www.fit4work-aal.eu

FIT4WORK

SELF-MANAGEMENT OF PHYSICAL AND MENTAL FITNESS OF OLDER WORKERS

CO-FUNDED BY

PARTNERS

Project coordinator: Poznań Supercomputing and Networking Center, ul. Jana Pawła II 10, 61-139 Poznań, Poland, email: fit4work@fit4work-aal.eu

Cloud services specification

Ambient Assisted Living Joint Programme project no. AAL-2013-6-060

Deliverable 5.3, version 1.0

Lead author: Cosmin Carjan, Teamnet

Co-authors: Cristian Neagu, Teamnet

Maciej Bogdański, Poznań Supercomputing and Networking Center

Aleksander Stroiński, Poznań Supercomputing and Networking Center

Project coordinator: Poznań Supercomputing and Networking Center, ul. Jana Pawła II 10, 61-139 Poznań, Poland, email: fit4work@fit4work-aal.eu

Published on 30th of April, 2016

The Fit4Work project is co-financed though the AAL Joint Programme by:

- European Commission
- National Centre for Research and Development, Poland
- Ministry of Industry, Energy and Tourism, Spain
- Executive Agency for Higher Education, Research Development and Innovation Funding, Romania
- Ministry of Higher Education, Science and Technology, Slovenia
- The Netherlands Organisation for Health Research and Development (ZonMW), The Netherlands

Table of contents

Project coordinator: Poznań Supercomputing and Networking Center, ul. Jana Pawła II 10, 61-139 Poznań, Poland, email: fit4work@fit4work-aal.eu

1. Introduction

The Fit4Work project aims to develop an innovative easy-to-use and unobtrusive system that offers support to older workers and the relevant stakeholders in reducing and managing physical and mental stress resulting from their occupation.

The Fit4Work objective is to develop an innovative system able to help to “preserve cognitive and physical capacities“ of older workers. It tackles the biggest occupational challenges of older adults, which are “physical strain and mental stress”. The project thus aims to “enable older adults to continue managing their occupation” and at the same time it supports “preserving health and motivation to remain active”. Through this the proposed solution promotes “health and well-being”, both in the workplace and at home.

The Cloud environment allows to host all back-end services of the system. This deliverable defines how cloud infrastructure is used within the system together with defining relevant access protocols and policies and, at the same time, describes the technologies used to create it. Service oriented architecture concepts are used for accessing the back-end services; security and efficiency are at the core of access protocols.

This document starts from the general system architecture overview, describing interactions between system components on a high-level view and afterwards goes into more details in regards to the services and cloud implementation.

The current report is divided in 2 major parts:

- General system architecture overview in Section 2
- Details on the planned Cloud services layer details in Section 3

2. General system architecture overview

As described in deliverable 5.1 System Architecture Definition (Carjan et al, 2016), the general system architecture has been defined in accordance with all involved partners, as shown in the diagram presented in Figure 2.1.

