AAL PROGRAMME	**PersonAAL**	

Deliverable D1.1a

Architecture Specification (Initial)

Responsible Unit: CNR, ISTI, HIIS Laboratory

Contributors: USI

Document Technical Details:

Document Number	D1.1.a
Document Title	Architecture Specification (Initial)
Version	4
Status	Draftl
Work Package	WP1
Deliverable Type	Report
Contractual Date of delivery	30/09/2016
Actual Date of Delivery	30/09/2016
Responsible Unit	CNR-ISTI, HIIS Laboratory
Contributors	USI
Keywords List	Personalization, Web Applications, Architecture, Context-dependent systems
Dissemination Level	Public

Document Change Log:

Version	Date	Status	Author	Description
1	25/8/16	Draft	CNR	First Draft
2	15/9/16	Draft	USI	Draft with Security Integration
3	23/9/16	Draft	USI	Updates to Security Section
4	23/9/16	Draft	CNR	Revised Version for Review
5	29/9/16	Final	CNR	Final Version after Review

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

Contents

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

1 INTRODUCTION

The PersonAAL project aims at extending the time older people can live in their home environment, by increasing their autonomy and assisting them in carrying out their activities of daily living.
In the PersonAAL Project we plan to achieve this goal by providing the elderly intelligent and easy-to-use **Web applications** enabling them to receive ***personalized*** and ***context-dependent*** assistance directly in their own homes with the goal to improve their quality of life and also decrease healthcare delivery cost. In order to do this, in the PersonAAL project we plan to develop a technological platform (the "PersonAAL Platform") providing adaptation and persuasion features obtained through the specification of personalization rules and other means, which will be exploited by existing Web applications, thus improving elderly quality of life (Figure 1).

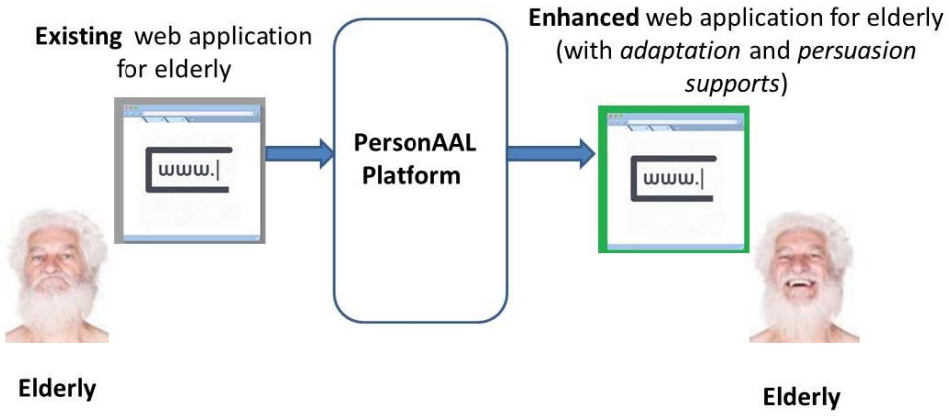

Figure 1: Abstract view of the PersonAAL

In this way, existing Web applications aimed at helping elderly people in living autonomously at home, by communicating with the PersonAAL platform will also be equipped/augmented with adaptation and persuasion features. This means that, if for instance we have a Web application targeted for the elderly, which e.g. help them in rehabilitation after a stroke event, by accessing the PersonAAL Platform, this application will also be capable of e.g. sending reminders to elderly **personalised according to their specific needs**, **adapt its user interface to the specific abilities (and disabilities) of the elderly and/or to the current surrounding environment**, etc.

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

2 HIGH LEVEL DESCRIPTION OF THE ARCHITECTURE

In the following, we will progressively refine the architecture of the PersonAAL Platform by describing the architectural modules needed for achieving the goals of the project in a *step-wise refinement manner*: starting from Figure 1 – where the architecture of the PersonAAL Platform is depicted just as a "black box"– we will progressively add further architectural modules to this black-box, explaining their objectives/goals and describing (in high-level terms) the main logical/functional relationships occurring between such introduced modules. The step-wise refinement approach should help the reader of this document in more easily understanding the rationale of the architecture. In order to further improve comprehensibility, in the various pictures of the architecture we also highlighted the elements that are progressively added/modified.

Therefore, by zooming into the Platform we can notice the two main modules named Adaptation and Persuasion, as there will be two logical/architectural modules in the PersonAAL Platform supporting such features. This is visually reflected in the following Figure 2.

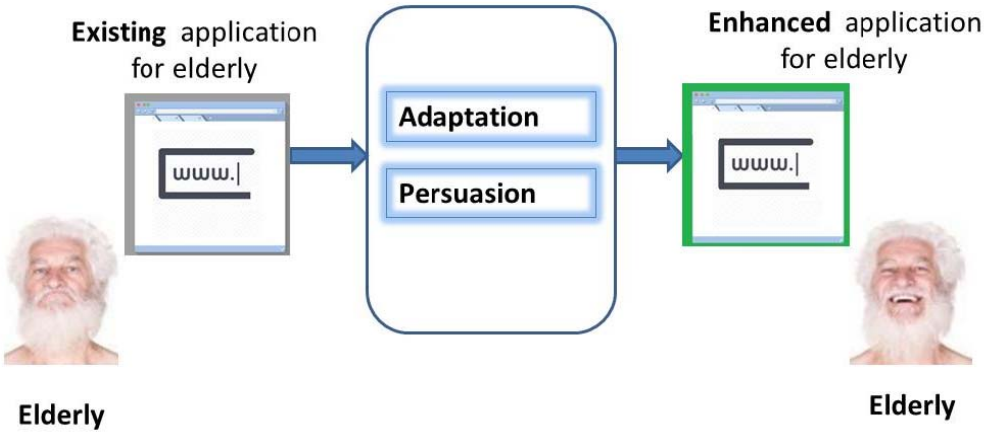

Figure 2: Architecture of the PersonAAL Platform containing adaptation and persuasion modules

The **Adaptation** module will provide support for adapting the remote assistance applications to the current context of use. Example: Suppose that the application for the elderly currently provides them with support for reminding to take a particular medicine at a certain time. The application could be instructed to send *vocally* some specific reminders to the elderly if it is detected that the elderly is currently sleeping and/or if it is known that speech is the preferred interaction modality for that particular elderly. This is an example of possible adaptation that the PersonAAL Platform could support, although further, more relevant examples will be identified in the continuation of the project according to the user requirements and needs. The **Persuasion** support is expected to identify

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

European Commission
Ministero dell'Istruzione, dell'Università e della Ricerca
FCT Fundação para a Ciência e a Tecnologia
Forskningsrådet
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

situations in which persuasive messages should be provided to users to stimulate them to change their current behaviour. In continuation of the previous example, if the application detects that the elderly did not take the medicine, even after the reminders, the Persuasion module could select one behaviour change technique appropriate for the elderly, and enrich following messages in a way that increases the chance that the elderly takes the medicine.

