GRAPHICAL MODEL-BASED PRIVACY POLICY EDITING FOR SMART VIDEO SURVEILLANCE

Pascal Birnstill[1], Christian Burkert[2] and Jürgen Beyerer[1]

[1] *{pascal.birnstill, juergen.beyerer}@iosb.fraunhofer.de*
Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB
Fraunhoferstr. 1, 76131 Karlsruhe (Germany)

[2] *burkert@praedmandatum.de*
praemandatum GmbH
Goseriede 4, 30159 Hannover (Germany)

Abstract

Poor usability and, as a consequence, human errors can render powerful security and privacy mechanisms useless. Borrowing from the concept of visual programming we introduce a graphical editor for authoring privacy policies for smart video surveillance systems, i.e., systems involving computer vision technology to some extent. We built this editor upon a meta model, which we derived from a generic architecture for smart video surveillance. Employing our tool, privacy-related requirements can be assembled from readily understandable graphical blocks and exported into machine-readable usage control policies. We give a demonstration based on a smart video surveillance system employed for fall detection in medical facilities.

Keywords: Smart video surveillance, privacy, usage control, policy authoring, usability

1 INTRODUCTION

Privacy-related requirements in smart video surveillance range from *which data the system extracts and analyzes* over *which data it visualizes for situation assessment* (e.g., "only anonymized video data must be exposed") to *when data must be deleted*. We typically specify such requirements in policies using machine-readable markup languages. However, editing such policies using text editors is effortful, error-prone, and requires the knowledge of technical details.

We propose a graphical policy editor for smart video surveillance based on visual programming. Assembling privacy policies from combinable graphical blocks relieves organizations that operate video surveillance systems and particularly involved system architects and administrators from knowing syntax rules and abstracts from the technical implementation. We introduce a meta-model, which provides our editor with semantics of a generalized smart video surveillance system equipped with a usage control enforcement infrastructure. Building on this meta model, we are able to graphically represent the operation of a smart video surveillance deployment, integrate privacy-related requirements in terms of restrictions, mechanisms and obligations, and export XML-based policy syntax. We instantiate our editor for a scenario, in which smart video surveillance is deployed for fall detection in a medical facility in order to illustrated how it is employed.

This approach is application-oriented in a sense that we do not aim to entirely cover the complexity of the policy specification language. In this sense, we do not provide a usage control policy editor, but a tool for modelling common privacy requirements, which we translate into usage control policies.

Our contributions are (i) a convenient and efficient approach for specifying privacy policies for smart video surveillance measures, (ii) an easily understandable

representation of the behavior of a smart video surveillance system, and (iii) machine-readable policies understood by a usage control infrastructure.

1.1 Distributed Usage Control

Usage control [3] generalizes access control to the time after the initial access to data. Requirements include rights and duties, e.g., “data must not be forwarded”, “data must be deleted after 30 days”, “usage of data must be logged”. Usage control requirements are specified in policies referring to intercepted events in the workflow of a system. In distributed setting, i.e., forwarding data with an attached policy to another system, usage control requirements can be enforced on the receiver’s machine, too, requiring appropriate usage control enforcement mechanisms at the receiving end [6].

1.2 Usage Control Enabled Video Surveillance Architecture

Using our policies, we govern a smart video surveillance system design, which provides at least three operational modes. Its *default mode* is optimized for privacy: It collects and reveals a minimal amount of data. Event-specific *assessment modes* create views of the scene and available meta-data, such that human operators can distinguish critical incidents from false alarms. At this stage, we still protect observed people's privacy as far as possible, typically by applying anonymization techniques before exposing video data. Finally, *investigation modes* unlock additional functionality for handling a specific type of incident and may involve deeper privacy intrusions. However, usages are logged and coupled to critical incidents detected by the system. By this means, usage of such functionality in a groundless and unjustified manner can be revealed in hindsight.

2 SMART VIDEO SURVEILLANCE META MODEL

In the following paragraphs we introduce a smart video surveillance meta model, which provides entities for modelling components, data structures, data attributes, and mechanisms commonly found in such systems. It enables designers of concrete surveillance systems to model their individual deployment with particular focus on privacy-related requirements.

2.1 Sensors

A sensor is typically attached to a wall or a ceiling and does not change its position. Its capacity of capturing environmental signals has a limited scope. Both, its location and scope can be described as a set of spatial divisions of the monitored area as depicted in Fig. 1.

