Towards a Usage Control based Video Surveillance Framework

Gianpiero Costantino

Consiglio Nazionale delle Ricerche

Co-authors:

Enrico Carniani

Francesco Marino

Fabio Martinelli

Paolo Mori

PDP Conference 19/02/2016

Outline

- **Usage Control Model**
  - Introduction
  - Our Goal
- **Our Architecture**
  - Componentes details
  - Implementations
- **Conclusion**

Usage Control (UCON) Model

- Usage Control models checks whether a **subject** has the right to perform a specific **action** on a **resource.**
  - *Peter can access room-25*;

- We focus on UCONs with mutable attributes;
  - *Number of people in a room;*

- Security Policies are evaluated before (**pre-evaluation**) and during the usage of the object (**ongoing-evaluation**)

Goal

- Using **Usage Control** and **Video Surveillance** to enforce security policy for a monitored room

Example of Policies

I) An employee can stay in the room.

II) A guest can stay in the room only if at least one employee is in the room too.

Face Recognition

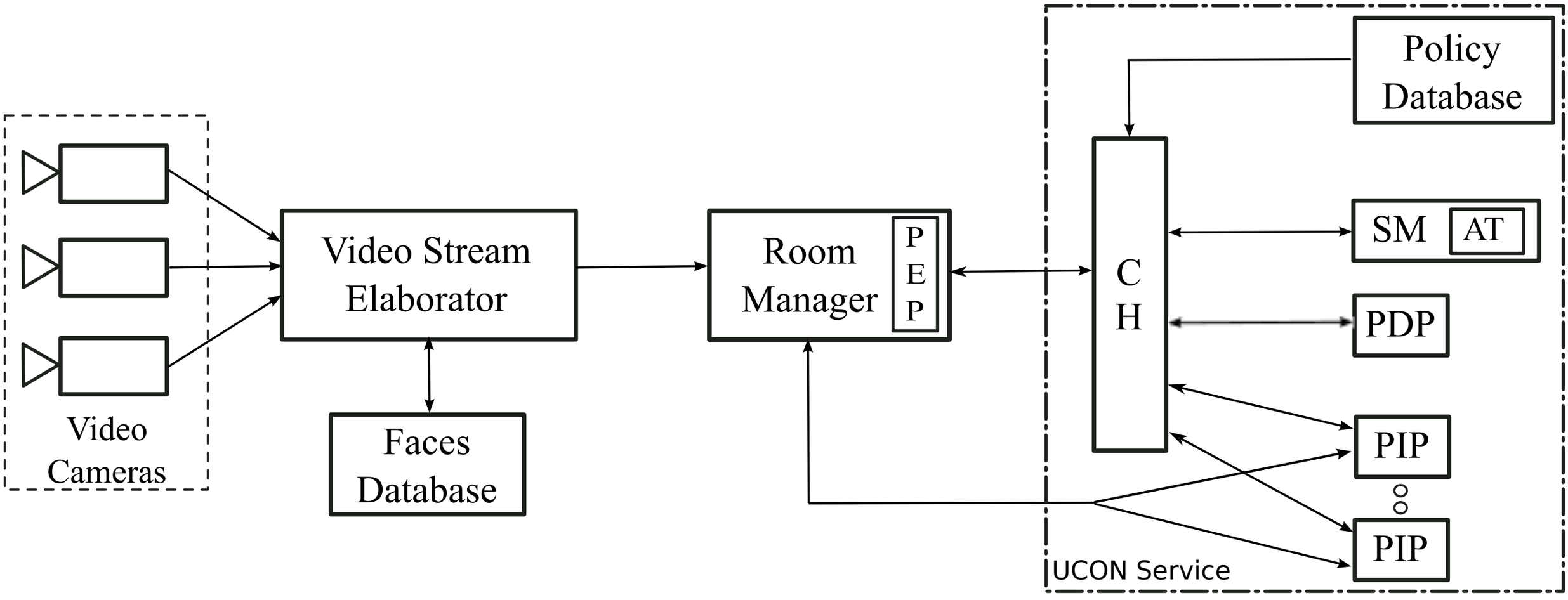
- We use Face Recognition to identify people inside the room;

- We use the *Scale-Invariant Feature Transform* (SIFT) as recognition algorithm;

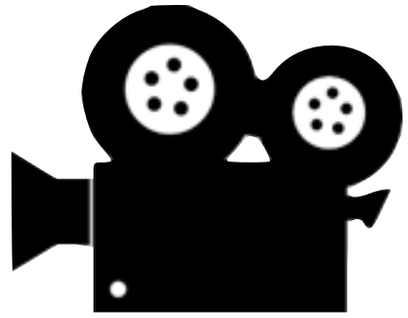
- SIFT works well with good light conditions and it is robust to variation of scale and orientation;

- Detected Faces are compared with those one in the *Faces Database*;

Architecture of the Framework



Video Stream Elaborator



- It is the component that processes the Video Stream taken from the *Video Cameras*.