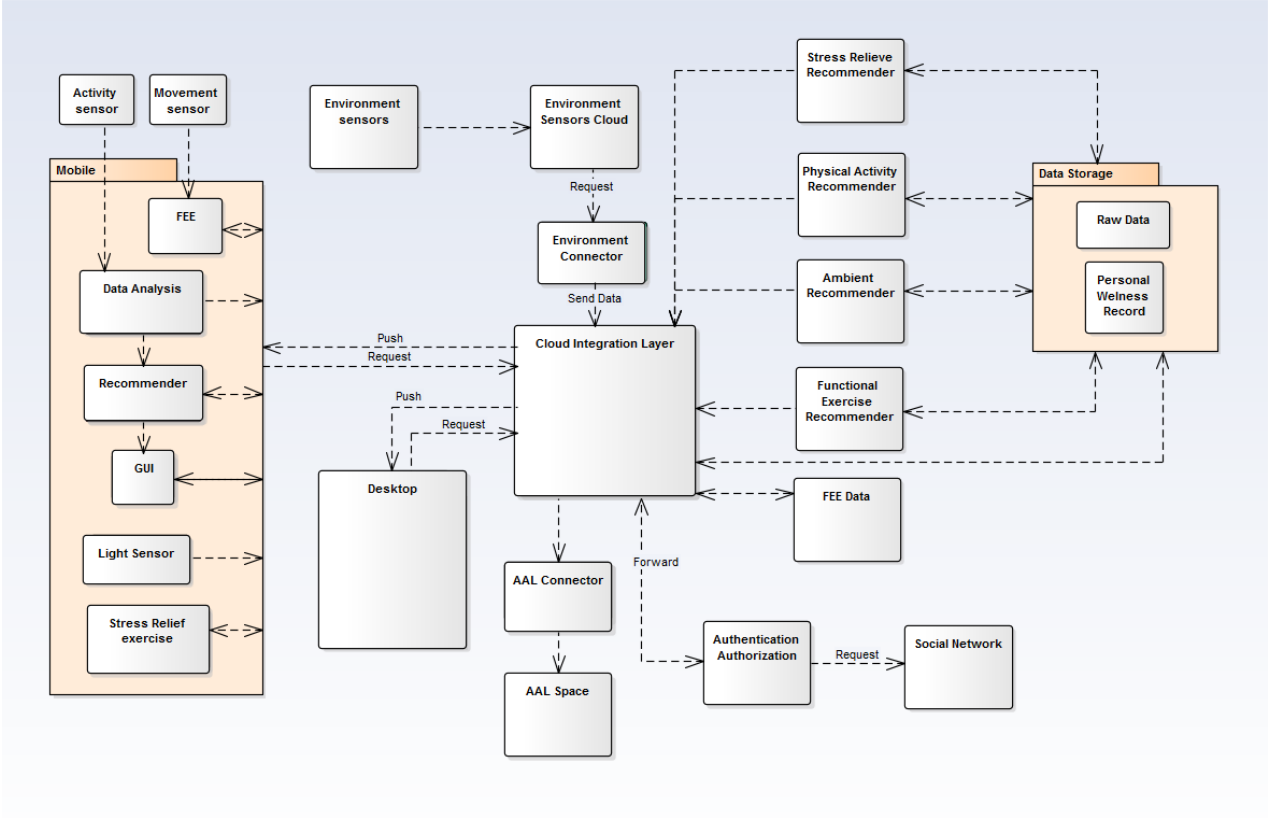

Figure 2.1 General Architecture of the Fit4Work system

The Fit4Work system could be separated in four different categories of components:

- Sensors (activity, movement and environment)
- User Gateways (mobile & desktop)
- Cloud Components
    a. Cloud Integration Layer
    b. Recommenders
    c. Data Storage
    d. Authorization
    e. FEE (Functional Exercise Engine) Data
- Connectors
    a. Environment Sensors Connector

Project coordinator: Poznań Supercomputing and Networking Center, ul. Jana Pawła II 10, 61-139 Poznań, Poland, email: fit4work@fit4work-aal.eu

b. AAL Connector

The sensors are used to collect data from the environment or from the user, in order to monitor the needed parameters. The information will be sent to the backend services, like the Recommenders, for the example. The details regarding the sensors will be highlighted in the corresponding deliverables.

The Fit4work Mobile and Desktop Applications are the **user gateways** into the Fit4Work system. They provide the user with information about the Fit4Work environment and, in the case of the mobile application, the other components with data from the various sensors connected to the mobile device.

The Cloud Components provide essential functionality of the Fit4Work Cloud system (Cloud Integration Layer, Recommenders, Data Storage, Authorization)**.** This section is the main focus of this document and the individual components will be detailed in the following chapters.

The connectors allow the Fit4Work system to communicate with external components or frameworks.

3. Cloud services

3.1. General description

The Cloud Integration Layer is at the center of the Cloud related interactions, as shown in Figure 3.1.

