UNIVERSITÀ DI PISA

CONSIGLIO NAZIONALE DELLE RICERCHE
ISTITUTO DI INFORMATICA E TELEMATICA

USAGE CONTROL

Athanasios Rizos

Pisa, 2017

Presentation Outline

- Introduction
- Access Control
- Usage Control (UCON)
- UCON Policy Components
- UCON Service Framework (UCS)
- UCON Implementation Examples
- UCON Conclusion
- Future Work
- Conclusion

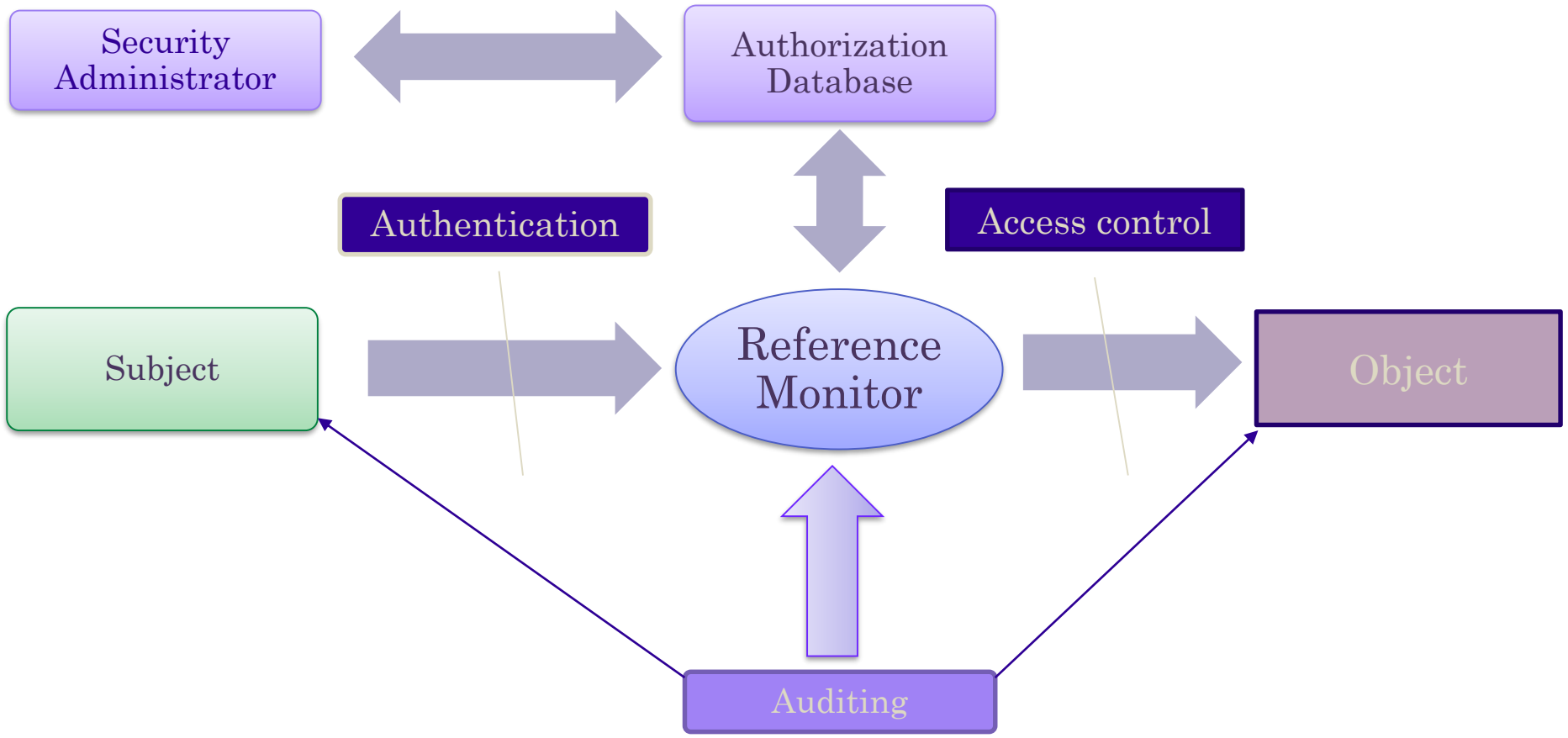
Introduction (1)

- Age: 29
- Hometown: Ioannina, NW Greece
- Degree: Electrical and Computer Engineering
- Specialization: Computer Science and Engineering
- Passion: Automotive Racing
- Hobbies:
  - Music (Playing wind instruments)
  - Trekking

Introduction (2)

- ESR in: Consiglio Nazionale delle Ricerche (CNR)
- CNR location: Pisa, Italy
- Institute : Istituto di Informatica e Telematica (IIT)
- Research Group: Trustworthy and Secure Future Internet
- Supervisor: Dr. Fabio Martinelli

Access Control



Policy

- Set of rules that determine whether access should be allowed or denied
- Rules are depended on attributes
- Access control evaluates attributes only once, before the start of a session
- What if they change during this session?