The "Adaptation" architectural module (see Figure 3) is expected to provide support for enabling the Web applications used by the elderly to have adaptive behaviour, which means that they could change according to e.g. relevant events occurring in the current context of the elderly, also taking into account e.g. their specific needs, requirements, (dis)abilities, etc.
This is where the role of the caregiver comes into play. Indeed, since elderly persons can span over a wide variety of different backgrounds/abilities/skills/ etc., it is important to offer **personalised support** to the elderly, so that the Web application will behave in a way that is actually effective for the particular needs of the considered elderly. However, it is not feasible to expect that most elderly will be able – on their own - to personalise their applications according to their specific needs. Therefore, we suppose that usually primary or secondary caregivers will provide support to personalise the applications of the elderly, on behalf of the elderly. Indeed, since the caregiver has a deep knowledge of the elderly, then s/he should be in the best position to identify the personalisation more suitable for the elderly. In order to enable the caregiver to specify such personalisation, we will provide also the caregiver with a Web authoring environment for defining the relevant customizations. So, in this regard, it is important to understand that the tool provided to the caregiver is also aimed at describing in which way the application for the elderly should adaptively behave. In other terms, the authoring tools provided to caregiver (which in the project is generally indicated as "**End User Development environment**") should enable the caregiver to specify the so-called "personalization rules" according to which the application used by the elderly should be adapted/behave in various contexts of use. Such adaptation rules will be provided to the "Adaptation" architectural module which will actually manage the adaptations in the PersonAAL platform (Figure 3). In general, End-User Development (EUD) approaches [4] aim to identify methods and tools to allow people who are not professional developers to create or customize their software applications.

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

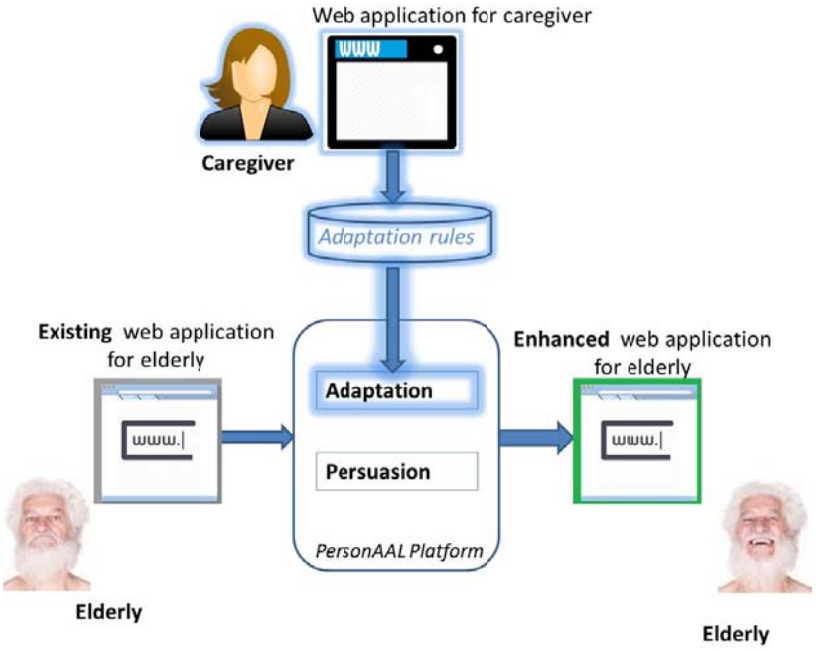

Figure 3: The personalization rules in the architecture

Figure 4 shows a refinement of the relationships occurring –at a high level- between the Adaptation module and the Persuasion module. The Persuasion will offer to the elderly some persuasive messages, which should be adapted before being made actually available to them in the application. For instance, once the Persuasion module has detected that a persuasive message should be delivered to the elderly, the Adaptation module could decide the best combination of interaction modalities to use is to provide the elderly with such messages rendered not only graphically but also vocally.

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

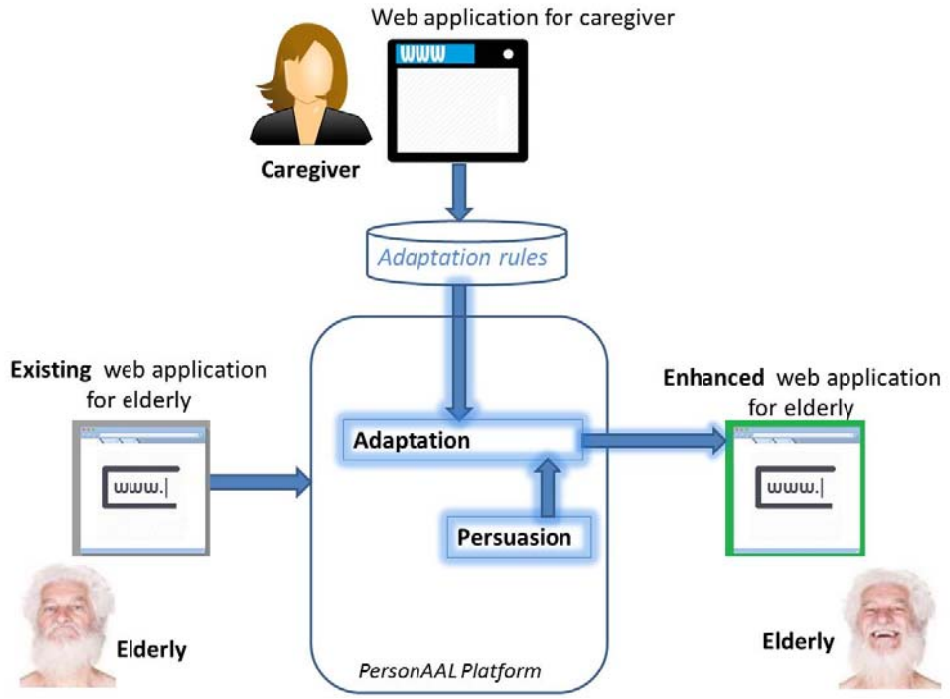

Figure 4: Relationships between the Persuasion module and the Adaptation module

Figure 5 shows another module, the "Behaviour Analysis", and highlights the architectural modules that get input from it. Indeed, in the PersonAAL Project we plan to monitor the elderly's activities and e.g. offer users assistance when their behaviour deviates from the expected one. The "Behaviour Analysis" module is the module expected to analyse the user behaviour and provide input: i) to the Adaptation module, because the outcome of the analysis of the user behaviour should be adapted before being delivered to the elderly (e.g., if the result is that the user has forgotten to take a pill, the alerting message should be adapted to user's needs), and ii) to the Persuasion module because, if the analysis of the user behaviour is that a persuasion message is needed, this should be communicated to the Persuasion module, which is in charge of this.