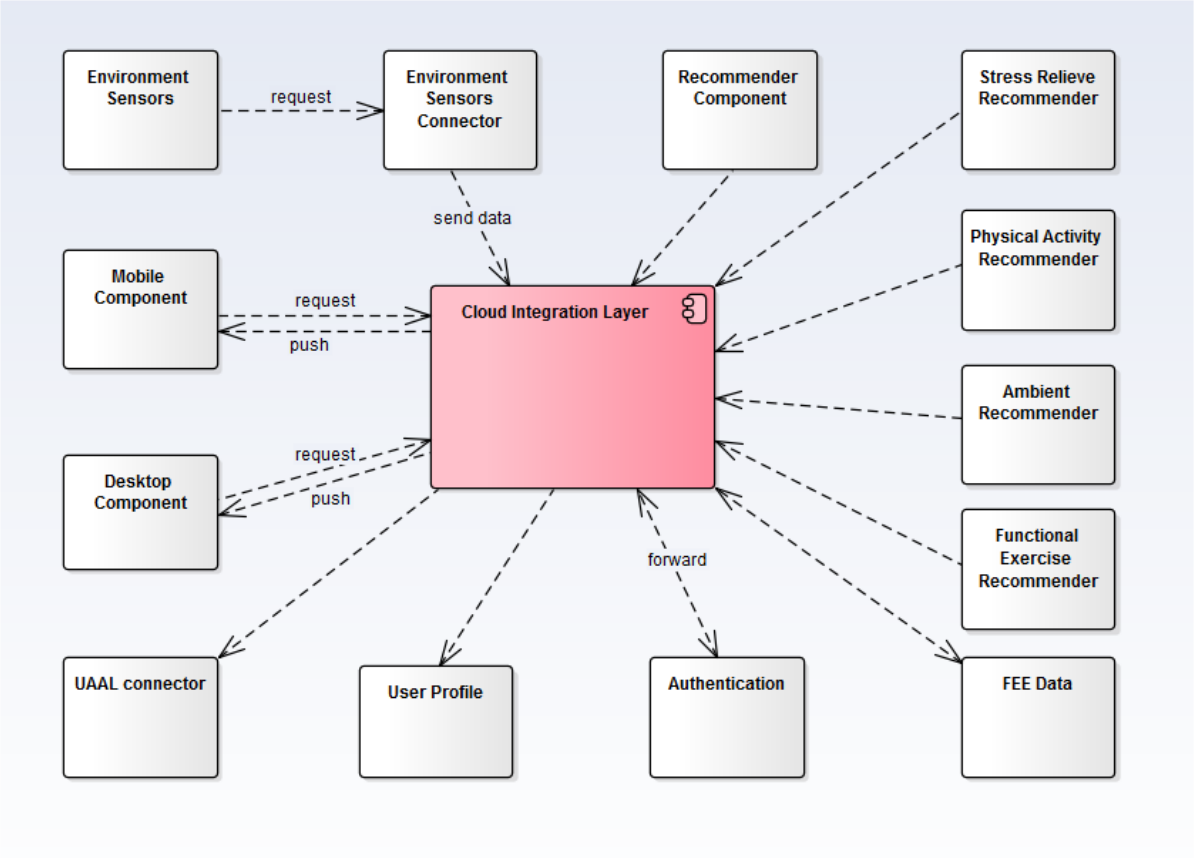

Figure 3.1 The Cloud Integration Layer

The **Cloud Integration Layer** functions as a link between Fit4work system's components and it is defined by a series of services that will lay out diverse methods of accessing and delivering data to each and every component integrated in the system.

The Cloud service is similar in behaviour to a proxy server that is able to adapt both the request from the client and the response from the target server.

From the user gateways' point of view, the Cloud Integration Layer is the one component that enables them to:

- login into the system, using the **Authentication** component
- create and manage a user account, storing the user data in the **Data Storage**

- receive recommendations from the **Stress Relieve Recommender, Physical Activity Recommender** or **Ambient Recommender**
- receive data about functional exercises, using the **Functional Exercise Engine**
- connect to other AAL-enabled applications through the **AAL Connector**
- connect to external services, such as the one for ambient monitoring sensors, through **Environment Sensors Connector**
- connect to **Social Networks**

The Cloud Integration Layer is composed of a set of services that will expose different methods of accessing available data in the Data Storage service. The role of this system is to process requests from the user gateways, find in the Cloud recommender management module relevant information (from the **Personal Wellness Record** for example) and send it back in a format that is understood by the Mobile Application.

The Cloud Integration Layer will also process requests from the Cloud services, not just the ones coming from the user gateways. This will be determined based on the system requirements, but one example would be the interaction between the data coming from the Environment Sensor Connector and the different recommenders.

The Cloud Integration Layer provides data from the sensors to the Data Storage. Any other data supplier (device) can be added to the Cloud Integration Layer because the process is transparent and the Data Storage service is not aware of the actual data provider.

The Cloud Integration Layer is able to facilitate a number of functionalities to the **Recommendation Services** (Stress Relieve Recommender, Physical Activity Recommender, Ambient Recommender):

- provide information about ambiental environment and information from the Activity Sensor.
- provide information about the user from the User Personal Wellness Record;
- provide full or part of the user's history (e.g.: last activity performed) from the Data Storage;

The only component that has a record of the physical addresses of every other component is the Cloud Integration Layer. When a component is moved, the system will still be able to function in normal parameters, and the services that would otherwise interact directly with it do not need major modifications.

Moreover, any change in the interface of that component will mainly impact the Cloud Integration Layer, with minimal changes requested in the components that are consuming the data.

