Policy Machine

Tailoring access control policies and data services to missions

David Ferraiolo, Serban Gavrila
National Institute of Standards and Technology

dferraiolo@nist.gov

The Policy Machine (PM)

A logical “machine” comprising:

- a fixed set of data and relations used to express (combinations of) access control policies and delivery of data services
- a fixed set of administrative operations for configuring the data and relations
- a fixed set of functions for making access control decisions and enforcing the policies

PM Data & Relations

- Basic elements
  - Users, processes, operation, and objects
- Containers
  - User attributes, object attributes, and policy classes
- Relations
  - Assignments (defining membership in containers)
  - Associations (defining privileges)
  - Prohibitions (denies for users and processes)
  - Obligations (Event/Response)

Unification through Privileges

- Through proper representation privileges (user, operation, object) appear and disappear at three levels
- Level 1: Privileges are specifications of users that can perform read and/or write operations on data objects.
- Level 2: Users are administrators that can perform administrative operations on the data elements and relations resulting in privileges of first and second levels. The users in the second level may include those of the first.
- Level 3: Capabilities in the third level are arbitrary, but data service specific. Although arbitrary, capabilities are composed of sequences of operations of the second level that act upon objects of the first and second levels. These sequences are executed by authorized users of the second level, as commands, or automatically, in response to events, by the PM as obligations.

Some Benefits
(demonstrated by prototype)

- Access Control as a Data Service operate as one
- Through a single authenticated session, users are offered capabilities of file management (to include support for office applications), email, workflow, and forms and records management. Others services could be accommodated as well.
- Select capabilities (of different services) are delivered to select users, under combinations of arbitrary, but **mission tailored** forms of **discretionary**, **mandatory**, and **history-based** (event driven) access controls.
- Data services naturally interplay.
- Data is naturally protected across services.

Prohibitions (Denies)

- User denies
  - u-deny(*u*, *opset*, *oset*}). Any process executing on behalf of user *u* cannot perform any operation in *opset* on any object in *oset*.
- Process denies
  - p-deny(*p*, *opset*, *oset*). Process *p* cannot perform any operation in *opset* on any object in *oset*.

Computing an Access Decision

A process access request $<op, o>_p$ is granted if and only if there is a PM privilege (u, op, o) where u is process p's user, and (op, o) is not denied to u or p.

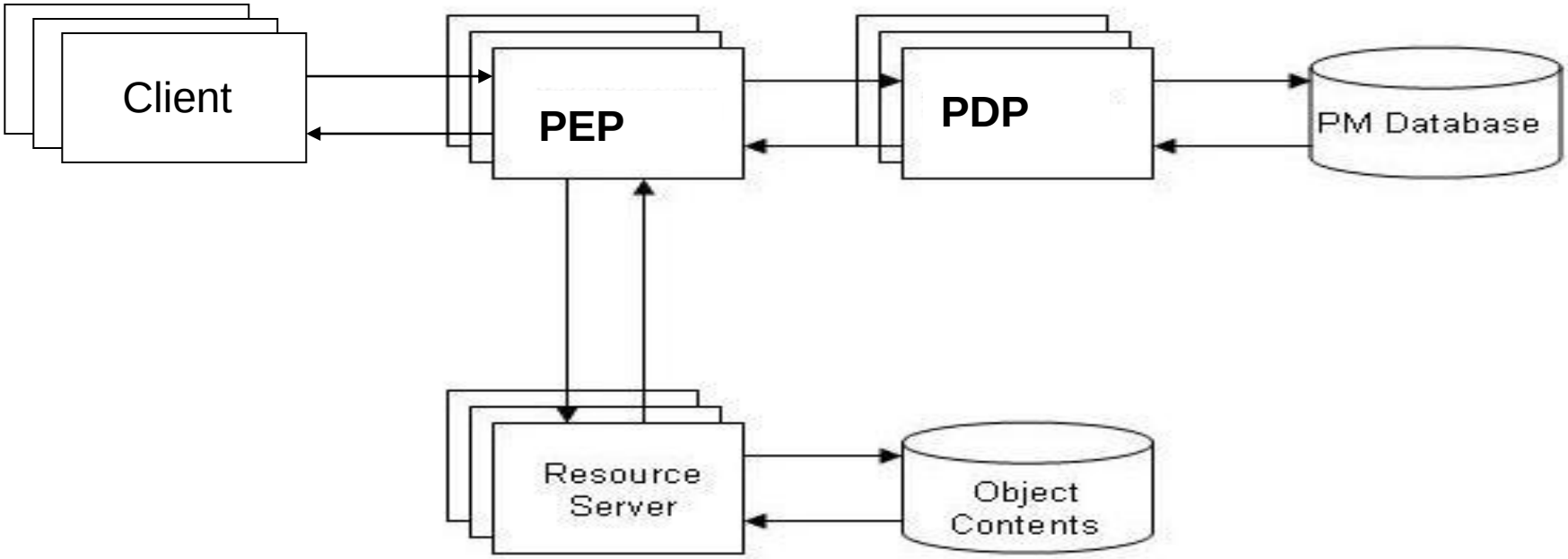
Obligations (Event-Response)

- Format: **when** *event-pattern* **do** *response*
- Event: successful execution of an operation (e.g., reading of an object's content, or creation of a user).
- Event pattern: the context in which an event occurs (operation, object, user, containers, etc.)
- Response: sequence of administrative operations that may dynamically change the configuration of PM relations.
- Example: **when** process reads object from "Top Secret" **do** create p-deny(process, {write}, not "Top Secret").

Policies

- Tailored policies through configuration of relations alone. E.g., Combinations of mission specific forms of:
  - DAC
  - RBAC
  - History and object-based Separation of Duty
  - Workflow
  - Forms of confinement (e.g., only doctors can read medical records, MLS, I know who has access to my data and I can revoke access, only users in *abc* group can read message *xyz*)
  - Chinese wall (conflict of interest)
- Library of configurations and commands exist for immediate instantiation

General Architecture



Conclusion

Native PM advantages

- The PM is a unification framework where data services and access controls operate as one.
- A user authenticates once but may exercise legitimate capabilities over a multitude of data services
- Data services (to include AC as a data service) naturally interplay
- Code normally included in apps for controlling access, sharing and distributing data is displaced by PM configuration– Eliminating some vulnerabilities and making Apps development easier
- Data can’t leak into local environment or to outside world
- Combinations of mandatory, discretionary, and history-based access controls are comprehensively enforced across applications
- Access control policies can be tailored to mission of subscriber
- Library of pre-existing configurations