Usage Control - UCON

- Guarantee that subjects authorized once, remain while a session is in progress
- Security policy based on attributes defines when a subject should be authorized
- Mutable and immutable attributes
- Attributes might change in time (mutable)
- Three levels of abstraction
  - Policy, Enforcement, Implementation

Why UCON

- Continuity of control
  - Decision should not only be evaluated before granting access to subject but also while the subject uses the resources
- Mutability of attributes
  - Attribute changes might cause policy reevaluation which might lead to revocation
- Attributes change while access in progress and if security policy is not satisfied, reference monitor revokes access and terminates the subject's usage of resources

UCON Mechanisms

- UCON uses mechanisms to enforce security policies
  - They intercept each request to a resource, determining if trusted according to security policies and enforcing the access decision
- During access mechanisms work continuously
  - If subject remains trusted access is continued
  - If not, access is revoked and resources are released (session terminated)
- Sufficient collection of mechanisms is call reference monitor

Access Control versus UCON - Example

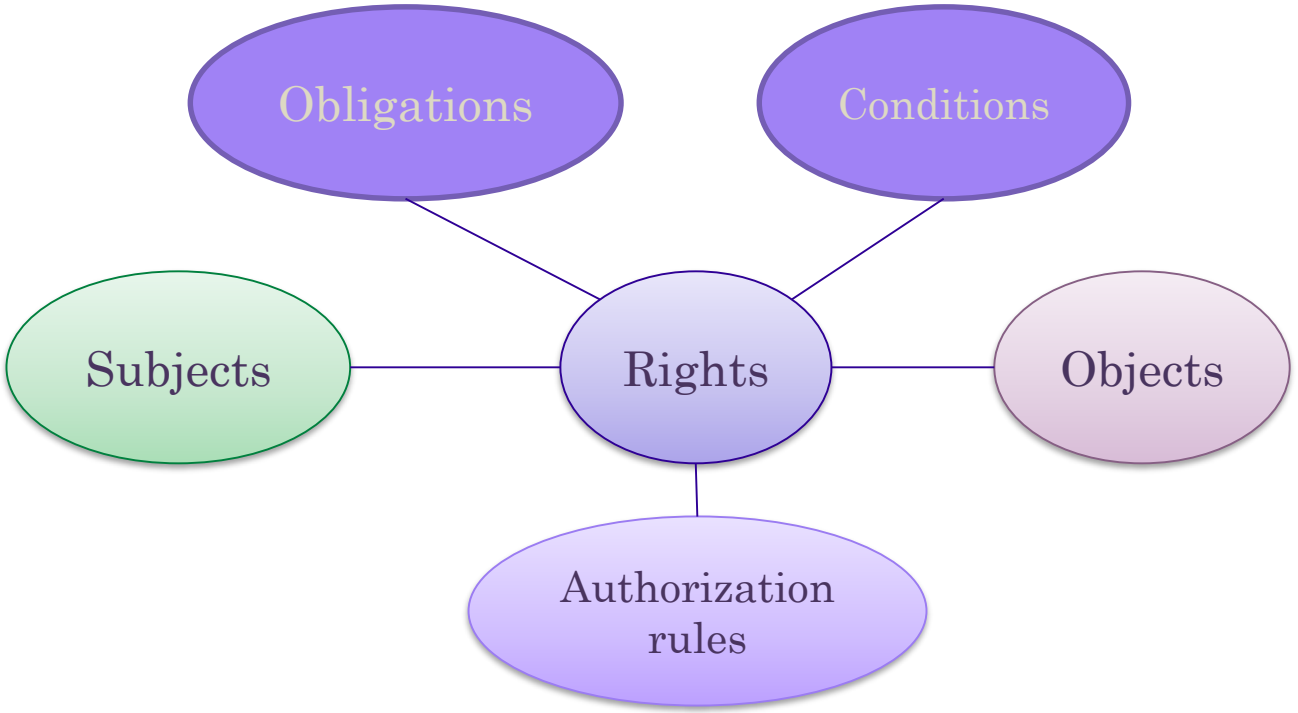
Access Control

- Video streaming access is granted if balance is sufficient
- Subject is granted to use the resources until the end
- Session ends early only by subject's request

UCON

- Video streaming access is granted if balance is sufficient
- If Subject's balance is no more sufficient after sometime, access is revoked and resources are released
- Session may end earlier if balance is no more sufficient