This approach is preferred, due to several reasons:

- transparency of component location throughout the entire system
- scalability

- flexibility

3.2. Framework candidate for the Cloud Integration Layer

The main focus is to build an easy to implement solution in order to be maintainable, but flexible enough to adapt to the needs and requirements of every part of the system.

In order to identify the best solution for the needs of Fit4Work, several alternatives have been explored:

- Talend Open Studio for Data Integration
- Spring Framework
- Apache Struts 2
- Scalatra

3.2.1. Talend Open Studio for Data Integration

Talend Open Studio for Data Integration (Talend, 2016) is an open source data integration product developed by Talend and designed to combine, convert and update data in various locations across a business. It operates as a code generator, producing data-transformation scripts and underlying programs in Java. Its GUI gives access to a metadata repository and to a graphical designer.

This solution is an Integrated Development Environment based on Eclipse RCP. Users design individual jobs using graphical components for transformation, connectivity, or other operations. The jobs created can be executed from within the studio or as standalone scripts.

An Eclipse RCP application is a stand-alone application based on Eclipse platform technologies. An Eclipse application consists of individual software components. The Eclipse IDE can be viewed as a special Eclipse application with the focus on supporting software development.

Talend products are used by a large number of companies, including Alcatel-Lucent, Allianz, AOL, GROUPON, Lenovo and Orange.

ETL processes retrieve the data from all operational systems and pre-process it for the analysis and reporting tools.

Talend Open Studio (see Figure 3.2) for Data Integration offers nearly comprehensive connectivity to:

- Packaged applications (ERP, CRM, etc.), databases, mainframes, files, Web Services, and so on to address the growing disparity of sources.
- Data warehouses, data marts, OLAP applications - for analysis, reporting, dashboarding, scorecarding, and so on.
- Built-in advanced components for ETL, including string manipulations, Slowly Changing Dimensions, automatic lookup handling, bulk loads support, and so on.
- Data migration/loading and data synchronization/replication are the most common applications of operational data integration, and often require:

- Complex mappings and transformations with aggregations, calculations, and so on due to variation in data structure
- Conflicts of data to be managed and resolved taking into account record update precedence or "record owner"
- Data synchronization in nearly real time as systems involve low latency.