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

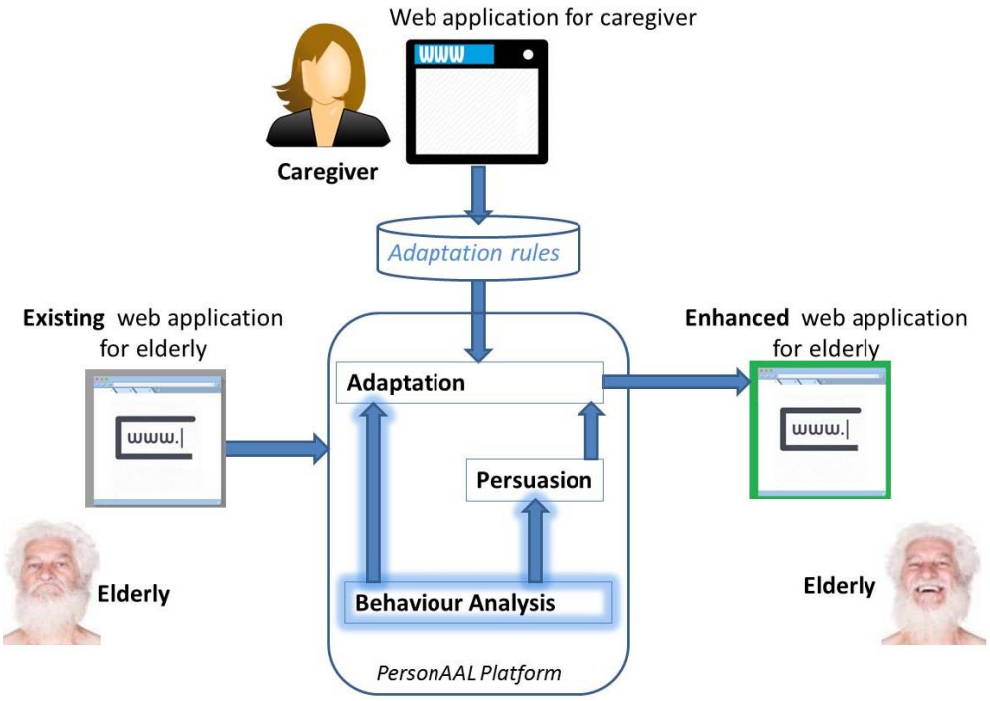

Figure 5: The "Behaviour Analysis" module provides input to the "Persuasion" module and to the "Adaptation" module.

Figure 6 shows the input needed by the Behaviour Analysis module (which is discussed in the PersonAAL Deliverable D1.2a): since it compares the actual user's behaviour in the current context (sensed in the real world) and the expected user behaviour (described in so-called task models), it needs input from the "Context Manager" module and data from the "Task models" repository.
The Context Manager is the module that is expected to gather and manage contextual data. It receives the contextual information from the Context Delegates, which are pieces of software associated with each sensor (can be standalone applications or embedded into the Web browser), and which are able to supply the context data detected by the associated sensors. For example, the Context Delegate for monitoring indoor user position can detect beacons (e.g. bluetooth modules) deployed in the environment, while the Context Delegate for monitoring user's physical activity relies on the device accelerometers data.

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

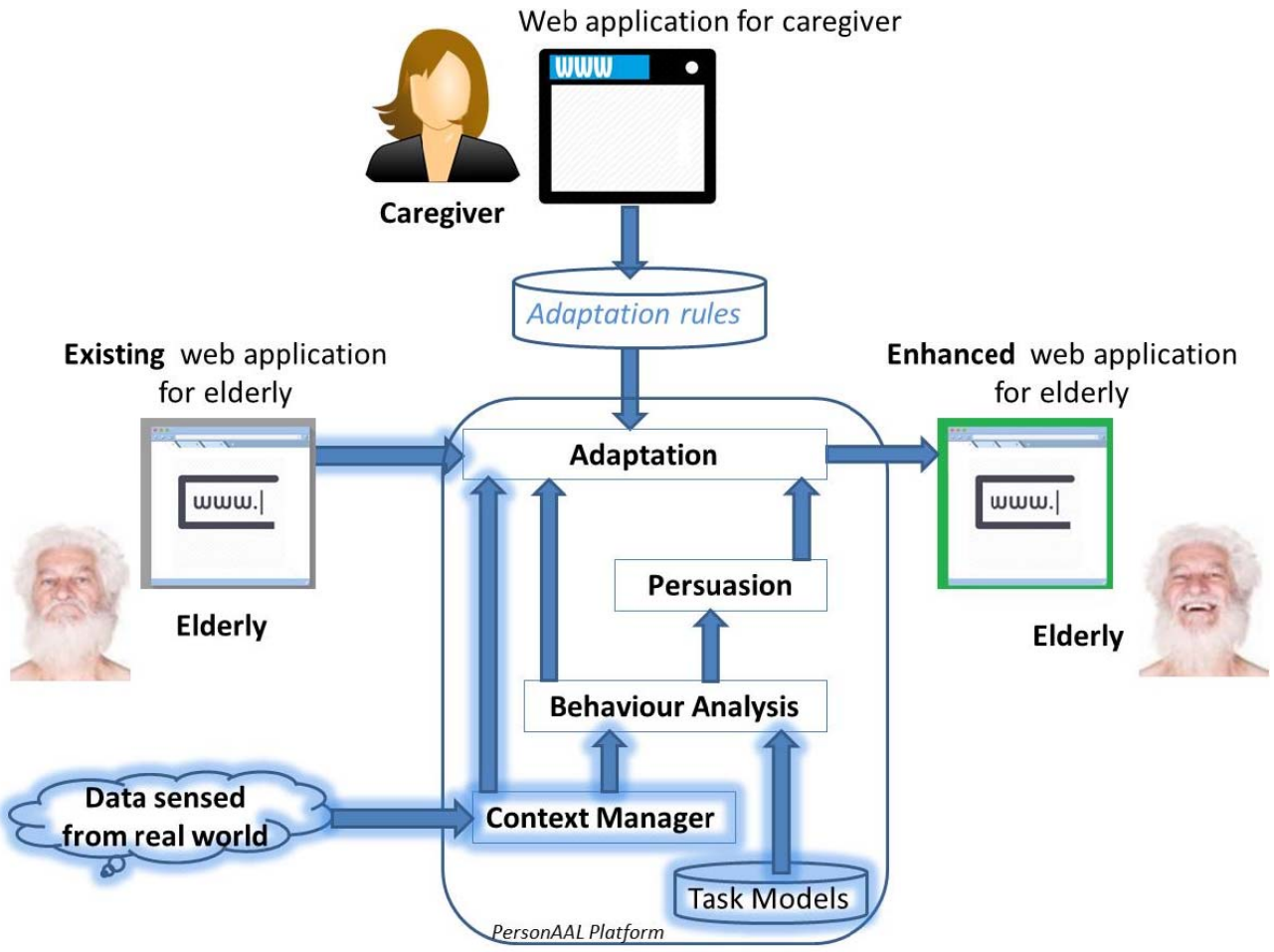

Figure 6: The Behaviour Analysis module interactions in the platform

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

3 PRIVACY AND SECURITY IN THE PERSONAAL PLATFORM

The PersonAAL platform collects and manages *personal information.* From a legal point of view, "personal information" is any data that "can relate to an identified or at least identifiable person", where "an identifiable person is one who can be identified, directly or indirectly, by reference to an identification number or to one or more factors specific to his physical, physiological, mental, economic, cultural or social identity" [1]. In the context of this project, such data is present due to three sources:

1. The user management inherent in the caregiver's Web authoring environment (which needs to manage data on both patients and caregivers),
2. The data collected from the context sensors (Context Delegates) and processed by the Context Manager, and
3. In the form of personal data collected and processed by the Web applications that PersonAAL personalizes.