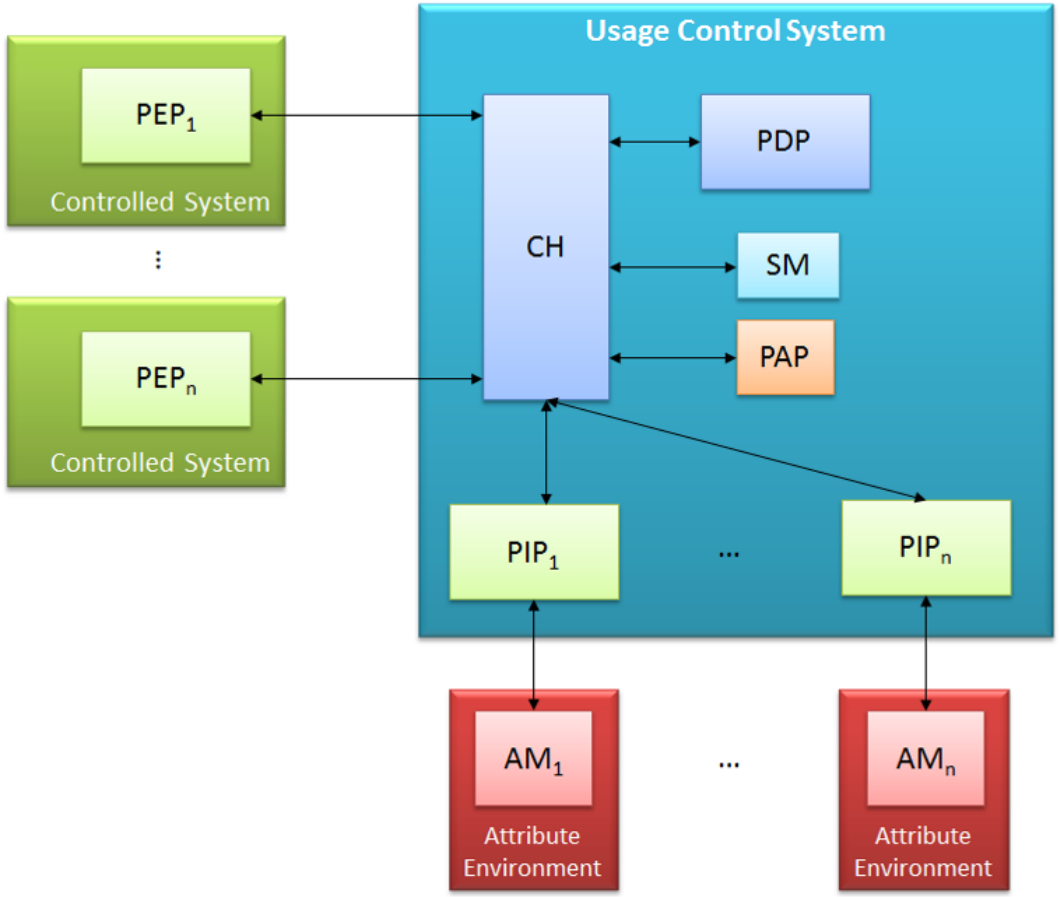
UCON Policy Components



UCON Components Example

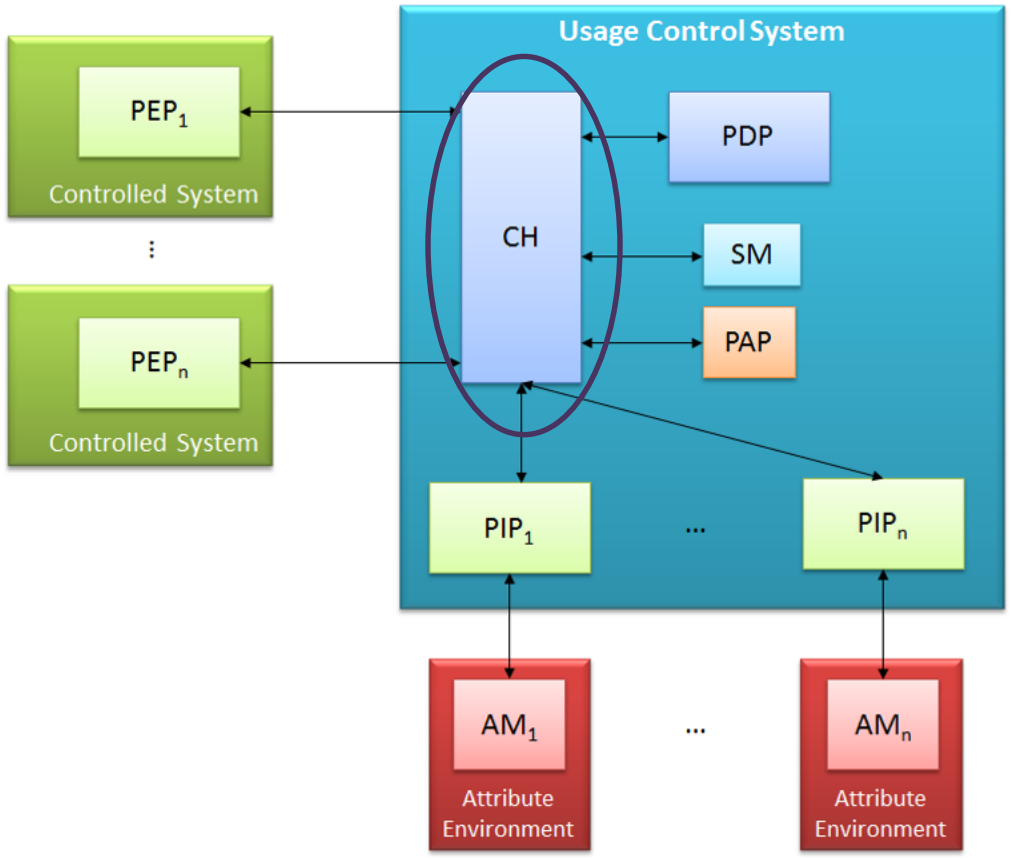
- Subject: user who wants to see the video
- Object: the video stream
- Right: watching of the video stream
- Authorization rule: having enough balance in the account for all the video duration
- Conditions: details of connection between subject and object (ex. bandwidth, resolution etc.)
- Obligation: a set of conditions that must be enforced when a related policy rule is activated