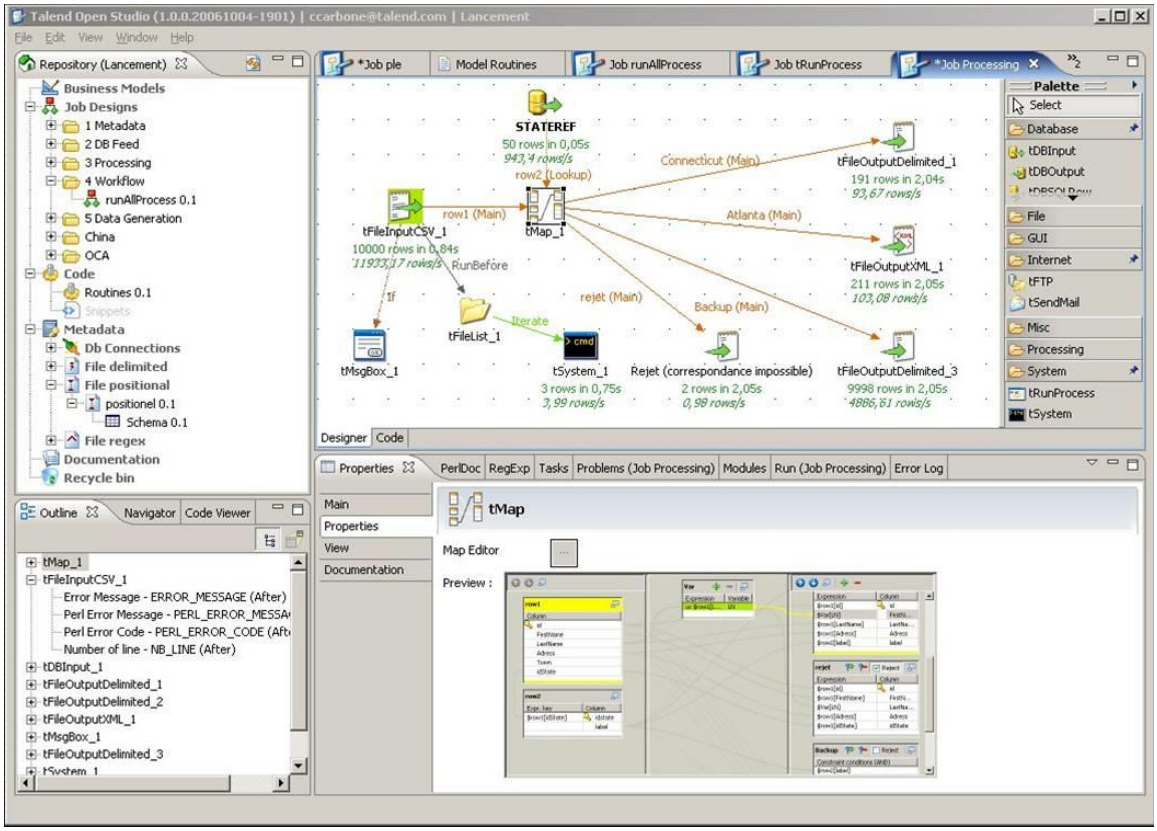

Figure 3.2 Talend Open Studio

3.2.2. Spring Framework

The Spring Framework (Pivotal, 2016) provides a comprehensive programming and configuration model for modern Java-based enterprise applications - on any kind of deployment platform (see Figure 3.3). A key element of Spring is infrastructural support at the application level: Spring focuses on the "plumbing" of enterprise applications, so that teams can focus on application-level business logic, without unnecessary ties to specific deployment environments.

The technology that **Spring** is most identified with is the Dependency Injection (DI) flavor of Inversion of Control. The Inversion of Control (IoC) is a general concept, and it can be expressed in many different ways and Dependency Injection is merely one concrete example of Inversion of Control. Spring Integration supports pipe-and-filter based architectures.

Spring Integration is a framework for Enterprise application integration that provides reusable functions that are essential in messaging, or event-driven architectures:

- routers - routes a message to a message channel based on conditions
- transformers - converts/transforms/changes the message payload and creates a new message with transformed payload
- adapters - to integrate with other technologies and systems (HTTP, AMQP, JMS, XMPP, SMTP, IMAP, FTP (as well as FTPS/SFTP), file systems, etc)
- filters- filters message based on criteria; If the criteria are not met, message is dropped
- service activators - invoke an operation on a service object
- management and auditing

Spring allows the implementation of most of the Enterprise Integration Patterns, including:

- Endpoint
- Channel (Point-to-point and Publish/Subscribe)
- Aggregator
- Filter
- Transformer
- Control Bus
- Dependency Injection
- Aspect-Oriented Programming including Spring's declarative transaction management
- Spring MVC web application and RESTful web service framework
- Foundational support for JDBC, JPA, JMS
- Resource management - automatically acquiring and releasing database resources
- Exception handling - translating data access related exception to a Spring data access hierarchy
- Transaction participation - transparent participation in ongoing transactions
- Resource unwrapping - retrieving database objects from connection pool wrappers
- Abstraction for BLOB and CLOB handling

It easily integrates with a number of External Systems, thanks to the support of the following protocols and technologies:

- ReST/HTTP
- FTP/SFTP
- Twitter
- Web Services (SOAP and ReST)
- TCP/UDP
- JMS
- RabbitMQ
- Email

The framework has also extensive JMX support for easy monitoring of application components. This includes the following:

- Exposing framework components as MBeans
- Adapters to obtain attributes from MBeans, invoke operations, send/receive notifications