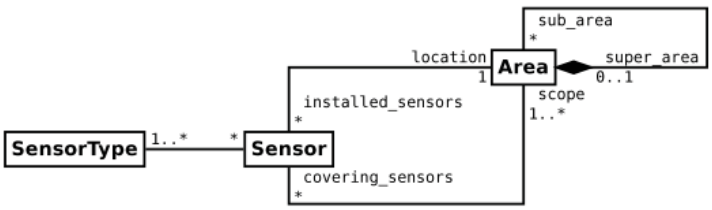

Fig. 1: Sensors have an installation location and capture signals within a spatial range

2.2 Information Persistence

In some deployments raw or processed information is stored permanently, e.g., for retrospective investigation or preservation of evidence. Smart surveillance systems usually maintain a semantic information base containing an abstraction of the observed environment's present state (*world model*) and possibly a history of state changes.

Another type of storages is archives that normally contain raw sensor data, possibly augmented with semantic tags.

A record represents a monitored object, which can be a person, but also a thing. In general, storage records comprise attributes like unique identifiers, privileges, position coordinates or face templates. Fig. 2 shows a generic storage sub-system.

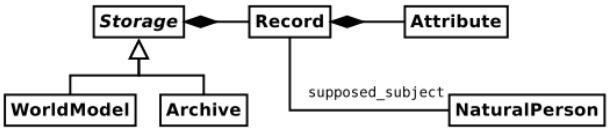

Fig. 2: State representation as records of a semantic world model and an archive

2.3 Information Embedded in Raw Sensor Data

Raw sensor data may contain information, which the surveillance system cannot extract or is not configured to extract. Unless explicitly given, the surveillance system is unaware of the possible existence of such embedded information. However, this kind of information is required, e.g., for describing privacy filters that aim at obfuscating certain features of person's visual appearance before releasing the data to a human operator.

Figure 3: Modelling which human attributes may be embedded in raw sensor data allows us to deploy according privacy mechanisms

Therefore our meta model covers human attributes, which are implicitly captured by sensors, described in record attributes, or affected by modifiers (cf. Figure 3). These attributes can be grouped into categories like visual appearance or movement behavior or, for instance, according to their privacy sensitivity as defined by company regulations or legislation (cf. § 3 (9) German Federal Data Protection Act).

2.4 Information Representation: Viewers

Human user interfaces provide operators with a suitable selection of information for assessing a recognized activity or situation. In our meta model, *viewers* generate a particular visualization of given data. We distinguish two types of viewers (cf. Fig. 4). *Raw viewers* generate a representation of raw sensor data or at least include such a one, whereas *rendering viewers* construct entirely synthetic views. This distinction accounts for the particular risk that raw sensor data may contain privacy sensitive data, which the system is not aware of when releasing data (cf. 2.3).

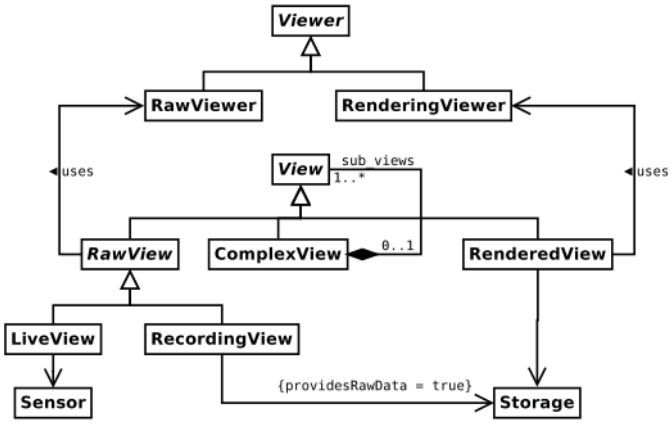

Fig. 4: Views represent a subset of their source's information, selected and composed by the used viewer

Views are concrete applications of viewers for a specific source of data. As depicted in Fig. 4, four different types of views are modelled: *live views* apply a raw viewer to a live stream of sensor data, *recording views* apply a raw viewer to recorded raw sensor data, *rendered views* create a synthetic visualization of stored meta data using a rendering viewer, and *complex views* are compositions of multiple other views.