UCON System Framework (UCS) Architecture



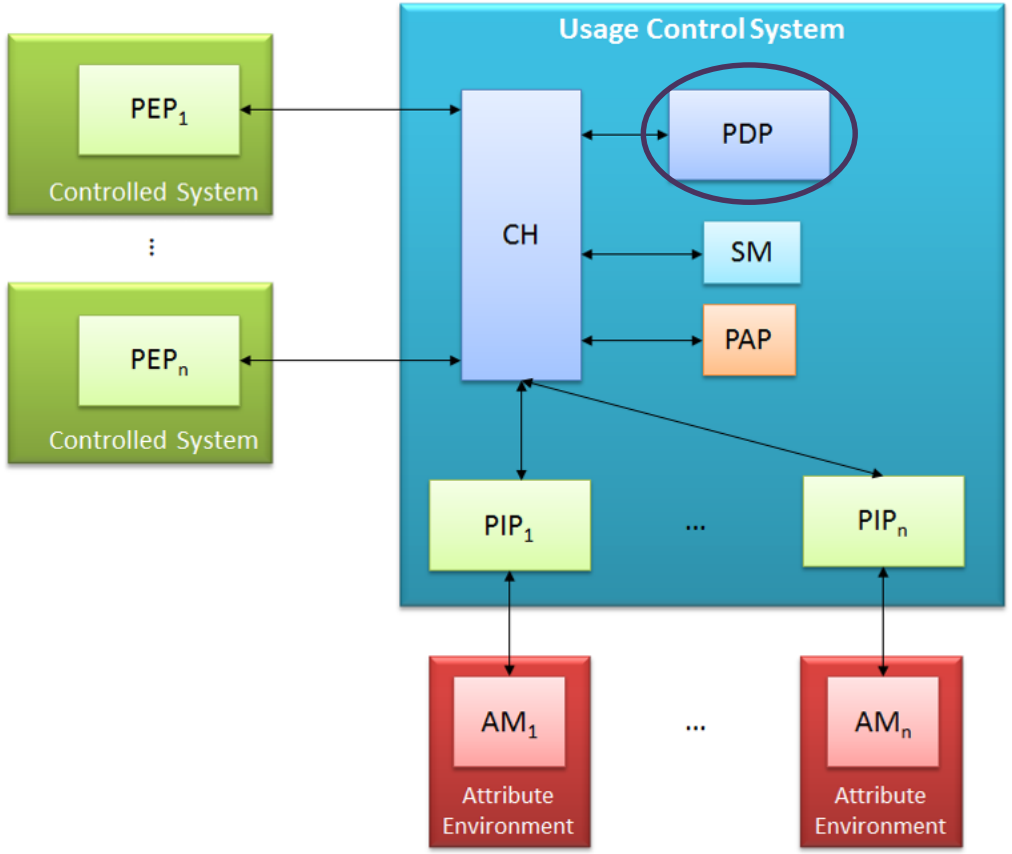
Context Handler (CH)

- Front – end of UCON service
- Handles communication with all UCS components
- Manages all functionalities of the framework



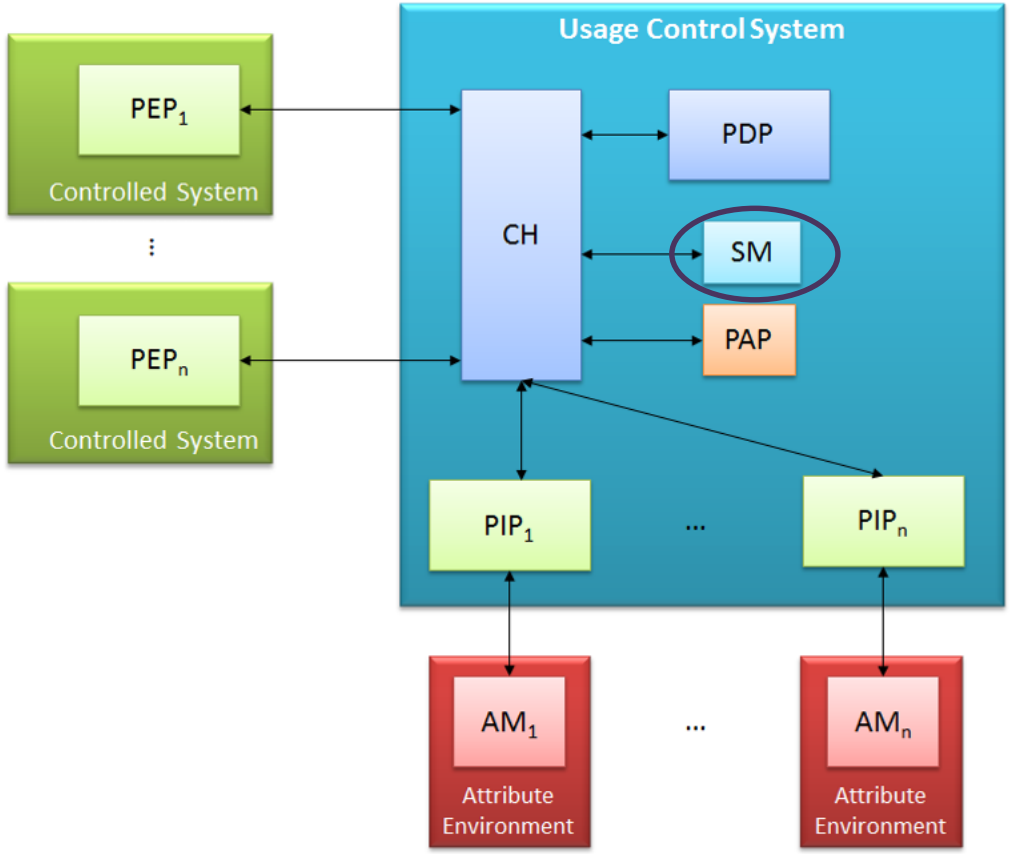
Policy Decision Point (PDP)

- Policy evaluation engine
- Takes access request and policy as input
- Returns the decision (Permit, Deny, Undetermined)