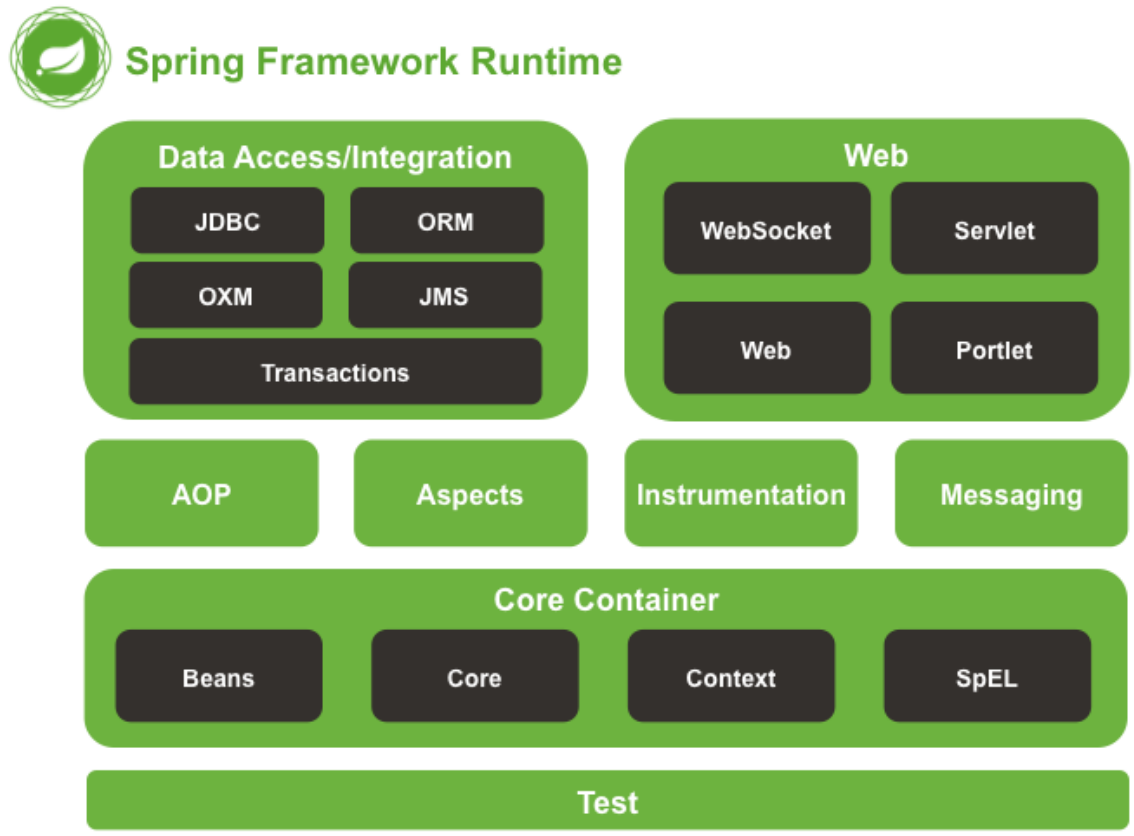

Figure 3.3 The Spring Framework architecture

3.2.3. Apache Struts 2

Apache Struts 2 (Apache Foundation, 2016) is another open source framework for developing Java EE web applications (see Figure 3.4). It uses and extends the Java Servlet API to encourage developers to adopt a model–view–controller (MVC) architecture. The WebWork framework spun off from Apache Struts aiming to offer enhancements and refinements while retaining the same general architecture of the original Struts framework.

The framework is designed to streamline the full development cycle, from building, to deploying, to maintaining applications over time. Apache Struts 2 was originally known as WebWork 2.