Personal data collection and processing in PersonAAL must follow existing legislation such as the EU Directive 95/46/EC[1] (the "Directive"). From an architectural point of view, the most relevant is Article 17.1, which states that a data controller "must implement appropriate technical and organizational measures" that "ensure a level of security appropriate to the risks represented by the processing and the nature of the data to be protected." In addition, Article 8 lays down additional requirements for so-called "sensitive data", which for example is "racial or ethnic origin, political opinions, religious or philosophical beliefs, trade-union membership, and the processing of data concerning health or sex life." While the Directive needs to be transcribed into respective national law, the Directive nevertheless provides overarching guidance for Security and Privacy requirements in PersonAAL. For health data, national laws typically require additional levels of security, such as strong authentication (see section 3.2.3).

Note that while much of the data collected by context sensors may not directly point to an individual (e.g., timestamps of doors opening and closing), it nevertheless may be considered "personally identifiable information" (see the Article 29 Working Party's Opinion 4/2007 on the concept of personal data) and thus "personal data" in the sense of the Directive.

In order to provide adequate security in the PersonAAL platform, we need to consider both data "at rest" and data "in motion". These concepts will be discussed in the following section, before we discuss specific security mechanisms that will provide us with the required security.

[1] Note that 95/46/EC is a *directive*, which means it will need to be transcribed into national law by each EU member state. However, for our purposes the articles in the directive provide sufficient guidance. Also note that directive 95/46/EC will soon be replaced by the General Data Protection Regulation (GDPR), which will come into force on May 25, 2018 (a *regulation* applies directly in each member state). For the purpose of technical security implications, however, no significant changes occur, hence looking at 95/46/EC is sufficient to inform our requirements.

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

3.1 Security Objectives

Generally speaking, security measures need to protect data "at rest" and data "in motion". In both cases, data needs to be safeguarded against unauthorized access (confidentiality) and alteration, i.e., edits, insertions, and deletions (integrity).

3.1.1 Data in motion

Data in motion (also called data in transit) refers to any data transmitted and exchanged between different components of a given architecture [2][3]. This concept includes data transmitted over the network (public or private) from a server, platform, or cloud to endpoints; data exchanged between components of a given server, platform, or cloud; and data moving between different servers, platforms, or cloud providers. Data in motion is particularly vulnerable to unauthorized access as malicious entities may monitor data exchange over the network or malicious software may run within the processing environment. In order to protect data in motion, all data in transit must be encrypted or sent over an encrypted transmission channel (see section 3.2.1).

In the context of this project, data is considered to be in motion in three situations. First, Context Delegates (sensors) installed at user homes, mobile or wearable devices stream collected contextual (and health) data to the PersonAAL platform. Then, the collected data is processed and exchanged between different platform components such as Context Manager and Adaptation Engine to achieve a desired level of personalization (see Figure 6). Finally, the platform exchanges information with endpoints, i.e., Web applications that support elderly people at home. Considering these three cases of data in motion, the platform must provide secure and trusted connection with Context Delegates and Web applications according to the assumptions presented in section 3.3 and identified security requirements listed in section 3.4.

3.1.2 Data at rest

Security when data is at rest is mainly concerned with the storage of sensitive information on mass storage media and proper authentication and authorization of users and software components (processes) in accessing and processing the stored information [2]. Information stored within software systems is vulnerable to theft and modification that can compromise both confidentiality and integrity [3]. The solution to these vulnerabilities is to encrypt the information storage (see section 3.2.2). Encryption of the storage must be complemented with adequate access control mechanisms, i.e., authentication (see section 3.2.3) and authorization (see section 3.2.4).

The PersonAAL platform collects and stores contextual information about users, information about expected user behaviour in the form of task models, and adaptation rules provided by the caregivers. Since this information may contain personally identifiable data according to the Directive, it must be stored on a secure (encrypted) storage. The stored information must be accessible only by authenticated and authorized users and platform components with appropriate access and management rights. For example, the platform must enforce that *Context Delegates* can only write

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

contextual information about users to the storage and *Context Manager* can only read the contextual information from the storage. Also, the access rights must assure that *Caregivers* can only access information and records of elderly users they are responsible for.

3.2 Security Mechanisms

A security mechanism is a specific implementation of a security service designed to detect, prevent, or recover form a security attack. In order to protect data in motion and data at rest, a software platform may use one or more security mechanisms to provide an adequate level of confidentiality and integrity. Considering the processing of sensitive data, the PersonAAL platform must provide a secure (encrypted) transmission to protect data in motion and a secure (encrypted) storage with appropriate authentication and authorization protocols to protect data at rest. The following subsections briefly discuss these security mechanisms.

3.2.1 Encrypted Transmission

Encrypted transmission refers to the transfer of information over a secure communication channel to preserve confidentiality and integrity of data [2]. Encryption is used to ensure that only authorized people can process transmitted information. Even if a malicious entity intercepts the data transmission, only entities that possess the corresponding keys can access confidential information. A popular way to encrypt data is to use public-key cryptography (or asymmetric cryptography) based on a pair of public and private keys. The advantage over symmetric cryptography (which only needs a single, shared key) is that key distribution is simplified: each party only needs a public/private keypair, in contrast to symmetric cryptography where each pair of communication partners needs a separate key.

In general, encryption can be achieved at three different levels [3]:

- Endpoint/Application level – where data is encrypted by the server and the endpoint prior to transmission over the network. This level of encryption requires that all entities in the transmission implement and integrate the same encryption standard. It can provide a higher level of security but makes harder to ensure interoperability.

- Link/Network level – where data is sent over a secure network channel protected by standard encryption techniques and protocols such as Secure Sockets Layer (SSL), Virtual Private Network (VPN), and Secure Shell (SSH). Encryption on this level is preferred by modern software architectures.

- Proxy level – where data is sent to a trusted proxy service that encrypts information before transmitting it over the network. This might be an easiest way to encrypt data but it is not generally recommended.

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

3.2.2 Encrypted Storage

Secure storage protects confidential information in case of theft or unauthorized access [3]. The solution to secure storage is to encrypt information on the storage media using cryptographic techniques such as Data Encryption Standard (DES) or Advanced Encryption Standard (AES). Depending on the software architecture, confidential information can be encrypted at four possible levels:

- Application level – where an application implements specific encryption techniques to protect data before writing it to a storage media. Encryption at this level provides high level of security, but requires additional development and maintenance efforts.