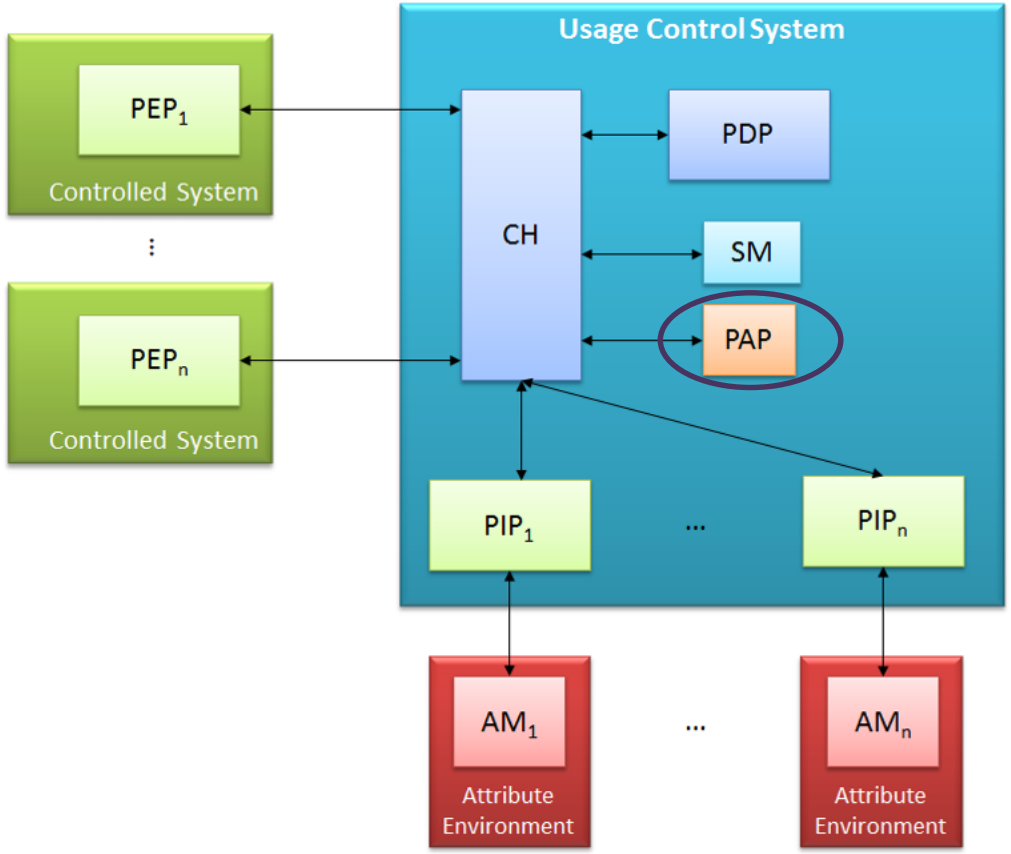
Session Manager (SM)

- Database for storing data about sessions
- In charge of keeping track of sessions
- Storage of status of active sessions
- Stores and provides to PDP useful information needed for reevaluation
- Provides further operation and query (ex. get the list of the active sessions)



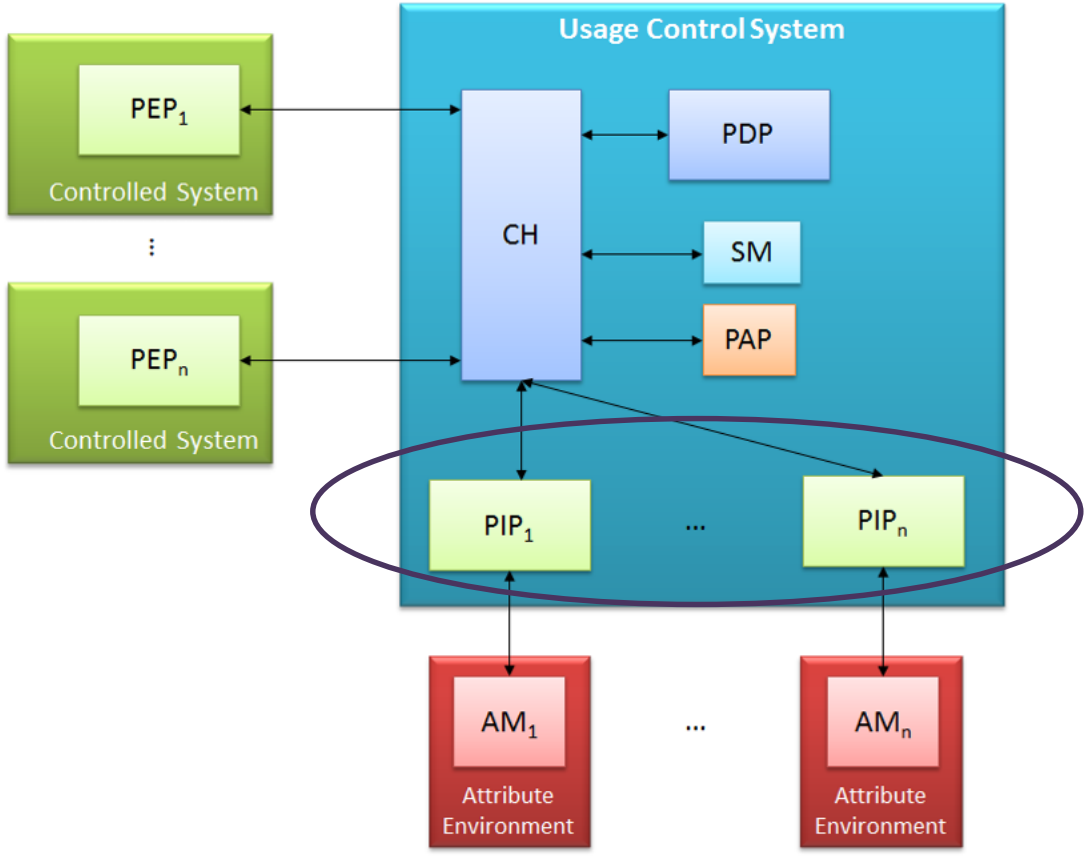
Policy Administration Point (PAP)

- Stores the policies for the PDP.
- Not mandatory since policy might be included in the access request coming from PEP.