2.5 Modification of Raw Sensor Data

Concerning raw views, privacy regulations may require an obfuscation of sensitive attributes, whereas in other applications augmenting overlays may be useful to quickly assess a critical situation. Our meta model therefore distinguishes between *data reductions* that reduce (anonymization/obfuscation techniques) and *data exploitations* that augment the information value of presented raw data. As depicted in Fig. 5, a raw view (live and recording views) can employ data modifiers to alter the raw data. A data reduction may be reversible (e.g., encryption).

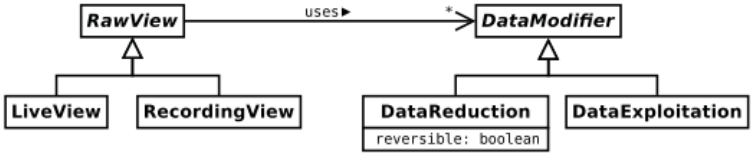

Fig. 5: Data reductions and exploitations can be associated to a raw view

2.6 Policy Structure

Our intuition of a *policy* is derived from usage control policies that can be specified as an *event-condition-action (ECA)* rule. A policy (cf. Fig. 6) is triggered by an *event*. The policy's *actions* are executed if and only if the specified *condition* concerning the event is evaluated positively by the *decider*, a so-called Policy Decision Point (PDP) in terms of usage control. An event either originates from a system internal process (e.g., the expiration of a storage permission) or from external actions (e.g., a detection of a computer vision algorithm, an operator interaction). Until now, it was not required to distinguish event origins in the meta model.

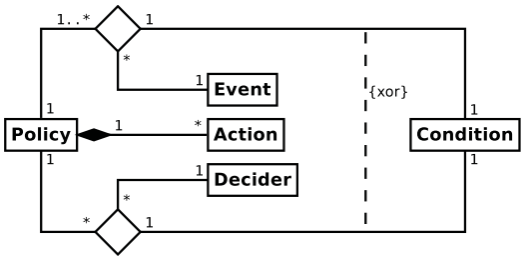

Fig. 6: Abstracted structure of (usage control) policies

3 INSTANTIATION: VIDEO-BASED FALL DETECTION IN HOSPITALS AND NURSING FACITLITIES

Due to space limitations we omit an introduction of the particular blocks provided by our graphical policy editor. Instead, we directly jump into policy authoring for the following video surveillance scenario and explain the employed blocks alongside.

3.1 Scenario: Privacy-aware Fall Detection Using Smart Video Monitoring

In our scenario, a hospital employs a smart video surveillance system for detecting falls of people in corridors and publicly accessible spaces, particularly in order to support the night shifts. We intend the system to operate according to the workflow depicted in

Fig. 7, which involves a *default mode* executing a fall detection algorithm on the video data captured by all cameras, *assessment modes* asking a nurse to differentiate between actual emergencies and false detections while preserving observed person's privacy as long as possible, and finally an *investigation mode* providing additional information for organizing emergency aid.

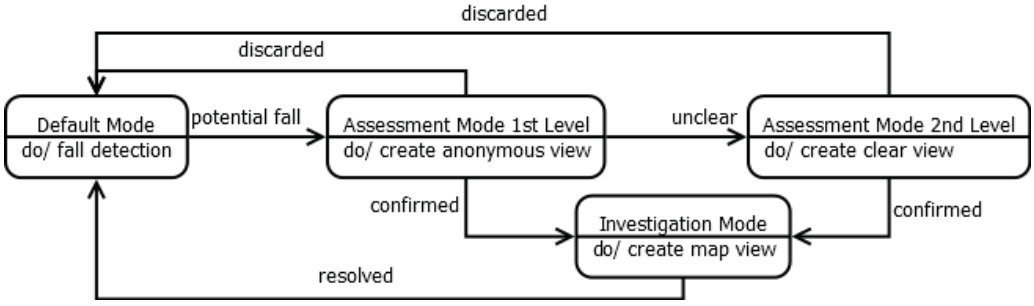

Fig. 7: Workflow for computer vision-based fall detection

3.2 Default Mode: Fall Detection Based on Computer Vision

While the system operates in its *default mode*, i.e., as long as no potential fall has been detected, data processed by the system cannot be accessed at all. Furthermore, a continously evaluated policy (cf. Fig. 8) demands that any collected data is deleted from the system's storages as soon as it is older than one minute. The storage *VideoArchive* buffers video streams from all cameras, while the storage WorldModelArchive buffers extracted meta data, such as positions of persons in the monitored area. The framing blue mechanism block represents the ECA rule structure of our policies. In addition to attaching a triggering event, a condition, and multiple actions, it can be configured to include predicates to be evaluated by some *external decider*. Policy enforcement is either *detective* or *preventive*: Detective mechanisms only react on events, while preventive mechanisms actually intercept events and are thus able to *allow*, to *inhibit*, to *modify*, or to *delay* them in case the condition has been evaluated to true.