- Database level – where encryption techniques are provided by a database vendor. Security and encryption options vary between vendors and database distributions and can be applied to secure entire database or specific subset of data within individual database tables.

- File level – where a software component handling encryption is integrated within the operating system. At this level, the software component intercepts all write and read calls to the storage media and encrypts and decrypts information according to a policy.

- Disk level – where all information written to a physical disk is encrypted. Encryption is either provided by the underlying hardware or achieved using a disk encryption software.

3.2.3 Authentication

Authentication is a process of verifying identity claims through credentials, i.e., verifying that users or software components are who they actually claim to be. A typical example of an authentication process is a login procedure using a username and password. While using simple and short password compromises security, using complex and long passwords may reduce the usability of software, especially for elderly users. Recently, many online platforms have started to use two-factor authentication to authenticate users. An authentication factor can be a knowledge factor (a password, a Personal Identification Number, or the answer to a secret question), an inheritance factor (biometric data such as fingerprint), or a possession factor (a hardware device providing a security token such as One-Time Password or a token compliant with the "Universal 2-Factor" authentication standard U2F, such as Yubikey[2]). By using two factors, attackers need to compromise two secrets (e.g., both the secret password of the user *and* stealing his crypto hardware device), making attacks much harder.

Compliance with the U2F standard provides strong authentication based on response-challenge procedure encrypted with a public and private key. Using an additional hardware-based security

[2] https://www.yubico.com/faq/yubikey/

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

token not only increases the security, but can also increase usability as simple to remember passwords can be used. The login password can even be written on the access devices because knowing the password does not grant the access without the hardware generating a security token. An example of the login procedure with 2-factor authentication using a hardware device is illustrated in Figure 7.

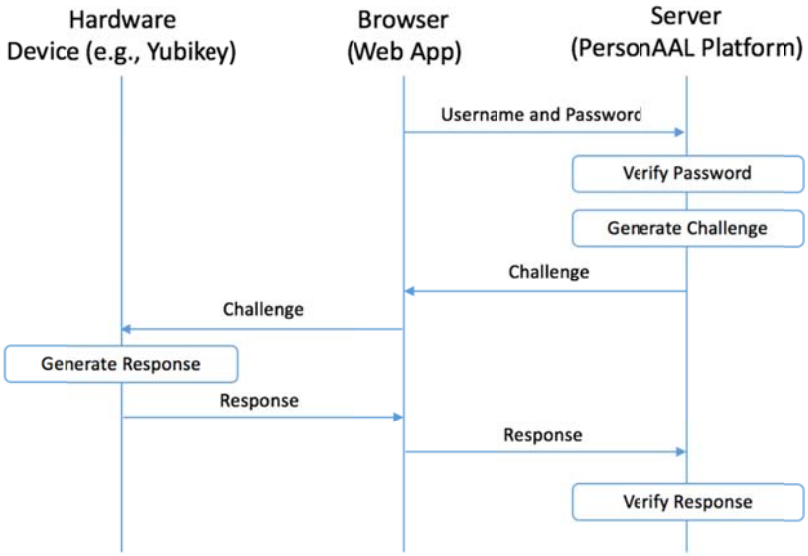

Figure 7 - An example of 2-factor authentication with a hardware device

3.2.4 Authorization

Authorization is the process determining and managing what users and software components can do within a software system after a successful authentication [2]. Authorization grants permissions and access rights to individual users and components to perform specific actions and process data. Typical examples of authorization include management of access rights in databases and file systems where a supervisor or administrator can grant or revoke explicit read, write, and execute rights to individual users and groups. The best practice in managing authorization is to follow the "principle of least privilege" [3]. The principle states that an individual, software component, or a system process within a software system should have only the minimal access rights necessary to perform a certain task.

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

3.3 Security Assumptions in PersonAAL

In the design of the PersonAAL platform there are several high level assumptions about the execution environment, the platform components, and security at home that have certain privacy and security implications.

- **Secure server platform.** We assume that the PersonAAL platform will be deployed on a secure server (both in terms of hardware and software). If the underlying server is not secure, any effort to secure personal data in PersonAAL is thwarted. We also assume that the server environment is maintained according to common and best security practices. This implies an up-to-date installation of all components, along with all necessary patches; a proper network port management; a proper management of access rights for development/testing/production; and prevention of common security attacks, such as distributed denial of service, at the server level.

- **Secure PersonAAL software.** We assume that no malicious code is present within the PersonAAL platform, and that all platform components use data according to the platform specification without purposely or accidently altering data during processing. If we assume non-trusted developers within the PersonAAL project, none of our security efforts will have an impact as none of the overall code-base could be trusted. This assumption implies an appropriate management of code development/review/release stages, and setting appropriate access rights for using and processing data by all platform components.

- **Secure PersonAAL sensors.** For the purpose of this project, we assume that all sensor data is uploaded to the platform through a secure and encrypted channel. This implies that sensors must implement appropriate authentication and encryption mechanisms in order to prevent injection of false sensor data into the platform that could initiate undesired adaptation and persuasion behaviour.

- **Security of the home**. The platform cannot protect and prevent physical theft of access devices and sensors, unauthorized data access within the home through unsecured devices, and unauthorized physical tampering with sensors. However, the platform must provide an appropriate mechanism for revocation of access rights for stolen or tampered devices and sensors.

- **Secure client platform.** The platform may rely on security options provided by the client access devices such as mobile phones, tablets, laptops, or desktop computers. This means that access to client devices should be safeguarded through standard login screens with the use of passwords, PINs, screen lock patterns, integrated biometric fingerprint scanners. or presence of trusted Bluetooth and NFC devices in a home environment. We hence must rely on client devices not being compromised. If an attacker gains access to a user's personal

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

device, all data being entered/collected at the client could potentially be read by the attacker. Given that PersonAAL will in many cases be used within a "Bring You Own Device" (BYOD) context, where participants use their personal laptops, tablets, or mobile phones, we cannot control the security of their devices.

3.4 Security Requirements in PersonAAL

The previously outlined assumptions and the overall design of the platform thus pose several high level security requirements, such as using secure (encrypted) communication, supporting full access mediation, providing strong and usable authentication, and relaying on peer-reviewed implementations of security protocols and mechanisms:

1. **Encrypted communication.** Transfer and exchange of data within the platform and between the platform and external endpoints must be explicitly encrypted (e.g., using AES) or sent using a secure protocol, such as Secure Sockets Layer (SSL) or Bluetooth. This includes streaming sensor data over short range wireless network to the access devices or directly to the platform using a secure API.

2. **Full access mediation.** Personally identifiable information, or any information that can be linked to individual, users must be encrypted when at rest. All entities (users, software components, and system processes) within and external to the platform must have appropriate credentials and authorizations to access such data. All instances of data access must be recorded and logged within the platform in order to support enforcement and security breach detection.