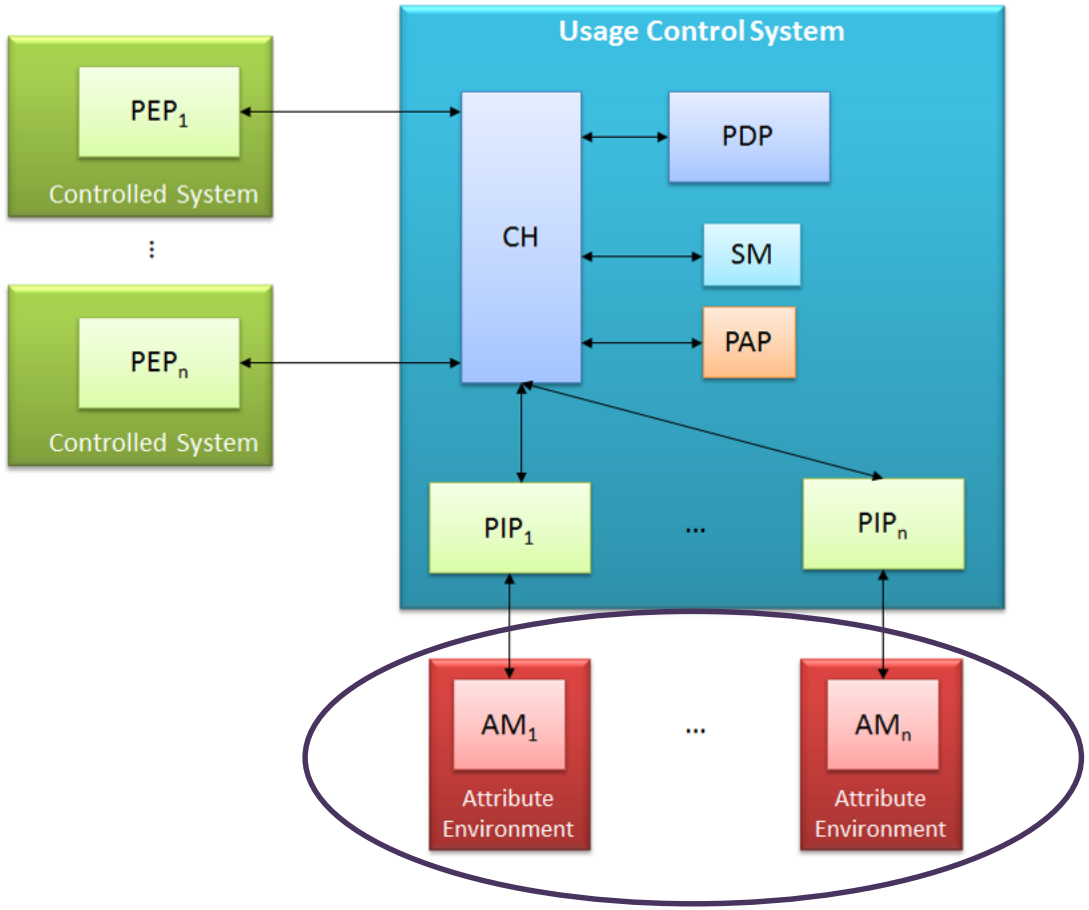
Policy Information Points (PIPs)

- Interface with AMs in order to:
  - Retrieve, (un)subscribe, update on attributes
- Same interface to CH but should be adjusted to communicate with each AM
- PIP acts as an architecture plugin to let UCON service as flexible as possible in interacting with different AMs
- Each PIP provides same interface to CH



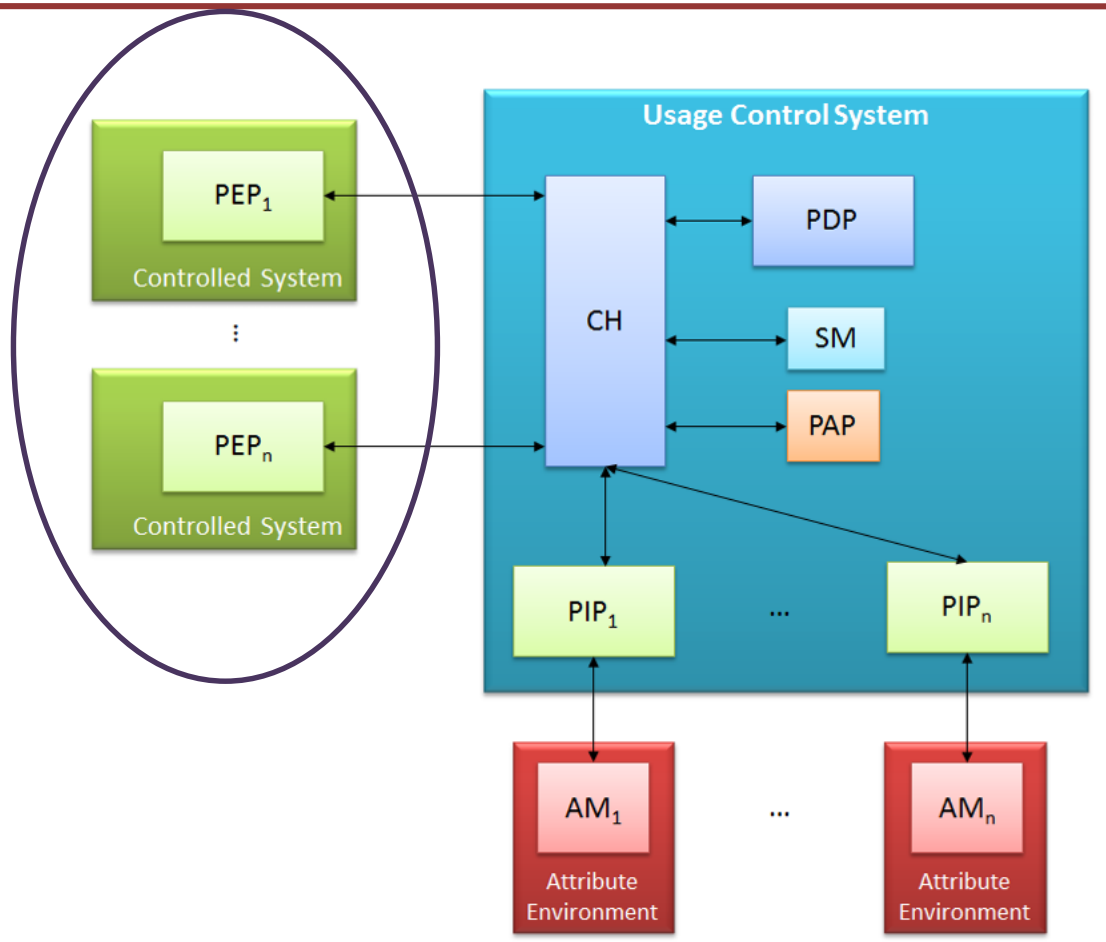
Attribute Managers (AMs)