- Detected faces are compared with those ones already present in the Faces Database;

- Faces Database provides also the **Role** attribute of the recognised person;

- Policy is evaluated as pre-evaluation:
  - Permit -> Video-Buffer deleted.
  - Deny -> Alarm triggered, Video-Buffer recorded

Room Manager

- It interfaces the *Video Stream Elaborator* and the *UCON*;
- The Room Manager includes **Policy Enforcement Points** (PEPs)
- PEPs interacts with the **Context Handler** (CH) sending and receiving status messages:
  - *tryaccess;*
  - *permitaccess* or *denyaccess;*
  - *endaccess;*
  - *revokeaccess;*

Room Manager

- A recognised user triggers a *tryaccess*, which is evaluated with a pre-evaluation authorization;

- When the CH receives a *tryaccess*, it forwards the message to the **Policy Information Points** (PIPs);

- PIPs retrieves others attributes needed to evaluate the policy, such as the Room Number;

- New attributes are sent to the CH. It calls the **Policy Decision Point** (PDP)

UCON - Policy Decision Point

- PDP evaluates the request against the **pre-policy** and returns the access decision to the PEP;

- The **ongoing-evaluation** starts in case of *positive* response;

- PIPs observe the **status** of the attributes;
  - If an attribute changes, the PIP informs the CH;

UCON - PDP - *ongoing*

- The PDP evaluates the **ongoing-authorization** and decides if *permitaccess* or *denyaccess*;

- In case of *denyaccess*, the CH sends a *revokeaccess* to the PEP, which starts recording the video, and eventually triggers an alarm.

UCON - Session Manager

- The Session Manager (SM) manages all the **sessions** in the UCON;
- A session starts when a new *tryaccess* is created;
- Sessions are store into **Access Table** (AT);
- The AT contains info like:
  - *SessionID*;
  - *SessionStatus*, i.e., active, ended, revoked;

