAPPER 6.06.1 LTS
  - NET-SNMP version 5.2.1.2 for the SNMP agent et sending SNMP traps
- FTP server
  - PC core 2 Duo 1.66 GHz, 1Gbyte RAM and Windows XP Pro
  - Sun’s XACML implementation version 1.2 (PDP and java API for PEPs, PIPs and PAPs)
  - SNMP4J java API version 1.8.2 for the SNMP client and the SNMP traps server
- Network
  - Ethernet 100Mbps
  - No Routing !

Results



- Evaluation
  - 23 faster without looking at the MIB
- Modification of the policy represents:
  - 0.3% of the evaluation process when looking at the MIB
  - 8.7% of the evaluation process when not looking at the MIB
- Network
  - Consulting the MIB = 2 SNMP messages/decision
  - Notification approach = 1 SNMP trap message when needed

Conclusion

- All the attributes should not be considered and processed in the same way
  - Concept of stable conditions
- Notification approach in the XACML architecture
  - Extended XACML architecture to deal with stable conditions
- Experiments

Future works

- Long term objective = self-optimization behaviour

- We have to :
  - Automatic detection of stable conditions
  - Management of policy modifications
    - Modify of the policy and keep it correct according to the original one
    - Make this process as light as possible
  - Dialogue between Policy Information Points and Extended Attributes Providers
  - …

Thank you …