3. **Strong authentication.** The platform must provide strong and secure authentication such as universal 2-factor authentication for both elderly users and caregivers. Such a functionality should be also provided by all Web applications that access and use the platform.

4. **Usable authentication.** A high level of security usually influences and reduces the usability of software, especially for elderly people. A strong authentication procedure introduces an additional login information that can be troublesome for some users. This additional complexity can be counterbalanced by using a hardware device that can automatically generate a security token without user's intervention.

5. **Rely on trusted, peer-reviewed implementations of security protocols and mechanisms.** The design, development, and implementation of novel security protocols and mechanisms is out of the scope of this project. Instead, we must rely on existing, trusted implantations of common security standards, such as OpenSSL[3], OpenVPN[4], SSH, AES, OAuth[5], OpenID[6], or

[3] https://www.openssl.org/

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

U2F – both to limit development cost and to increase the overall security of the system. Only well-established, peer-reviewed implementations of security mechanisms that are widely used offer sufficient levels of security.

Subsequent drafts of this specification will ensure that these five key security requirements will be implemented in the PersonAAL platform.

[4] https://openvpn.net/
[5] https://oauth.net/
[6] http://openid.net/

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

4 HOW THE COMPONENTS COMMUNICATE AND INTERACT

In terms of implementation of the architectural modules, in the first year we have focused on the rule-based authroing environment, the context manager, and the adaptation support. In this section we provide technical indications about how the components developed communicate in the implementation.

In brief, the modules developed (at least in an initial version) are:

- Authoring Tool: Web-based application (RESTful Web services, Spring) for creating, saving, sharing adaptation rules. Support login via HTTP(S);
- Adaptation Rules: XML files defining adaptations of applications based on contextual events;
- Adaptation Engine: Web-based application (Java Servlets) that applies the adaptation rules to the applications;
- Context (Model) Manager: Web-based application (Java Servlets, RESTful Web services) that gathers contextual information and makes it available to other platform modules;
- Distribution Manager: service that manages UI distribution across multiple devices;
- User devices: stationary or mobile devices running a Web browser that access to the PersonAAL applications.
- Context Delegates (not visible in Figure 8), i.e. applications installed in mobile/stationary devices that gather data from environment/user/external services/device and send it to the Context Manager. Context Delegates can be implemented in various technologies: Java in Android/Windows, C# in Windows, JavaScript. Every Context Delegate should provide a mechanism for login (at the moment login is made simply through the userid).

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

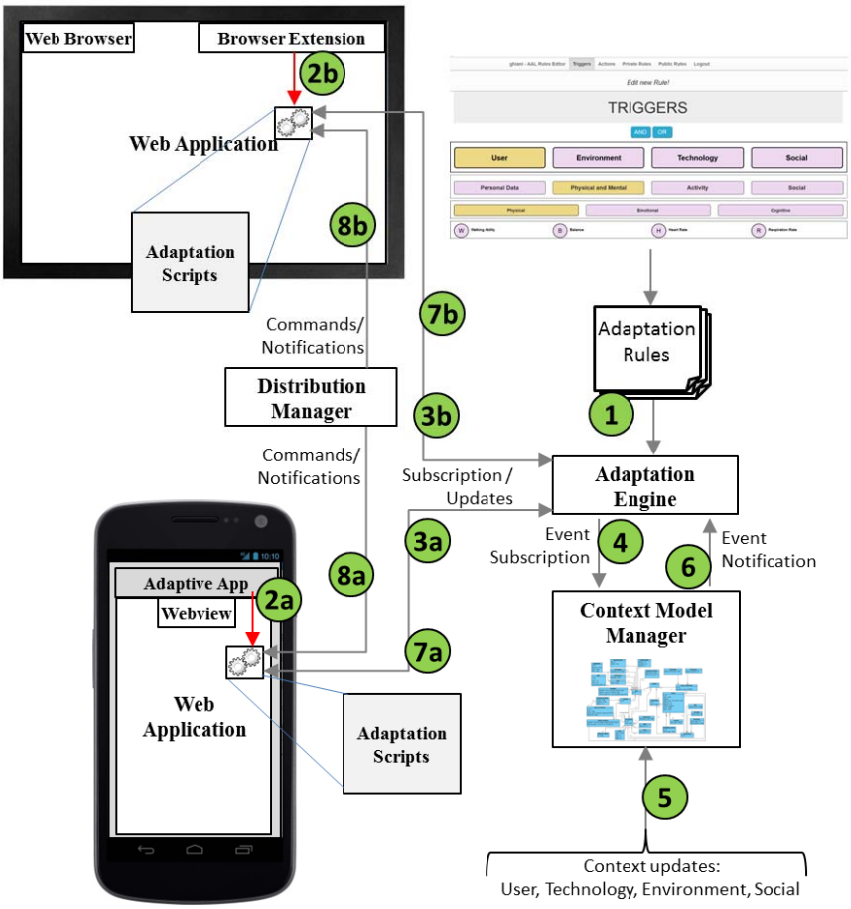

Figure 8. Main modules involved

When a rule is saved through the Authoring Tool, its XML description is sent via HTTP(S) POST to the Adaptation Engine, see Figure 8 (1). As soon as a PersonAAL application is accessed from the user device, the adaptation scripts are injected by a browser extension (2b) or by the inner functionality of the custom browser (2a). Injection is typically made by a browser extension in stationary devices, while it is done by a custom browser on mobile devices (e.g. Web browsers for Android do not allow the installation of extensions).

The application on the user device, when loaded, subscribes for updates by sending a request to the Adaptation Engine (3a/b) via HTTP(S) POST. This is done by the scripts previously injected into the

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

HTML. The Adaptation Engine extracts the rules content and subscribes to the Context Manager for the specified events, conditions (4).

One or more Context Delegates (not visible in Figure 8), continuously send updates on contextual parameters (5). The information is sent from the Context Delegate to the Context Manager via HTTP(S) POST (more detail on this communication, and more generally on the context manager implementation, is available in the deliverable D1.2a). When one or more events are verified and the rule conditions are met, the Context Manager notifies the Adaptation Engine (6) via HTTP(S) POST; then the Adaptation Engine forwards the list of modifications to be applied to the UI (7a/b). This is done via HTTP(S) AJAX polling or via WS(S). If a distribution is required, then the Distribution Manager is involved (8a/b). When the adaptation engine finds rules that involve user interface distribution across multiple devices it sends the actions for performing such distributions to the involved devices, which will execute them with the support of such distribution manager.

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

5 THE UNDERLYING LANGUAGE TO SPECIFY PERSONALIZATION RULES

In this section we describe the characteristics of the language used to implement the personalization rules. The language must specify which contextual events should be considered and the consequent modifications that should be performed on the interactive applications.
The rules are expressed through an ECA-based (Event, Condition, Action) format; where events are changes of context state, conditions are Boolean predicates referring to context state (they are optional) and actions are changes in the interactive application, in the state of the appliance or they may activate some functionalities.