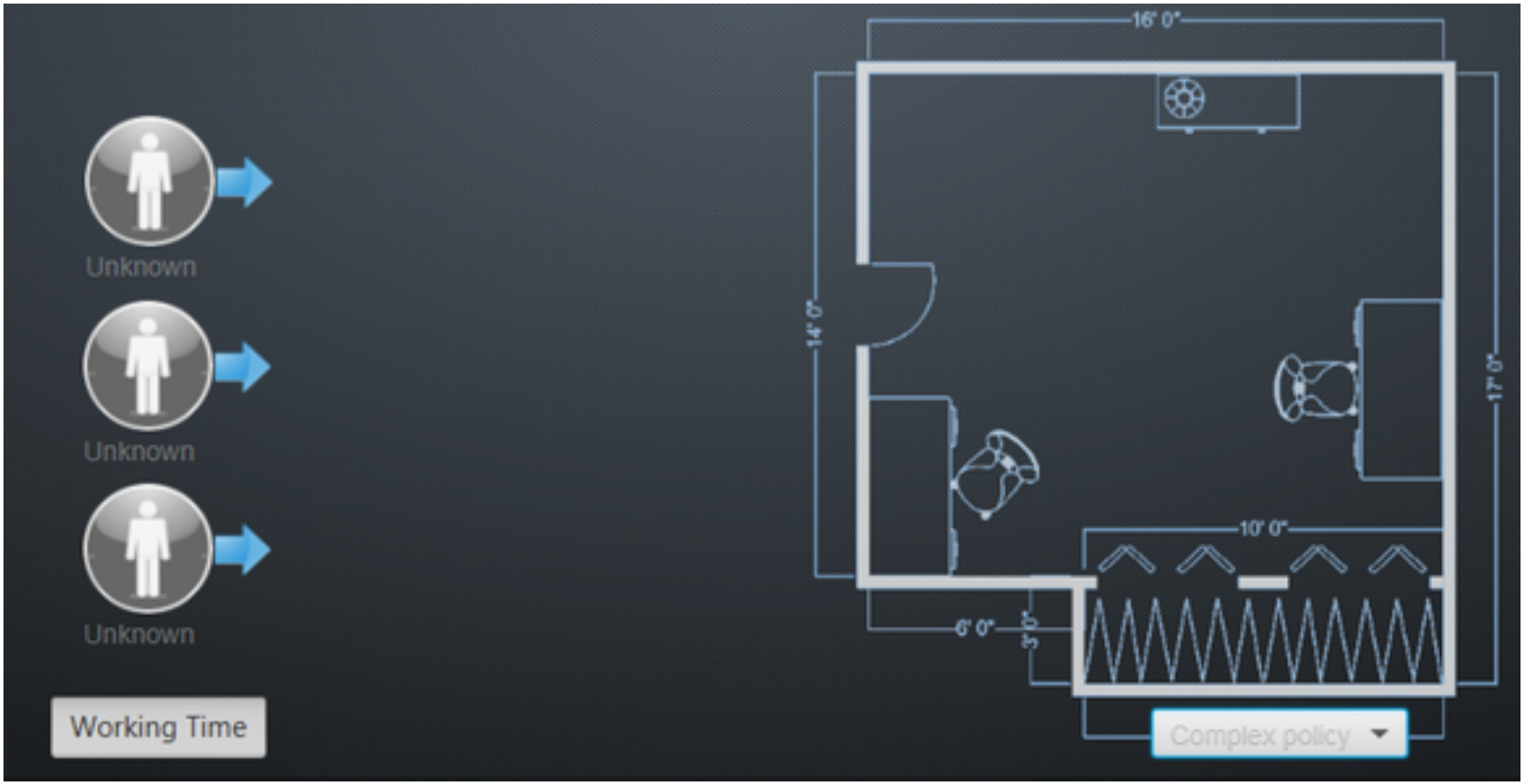
Architecture Development

- We implemented our architecture using a D-Link DCS-942L camera;
  - **20fps** with *640x480*resolution

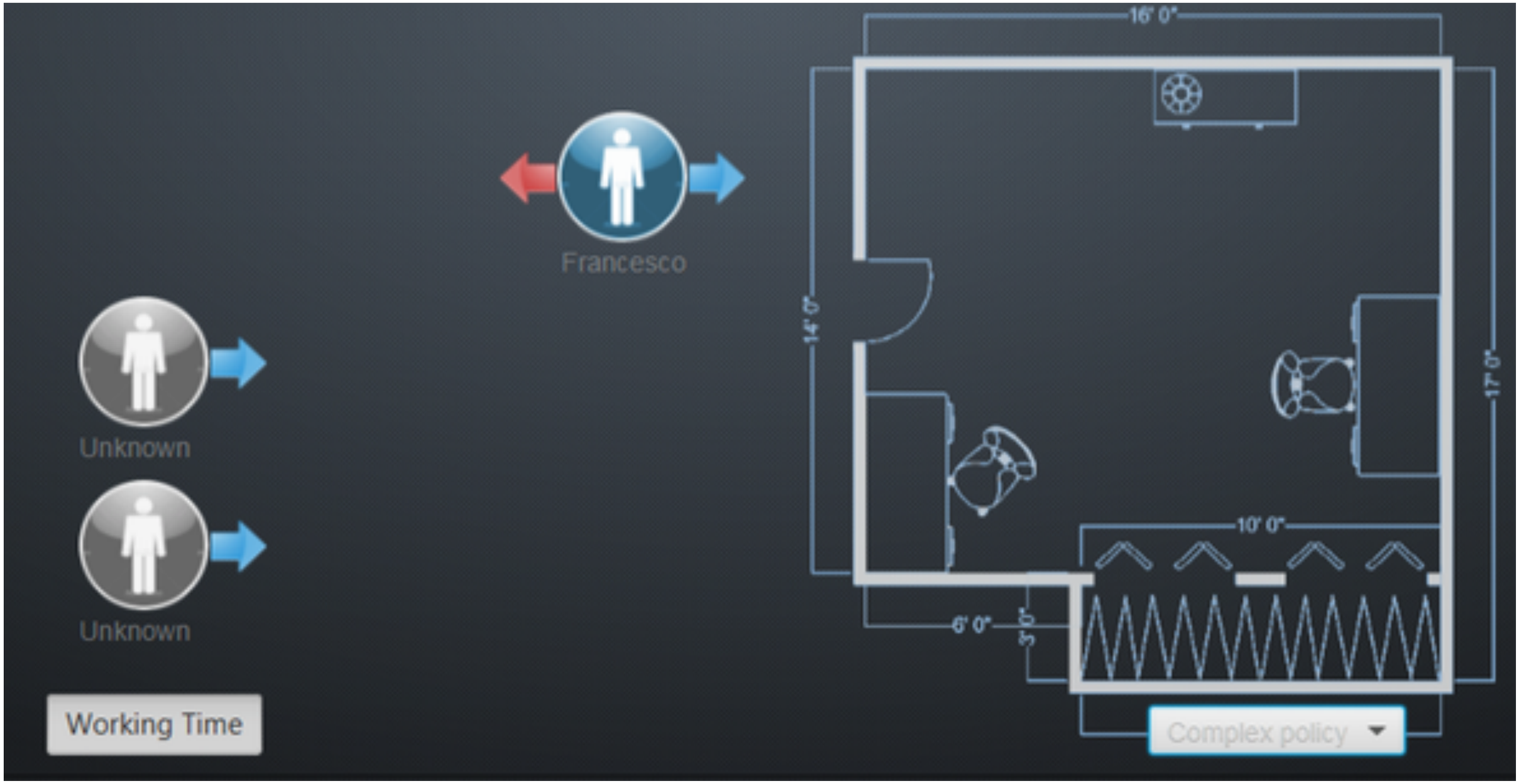
- The Video Stream Elaborator is a *JAVA* application that uses JavaCV to access computer vision functions;