- Components that manage attributes of objects, resources, environment and actions
- Part of the attribute environment
- Not part of UCON framework



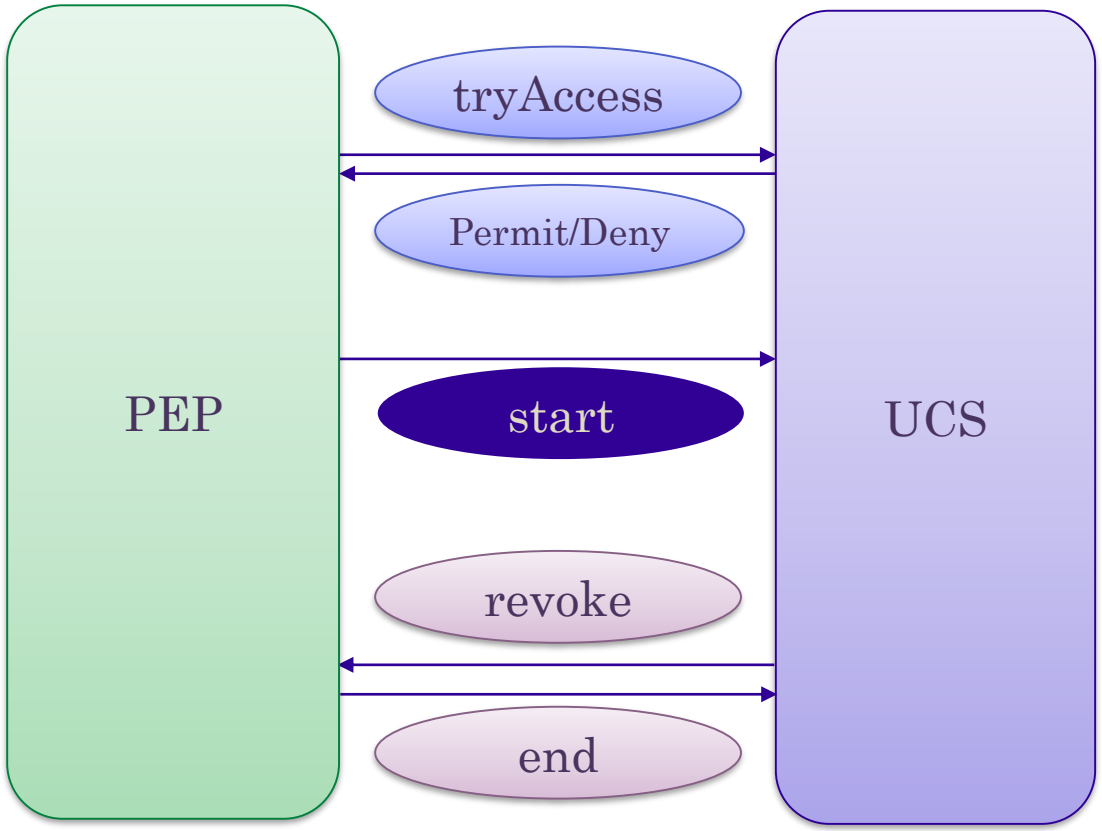
Policy Enforcement Points (PEPs)

- Part of framework to interact with subjects
- Can also manage, resources, environment and actions
- Placed in controlled systems
- Capability of interception or stopping of actions
- Describe Subject's attributes to the Policy Decision Point (PDP) / might also be taken from PIPs
- Receive and enforce a security decision