Fig. 8 also shows the usage control policy exported into machine-readable XML format, which is understood by Fraunhofer IOSB's prototype systems *NurseEye* and *Network Enabled Surveillance and Tracking (NEST)* [7]. We omit the XML representations of the following policies due to space limitations.


```
<?xml version="1.0"?>
<policySet>
    <actionDescriptions>
        <actionDescription name="Default"/ >
    </actionDescriptions>
    <detectiveMechanism name="DefaultCleanUp">
        <description>Default mode. Periodically delete archived data.</description>
        <timestep amount="1" unit="SECONDS"/>
        <trigger action="Default" index ="ALL" isTry ="false"</trigger>
        <condition><true/></condition>
        <executeAction name="delete" pxp="WorldModelArchive">
            <constraint mode="OLDER" type ="age" unit="seconds" value="60"/>
        </executeAction>
        <executeAction name="delete" pxp="VideoArchive">
            <constraint mode="OLDER" type="age" unit="seconds" value="60" />
        </executeAction>
    </detectiveMechanism>
</policySet>
```

Fig. 8: Default mode policy, also exported as XML usage control policy

3.3 Assessment Modes: Privacy-preserving Elimination of False Alarms

Upon detecting a potential fall, the system enters the *1st level assessment mode*, which sends an alarm to the mobile device of a nurse and provides an anonymized view of the according camera's live stream and buffered video data of one minute previous to the fall detection event. As the policy *FallAssessmentL1* (cf. Fig. 9) states, the anonymized view is created using an image filter, which reduces observed people to blurred silhouettes in order to hide their identities. The nurse can proceed by either

confirming the incident, discarding the incident in case of a false detection or by requesting a *2nd level assessment mode* in case the provided view does not provide enough evidence for a proper assessment of the potential fall. Thus, false alarms of the fall detector that are recognized at this early stage do not lead to any privacy breaches for people in the range of the according camera.