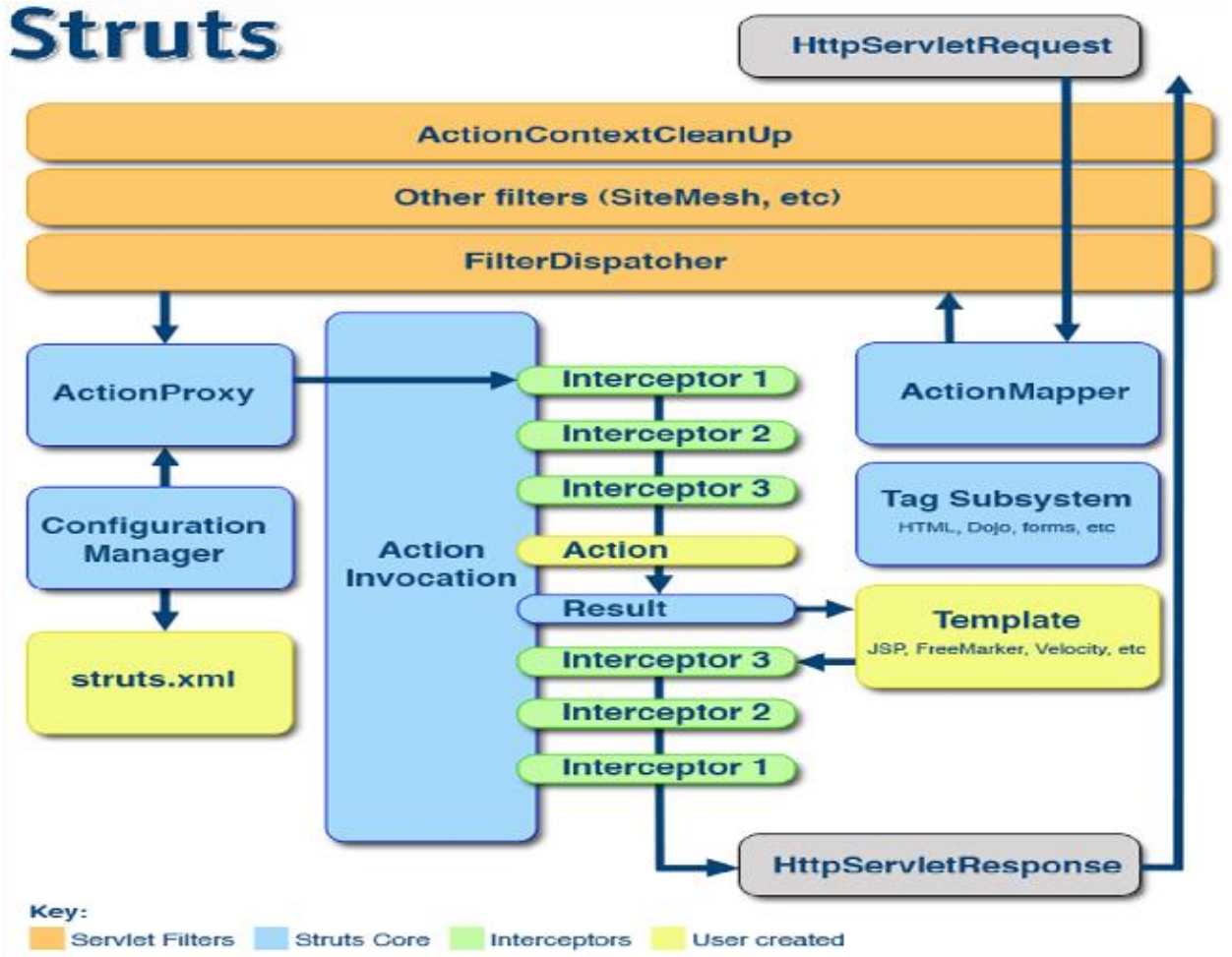

Figure 3.4 Apache Struts architecture

The following are key features of the Struts2 framework that influenced us to consider it as a candidate framework for developing cloud services layer of the Fit4Work system:

- **POJO forms and POJO actions** - Struts2 has done away with the Action Forms that were an integral part of the Struts framework. With Struts2, you can use any POJO to receive the form input. Similarly, you can now see any POJO as an Action class.
- **Tag support** - Struts2 has improved the form tags and the new tags allow the developers to write less code.
- **AJAX support** - Struts2 has recognised the take over by Web2.0 technologies, and has integrated AJAX support into the product by creating AJAX tags, that function very similar to the standard Struts2 tags.
- **Easy Integration** - Integration with other frameworks like Spring, Tiles and SiteMesh is now easier with a variety of integration available with Struts2.
- **Template Support -** Support for generating views using templates.

- **Plugin Support** - The core Struts2 behaviour can be enhanced and augmented by the use of plugins. A number of plugins are available for Struts2.
- **Profiling -** Struts2 offers integrated profiling to debug and profile the application. In addition to this, Struts also offers integrated debugging with the help of built in debugging tools.
- **Easy to modify tags -** Tag markups in Struts2 can be tweaked using Freemarker templates. This does not require JSP or java knowledge. Basic HTML, XML and CSS knowledge is enough to modify the tags.
- **Promote less configuration** - Struts2 promotes less configuration with the help of using default values for various settings. You don't have to configure something unless it deviates from the default settings set by Struts2.
- **View Technologies:** - Struts2 has a great support for multiple view options (JSP, Freemarker, Velocity and XSLT)

3.2.4. Scalatra

Scalatra (Scalatra Team, 2016) is a lightweight web framework written in the up-and-coming new language, Scala. It offers extremely fast development and execution speeds for HTTP applications.

Like Apache Struts2, Scalatra is a microframework, a web software development framework which attempts to be as minimal as possible. A full Scalatra application can be written in very few lines of code. Some of the software that use Scalatra are LinkedIn, The Guardian, IGN and Netflix.

Scalatra can replace other web development frameworks for most tasks. It is easy to install, lightweight, and fast. It makes possible to design and build out high-performance web APIs quickly, and it's integrated with special tools to produce functional, and correct API documentation. Scalatra incorporates advanced constructs for event-driven programming, so it is easy to push information into users' browsers—they see constantly updated information without having to refresh the page.

3.3. Criteria for selecting the best framework

From the options mentioned above, one that best suits the functional and nonfunctional requirements should be chosen. The following key points were taken into consideration when the comparison between candidates has been made:

- ease of use;
- flexibility;
- scalability;
- stability.

Below we discuss these aspects one by one for every candidate.