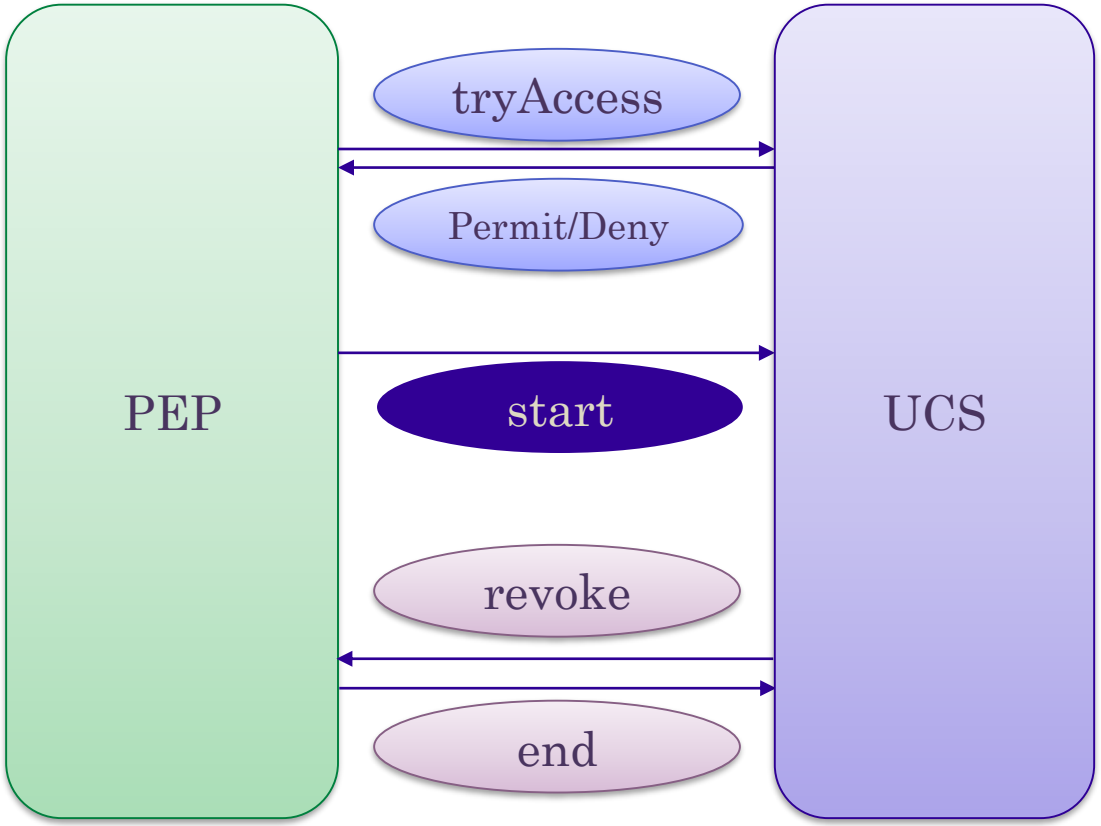
UCON Workflow

- Workflow components
  - TryAccess (login action requesting access)
  - PermitAccess/DenyAccess decision
  - StartAccess (start using resources)
  - RevokeAccess (action termination by violation of rules)
  - EndAccess (action termination by user)



UCON Workflow - Example

- Subject (via PEP) requests usage of the video streaming service (TryAccess)
- Pre-update of attribute “balance” (CH to PIP)
- CH asks PDP and receives decision of access
- CH answers to Subject (PEP) (PermitAccess)/(DenyAccess)
- If (PermitAccess) Subject (PEP) sends (StartAccess) and starts video watching.
- On-update of “balance” from PIP.
  - If balance >0 session continued
  - If balance <=0 access is no longer granted (RevokeAccess) (UCS to PIP and PEP)
  - If Subject wants to finish (EndAccess) from PEP to UCS
- Post-update of attributes that action is terminated
- CH informs SM for RevokeAccess/EndAccess



Internet of Things (IoT)

- Machine to Machine (M2M communication)
  - REST/HTTP protocols previously used
  - TCP flow control not appropriate for resource constrained devices
- Most popular protocols used nowadays
  - Constrained application protocol - CoAP (created as an extension with optimized HTTP functions)
  - Message Queuing Telemetry Transport - MQTT (uses hierarchical copy of Publish/Subscribe mechanism standardized by OASIS in 2013)
  - IPsec for 6LoWPAN
  - EventGuard (also uses Publish/Subscribe mechanism)
  - QUIP

MQTT protocol

- Introduced by IBM – Standardized by OASIS (2013)
- Uses Publish/Subscribe mechanism
- Publishers/Subscribers controlled by Broker
- Broker: software component which is the overlay infrastructure
- Broker is responsible for distributing messages to interested clients
- Enables “pushing” data from the cloud rather than polling by the device for the data from the server
- Most popular protocol for IoT

UCON in MQTT

- Problems
  - MQTT has only standard access control and authentication mechanisms
  - Information is no longer controlled if Broker distributes it
  - Once Subscriber is authenticated there is no longer check on his authentication
- UCON contribution
  - Provide selective access after message delivery
  - Continuous control of Publishers/Subscribers on both authentication and access

Future Work - UCON

- Work with and improve UCON framework prototype created by IIT – CNR team
- Create applications that use this framework
- Extend framework's functionalities
- Create a working prototype of Distributed UCON
- Assign weights on each attribute (not each attribute has the same significance)

Future Work - IoT

- Use UCON for enhancing security of MQTT protocol
- Make UCON work with UCON smoothly even in Distributed Systems
- Study UCON capability to cooperate with several other Protocols
- Study the possibility to use a different UCS for each Publisher/Subscriber and make Broker act as a central UCS in order to minimize bandwidth allocation

Conclusion

- UCON is a way of continuous monitoring and reevaluating of attributes in order to control access in every step of a service
- Solves problems especially in largescale systems when long lasting access is needed
- UCS framework must be as generic as possible for easier implementation
- Due to UCS easy implementation feature it is very easy to integrate with IoT protocols (e.g. MQTT) in order to enhance their security capabilities