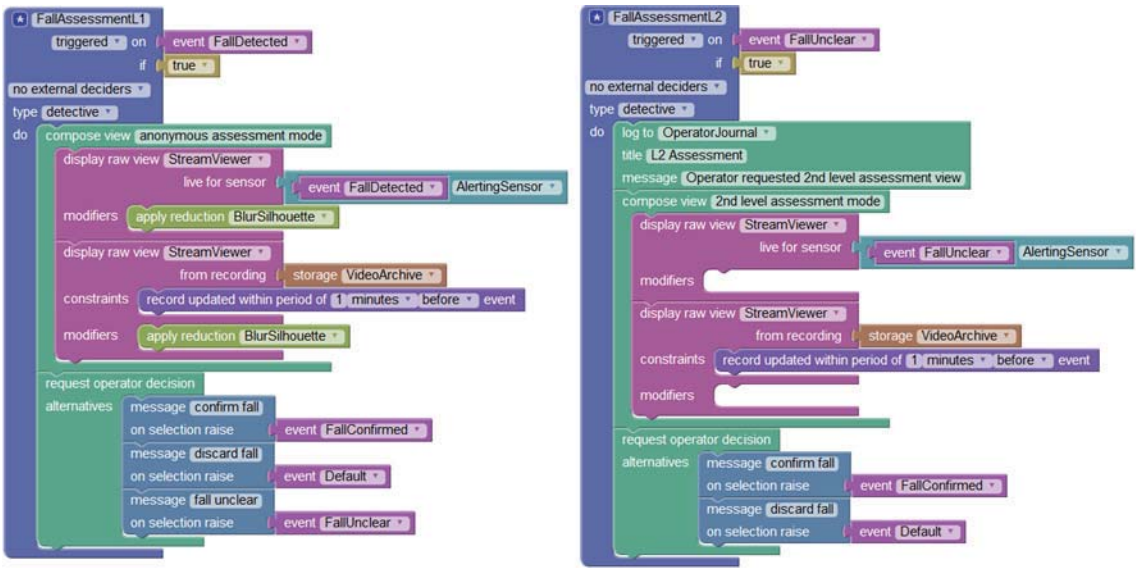

Fig. 9: Anonymized and clear assessment mode

In case the nurse cannot assess the incident properly, a *FallUnclear* event is induced, which triggers the *2nd level assessment mode*. The according policy *FallAssessmentL2* (cf. Fig. 9) grants access to the camera's live stream and the previous minute of recorded video data without enforcing the anonymization of released video data. However, each request to the 2nd level assessment mode is logged in the *OperatorJournal*, which can be accessed by employee representatives in order to detect misuse.

3.4 Investigation Mode: Handling Emergencies

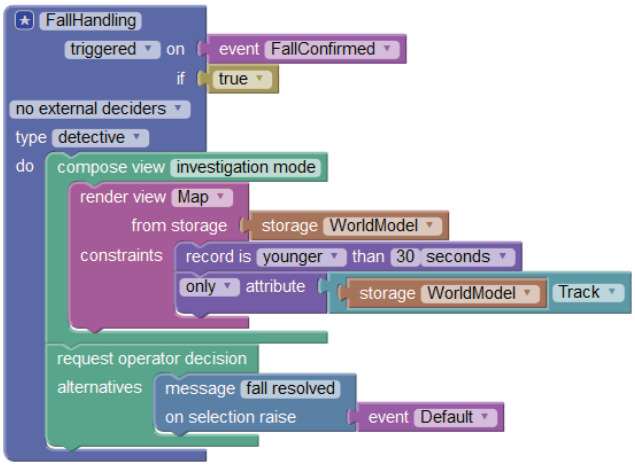

Fig. 10: Investigation mode: map view

Whenever a fall is confirmed by a nurse, the *investigation mode* of the system is triggered by the according *FallConfirmed* event. This mode creates a map view, which enables the visualization of meta data, such as positions of persons in the observed area. In our scenario, we only allow the system to release the position of the fallen person as well as the positions of other members of the medical staff. Three policies are required in order to specify these requirements.

The policy *FallHandling* depicted in Fig. 10 triggers the map view to be created and enforces global constraints: The map view is only granted access to records younger than 30 seconds from the storage *WorldModel* and, for each record, only the attribute *Track*, i.e., position records, are released. Furthermore, the system asks the nurse to give feedback after the emergency has been handled: As soon as the nurse confirms that the fall has been resolved, an *Default* event is raised, which triggers the system to switch back into its default mode.

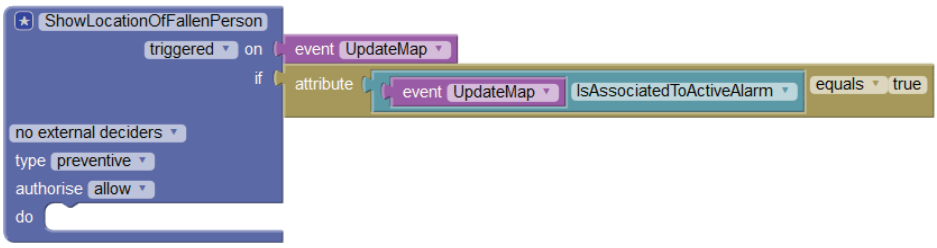

Fig. 11: Investigation mode: access to position of fallen person

Using the policy *ShowLocationOfFallenPerson* (cf. Fig. 11) we prevent the map view from accessing records other than the one associated to the active alarm to be handled, which refers to the record of the fallen person.

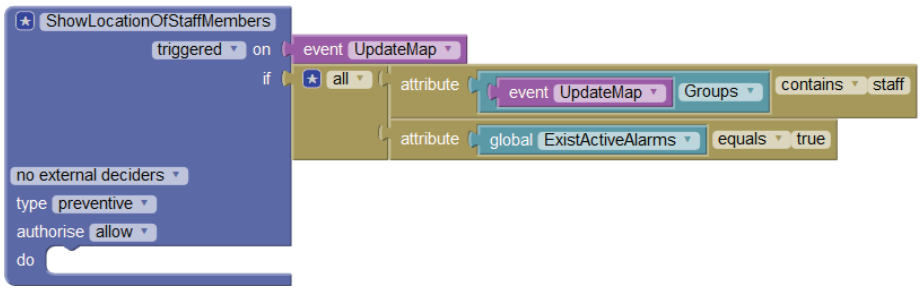

Fig. 12: Investigation mode: grant access to positions of medical staff

Finally, the policy *ShowLocationOfStaffMembers* as shown in Fig. 12 prevents the map view from accessing records other than those of members of the group *staff*. The condition also ensures that there is an active alarm whenever the records of staff members are read. By this means, the permission to access the staff members' positions expires as soon as the emergency has been handled and the system has returned into its default mode.