- *FaceDatabase* is implemented using MySQL
  - *200x200*pixels per Faces, along users attributes.

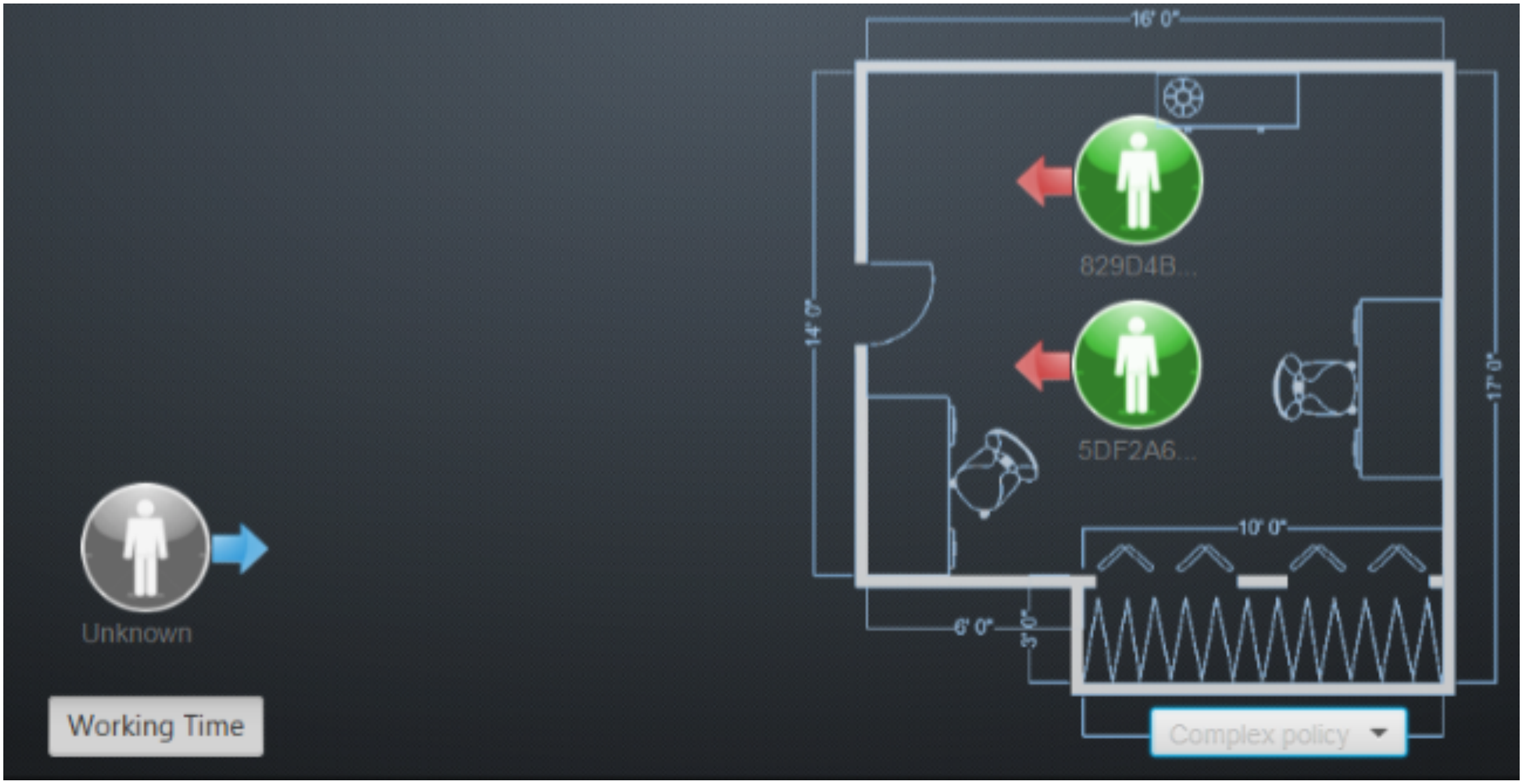
Simulation


Unknown
Unknown
Unknown
Working Time
16' 0"
14' 0"
17' 0"
10' 0"
6' 0"
3' 0"
Complex policy