The "event" part of a rule can be either an elementary event or a complex event (e.g. composed through some Boolean operators).
Conditions appearing in a rule can be either elementary (e.g. Boolean predicates) or complex conditions (composition of elementary conditions).
In the "action" part of a rule there is either one single action or a sequence of actions.

Rules can have priorities, which are useful when multiple, conflicting rules occur simultaneously, thus priorities act as a mechanism to identify the rule which is the most likely to be triggered.

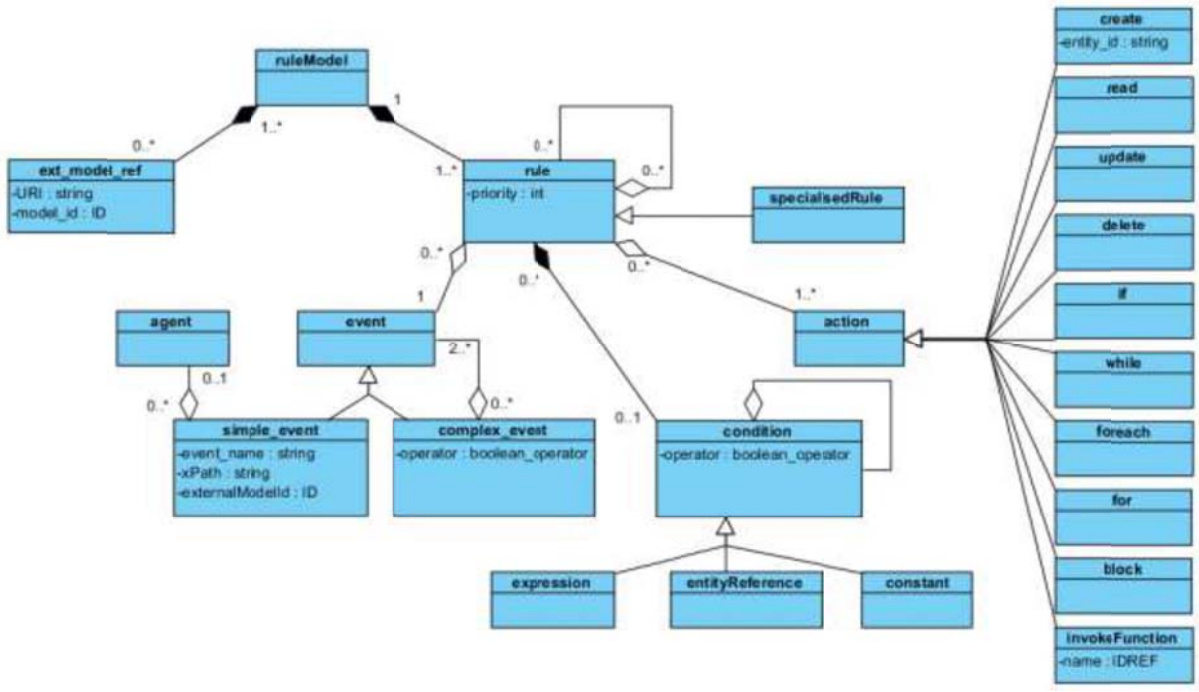

Figure 9: Metamodel of the language to specify the rules

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

Figure 9 shows the metamodel of the language: as a root we have the ruleModel that contains a list of rules composed of <event, condition, action>.
Events are those occurring in the context. Conditions are Boolean expressions that have to be satisfied in order to trigger the execution of actions. Actions are all the actions that can occur in an interactive application: create, read, update, delete, if, while, foreach, for, block, invokeFunction.

5.1 Some rules examples

In this section we describe some examples of modelling personalization rules by exploiting the presented language.
By using different types of sensors or devices we are able to monitor:

- the user activity (walking, running, stationary);
- user posture (standing, sitting, laying down);
- the presence of the user in a room or in the house;
- which task the user is performing;
- physiological parameters (hart rate, respiration rate);
- light level in a room.

Thus we are able to define this kind of rules:

- IF user is laying down DO close bedroom blind;
- WHEN user enters inside the bathroom, DO turn-on and set bathroom light colour to yellow;
- IF user age is more than 80, DO change font size to 40px;
- IF heart rate is more than 100, SEND an alarm to the doctor;
- IF user task is cooking, DO turn on kitchen light;
- IF user is sleeping and time is 11:00 AM, SEND alarm to the halt care assistant;
- IF user leaves home and time is after 23:00 SEND alarm to the relatives

The corresponding XML definition of the first rule is shown in Figure 10

```
"<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rule priority="1" id="821431317" name="rule1" xmlns="http://www.serenoa-fp7.eu/">
    <condition operator="eq" event_id="Posture">
        <entityReference xPath="User/UserPosture/@postureType" externalModelId="ctx"></entityReference>
        <constant value="laying_down" type="string"></constant>
    </condition>
    <action>
        <update>
            <entityReference xPath="/Bedroom/blind/@state" externalModelId="ctx"></entityReference>
            <value>
                <constant value="close" type="string"></constant>
            </value>
        </update>
    </action>
</rule>
```

Figure 10: Example of the XML definition of a rule

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

6 THE PERSONALIZATION RULE EDITOR

The editor to personalize the adaptation rules follows the End-User Development (EUD) paradigm that it is based on the idea that end-users can adapt the software to their own goals. It is a tool obtained through a meta-design approach [5] for Internet of Things applications. It needs first to be configured for the specific application and context of use for which the personalization rules has to be indicated. Indeed, some events and actions depend on the actual possibilities of the application to customize and the context of use addressed (e.g. it should not be possible to change the colour of the lamps if in the context of use considered coloured lamps are not supported).

The Authoring Tool is Web-based and enables the creation of (multiple) trigger action rules in an intuitive manner. It is flexible in the order in which rules can be created: users can start either from triggers or from actions (see Figure 11), and it provides the possibility to reuse previously specified rules as a starting point in order to create new ones.

Figure 11: The User Interface of the AAL Rules Editor

Each rule is composed of two parts: a trigger part, and an action part, thus, its basic structure is the following one:

IF/WHEN <*trigger_expression*> DO <*action_expression*>.

The *trigger_expression* defines the event(s) and/or the condition(s) that activate the rule application. The *action_expression* defines the action(s) that should be carried out when the rule is triggered.

Trigger_expression and *action_expression* can be either an elementary or a composite expression.

We also provide the possibility of combining multiple triggers and/or multiple actions). As both the evaluation of events and the evaluation of conditions result in Boolean values, multiple triggers can be combined by using basic Boolean operators, namely AND and OR.