4 RELATED WORK

The meta model introduced in Section 2 is based on a generic usage control-enabled smart video surveillance architecture introduced in [7]. This generic architecture is derived from earlier works by Fidaleo et al. [1], Hampapur et al. [2], and Bauer et al. [4]. The meta model also incorporates the idea of establishing a "privacy grammar" in order to define privacy-sensitive (combinations of) attributes that may be embedded in or extracted from raw video data, but must not leak.

Employing dedicated assessment modes in order to implement a shifting trade-off between privacy and utility, which preserves observed people's privacy as long as at all possible is motivated by results of legal analyses conducted by Roßnagel et al. [5], as well as Bretthauer and Krempel [8]. Furthermore, the authors of [8] explicitly discuss the scenario of deploying smart video surveillance for fall detection in medical facilities, for which we instantiate our graphical policy editor in Section 3. They also argue in favor of the concept of assessment modes in order to prevent automated individual decisions entailing legal or other adverse consequences for the person(s) affected (cf. § 6 b German Federal Data Protection Act).

The presented work is clearly domain-specific, i.e., transferring the approach to another domain at least requires an according meta model to be created. However, to the best of our knowledge no prior work concerning graphical authoring of (usage control) policies has been published.

5 CONCLUSION

We presented a model-based policy editor for privacy-related requirements in smart video surveillance, which employs visual programming as an approach towards user-friendly and less error-prone policy authoring and visualization. Our meta model aims to capture the characteristics of concrete smart video surveillance systems and their applications. It can easily be adapted to future needs and upcoming features of smart surveillance systems just as the subset of usage control capabilities supported by the graphical editor can be extended in case more complex conditions have to be specified.

The obtained graphical representations of policies could also be employed to explain the operation of the system as well as the privacy mechanism in place to the people concerned in order to increase transparency. When used for this purpose, the level of abstraction of the graphical representation should be increased further in order to hide any technical details that are irrelevant from a data protection perspective.

REFERENCES

[1] D. A. Fidaleo, H.-A. Nguyen, and M. Trivedi (2004). *The networked sensor tapestry (NeST): a privacy enhanced software architecture for interactive analysis of data in video-sensor networks*. In Proc. 2nd ACM intl. Workshop on Video Surveillance & Sensor Networks, pp. 46–53.

[2] A. Hampapur, L. Brown, J. Connell, A. Ekin, N. Haas, M. Lu, H. Merkl, and S. Pankanti (2005). *Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking*. IEEE Signal Proc. Mag., 22(2), pp. 38–51.

[3] A. Pretschner, M. Hilty, and D. A. Basin (2006). *Distributed usage control*. Commun. ACM, 49(9), pp. 39–44.

[4] A. Bauer, S. Eckel, T. Emter, A. Laubenheimer, E. Monari, J. Moßgraber, and F. Reinert (2008). *N.E.S.T. - Network Enabled Surveillance and Tracking*. Future Security, 3rd Security Research Conference.

[5] A. Roßnagel, M. Desoi, and G. Hornung (2011). *Gestufte Kontrolle bei Videoüberwachungsanlagen - ein Drei-Stufen-Modell als Vorschlag zur grundrechtsschonenden Gestaltung*. In Datenschutz und Datensicherheit, 35(10), pp. 694–701.

[6] F. Kelbert and A. Pretschner (2013). *Data usage control enforcement in distributed systems*. In Proc. CODASPY, pp. 71–82.

[7] P. Birnstill and A. Pretschner (2013). *Enforcing privacy through usage-controlled video surveillance*. In 10th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 318–32.

[8] S. Bretthauer and E. Krempel (2014). *Videomonitoring zur Sturzdetektion und Alarmierung - eine technische und rechtliche Analyse*. In 17. Internationales Rechtsinformatik Symposion (IRIS) – Transparenz, pp. 525–534.