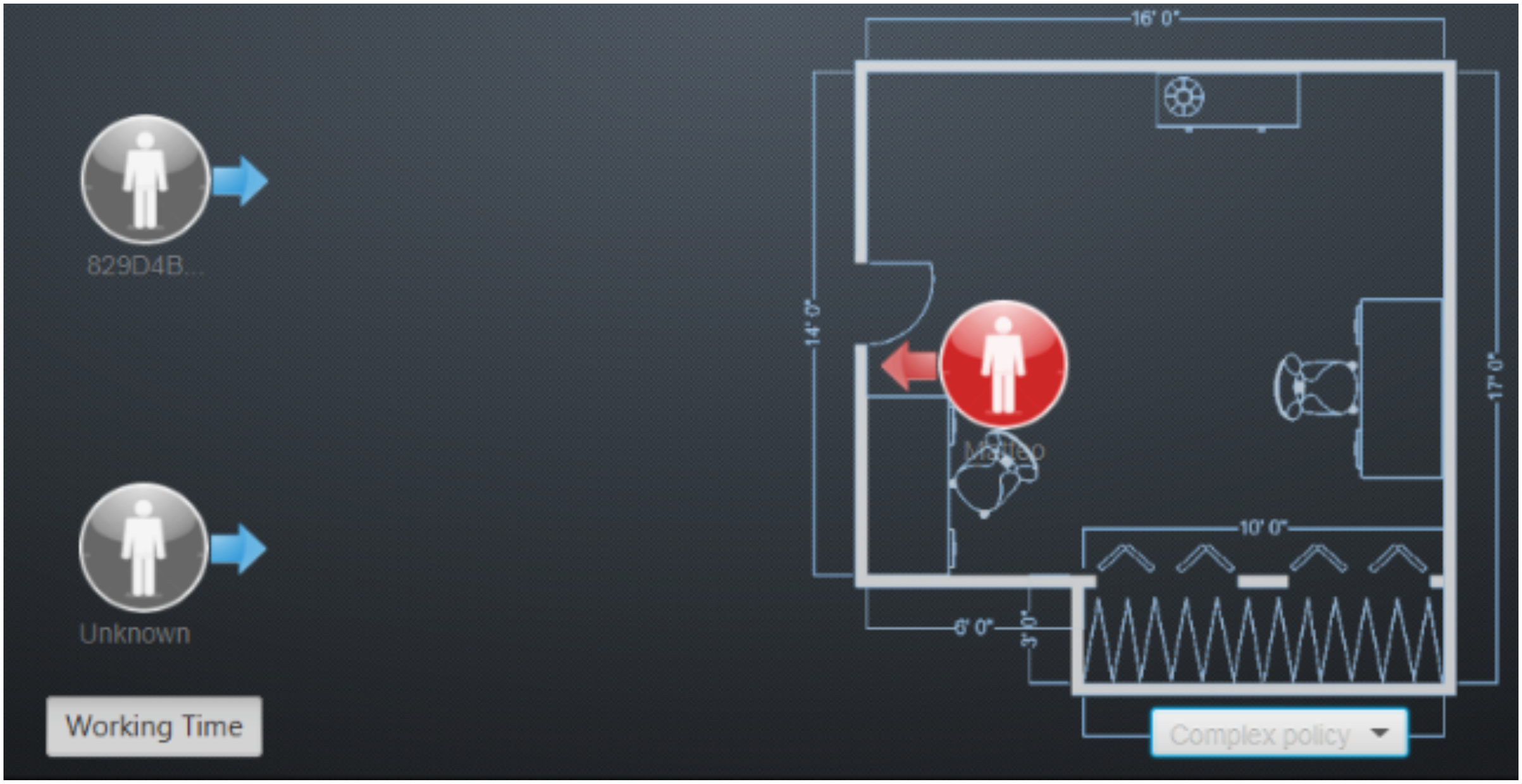
Simulation - *pre*



Simulation - *ongoing*



Simulation - *revokeAccess*



Conclusion

- We have combined **UCON** and **Video Surveillance** for room monitoring;

- The Architecture we have presented is the result of this combination;

- Our architecture propose also a way to protect the *privacy* of users in the monitored room;