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

It is also possible to define sets of actions to be sequentially performed.
On the top of the editor there is a continuous feedback in a subset of natural language indicating the current edited rule in an easy-to-understand language. The distinction between events and conditions is highlighted by using different keywords: "WHEN "is used for events, whereas "IF" is used for conditions.
In the editor, the selection of the relevant concepts is performed by navigating in the hierarchy of concepts associated with each contextual dimension. For this purpose, each contextual dimension is refined by going through a number of conceptual levels until basic elements are reached. In order to show only the relevant elements to the user, the refinements are presented in an interactive manner (i.e. only the decomposition of the element currently selected by the user is expanded), and highlighting the element(s) selected by the user through a different colour. The selected concepts are highlighted in the tool by the yellow colour. Figure 12 shows an example in which the yellow rectangles represent the path from the dimension to the event/condition aspect: User -> Activity Position -> Relative Position.

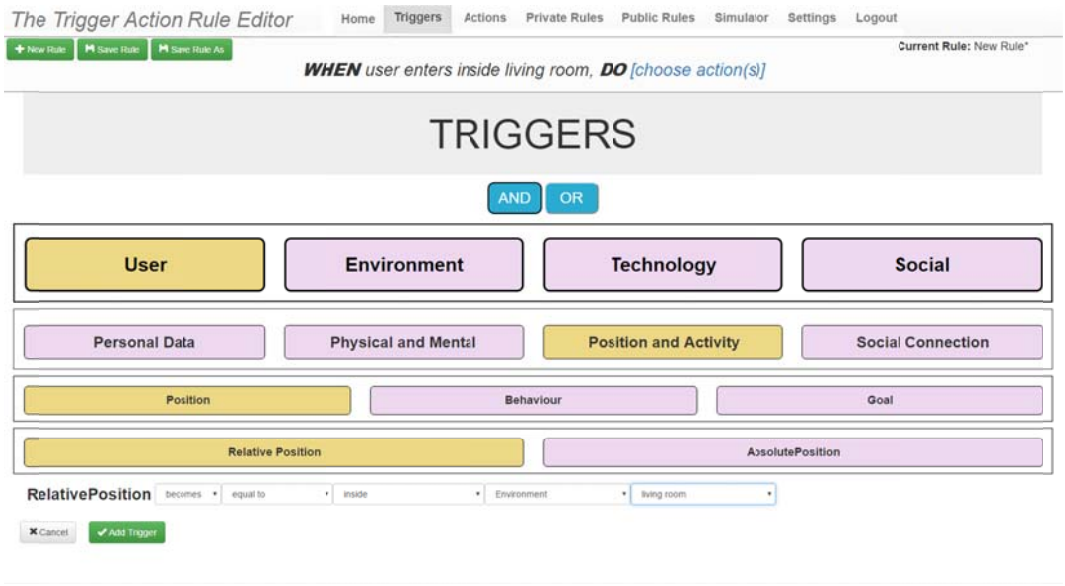

Figure 12: The editing of a trigger expression in the tool

In a similar way users can specify the desired actions. The authoring environment supports the classification of possible actions and when one of the main action types is selected, the tool shows the supported corresponding application-dependent actions .
Figure 13 shows an example in which the action is "set living room light colour to white".

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

Figure 13: Definition of an action by using the tool

In addition, after some rules are created they can be saved for possible reuse in different scenarios (Figure 14). When the user wants to re-use a previously created rule s/he has first to select it, and then activate its editing: after this the tool allows the user to change the rule by selecting its composing parts in the natural language section and updating some values or attributes in its specification. Furthermore, it is possible to add some more triggers/actions, or even delete some parts (e.g. one of the actions and/or one of the triggers).

The rules saved in the private workspace of the user can be uploaded in a public repository that it is shared between all the registered users; each user can also import a public rule in his private workspace.

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

Figure 14: Example list of saved rules.

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

7 HOW APPLICATIONS APPLY PERSONALIZATION RULES

The Adaptation module receives the adaptation rules defined by the caregivers tool and communicates with the applications and the context manager. In terms of software the Adapation module mainly consits of an Adaptation Engine.

The **applications** should contain simple JavaScripts that allow them to subscribe to the Adaptation Engine and establish a connection with it in order to be notified whenever an adaptation must be performed.

In technical terms, subscription is done by invoking a REST service and sending the application name, the user name and the action format preference (JSON or XML). If the client has subscribed correctly, the adaptation engine load the rules related to the user and to the application and then it subscribes itself to the context manager to receive updates about the entities monitored by the context manager. By relying on such standard Web technologies, any commercial device with a Web browser (smartphone, tablet, notebook, etc.) is technically able to host and properly execute the scripts.

The main aim of the communication is to inform the application whenever one or more adaptation rules are triggered. When this happens, the Adaptation Engine sends to the scripts included in the application the set of actions that define which Document Object Model (DOM) elements of the Web application have to be modified and how (e.g.: text content modification, style modification), or if new elements have to be added/removed (e.g., a paragraph, an image, etc.) from the document.
An action represents the description of how the interactive application should change in order to perform the requested adaptation;
There are different types of actions:

- Modifications of the user interface (presentation / content / navigation)
- User interface distribution on relevant devices
- Activation of some functionality
- Generate alarms (to highlight some potentially dangerous situations)
- Send reminders (to indicate some task that should be accomplished)
- Provide suggestions (persuasion)
- Send prompt messages
- Explanation messages
- Change the state of an appliance (light, fridge, …)

Application scripts are able to interpret such messages and perform the required actions. For example, the application adaptation script contains a function that implements the update action with the following signature: updateAction(elements_to_update, action_to_perform). Thus, if the

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

adaptation request that has been received is to perform an “update” action, then the application adaptation script will receive from the server information about: the element(s) to update (e.g. the element having ID=“div_1”) specified through an xPath string; and the type of update requested (e.g. set the backgroundcolour attribute of the page to a specific value, for instance: yellow).
At CNR adaptation scripts that can be easily included in existing Web applications have already been implemented. Such inclusion can be done independently of the type of Web application and on its degree of dynamicity. For instance, the script can be included in a static HTML page as well as on a PHP file.

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

8 CONCLUSIONS

We have presented the design of the architecture of the PersonAAL platform. We also provided indications about how the various modules can cooperate at implementation level and the securuty requirements.
A first version of the architecture implementation has been delivered. The second year will be dedicated to further improving the implementation of the existing modules and obtaining an implementation also for the remaining modules to complete the architecture. In addition, we plan to carry out usability tests for the personalization tools with caregivers.

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

9 REFERENCES

[1] European Union Agency for Fundamental Rights, Council of Europe, "Handbook on European Data Protection Law", 2014.
[2] Qing Li and Gregory Clark, "Security Intelligence – A Practical Guide to Solving Enterprise Security Challenges", Wiley, 2015.
[3] Cloud Securoty Alliance, "Security Guidance for Critical Areas of Focus in Cloud Computing V3.0", White Paper, Cloud Security Alliance, 2011.
[4] H.Lieberman, F.Paternò, W.Wulf (eds), End-User Development, Springer Verlag, ISBN-10 1-4020-4220-5, 2006.
[5] G. Fischer, E. Giaccardi, Y. Ye, AG Sutcliffe, N. Mehandjiev, Meta-design: a manifesto for end-user development, Communications of the ACM 47 (9), 33-37, 2004.

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.