3.3.1. Ease of use

Talend is probably the easiest solution to get accustomed to. It has a simple graphical user interface that allows to implement a complete server by mainly dragging and dropping graphical components in the IDE. Linking the components and setting their parameters is just as easy. Almost no line of code is necessary for creating quite complex systems.

Scalatra is a very lightweight framework and it is possible to write a complex service with very few lines of code. Its main drawback is that it is written in Scala, a fairly new programming language that introduces some concepts, like functional programming, that are quite unfamiliar to most programmers. Although Java could be used to write a service using Scalatra, the framework takes advantage of developing in its native language.

Spring is one of the most widely used frameworks (at least by Java developers). Given that it is created with the principle of Inversion of Control in mind, it is possible to easily develop a system which has its components decoupled from one another. One of the main incentives of using The Spring Framework is that it has a great community that offers support on any possible issue. One of the issues addressed by developers using Spring is that it is very large, with a great number of classes and functions, most of them not useful for a regular application.

Struts2 is similar to Spring and has the same advantages and disadvantages.

3.3.2. Flexibility

Given that it is a visual programming solution, Talend offers a limited set of components to choose from. A programmer could add snippets of Java code and even create custom components, but it is usually harder to implement these custom features.

All of the other solutions are approximately equally flexible, as they all offer support to most Java libraries available online. Also, creating custom code is mostly on the same level, with Scalatra differing slightly by writing shorter code that is harder to debug.

3.3.3. Scalability

Since version 5.5, Talend offers great scalability by using Hadoop, one of the most widely used frameworks for cloud computing. Talend created a wrapper that enables the developers to create distributed software easier than using Hadoop directly.

Being a lightweight framework, Scalatra doesn't directly offer support for scalable software, but the use of Akka is recommended. Akka is an open-source runtime and toolkit that allows the developers to build concurrent and distributed applications, by using the actor-based concurrency model. It is written in Scala, but it can be used in both Java and Scala based applications. Akka is also used by a large number of companies for cloud computing.

Spring offers its own solution for distributed software, but it is possible to use Hadoop, Akka or another library if it is preffered.

Struts2 doesn't offer a default method of creating scalable software, but as in Spring, third-party options can be used.

3.3.4. Stability

Debugging in Talend is harder than the other solutions, so a stability problem is more difficult to solve.

Scalatra has a few known stability issues that could create problems in particular situations, but these issues are being constantly fixed.

Being one of the most widely used frameworks, Spring is also one of the most stable. Issues are rapidly detected and solved by its community.

Struts2 is also a stable solution, considered even more stable than Spring. It is actually the recommended framework when stability is your main concern.

3.3.5. Final choice

Considering all the aspects mentioned above, Struts2 and Spring are the solutions most suitable to the needs of the Fit4Work system and project. Of these two options Spring has been chosen because of the existing partners' know-how and of positive previous experience with the framework.

3.4. Cloud services overview

In this section, each service or type of services that are part of the Fit4Work Cloud infrastructure are presented and discussed in more detail.

3.4.1. Environment Sensors Service

The environment sensors will collect temperature, humidity, noise and air quality (CO2 content) from an interior, but also from outdoors. The selected sensors will be described in detail in the appropriate deliverable.

The sensors do not provide an API which could allow to directly connect the sensor and the Cloud Integration Layer. The data is stored in a proprietary Cloud infrastructure and from this infrastructure it is possible to gain access to the data. The approach is to build a service which will pull the data from the proprietary Cloud and push it to the Cloud Integration Layer.

A decision will be taken whether this data will be pushed live, or at a specific time interval. The second solution will be chosen, most likely, as this requires less battery from the sensors and less computational power.

No data will flow from the Cloud Integration Layer to the proprietary Cloud of the environment sensors.

3.4.2. Mobile Application

The mobile application, the primary gateway, out of the two that exist in Fit4Work, will rely heavily on a set services which will provide the necessary data in order for it to function. These services will be bidirectional, as the application needs to send information to the recommenders via the Cloud Integration Layer, but also to receive data.

3.4.3. Recommenders Services

The recommenders will have dedicated services that will communicate with the Cloud Integration Layer and with the Data Storage Layer. Based on the data received from the sensors, the recommenders will calculate and reply with personalized recommendations which will be pushed to the user gateways, also via the Cloud Integration Layer.

All recommender services will be elaborated in collaboration with the partners developing the respective recommender, in order to fulfil the needed requirements.

3.4.4. Data storage Services

These services will be used to communicate with a database which will store information coming from the recommenders. The database will also be used to store historical data.

3.4.5. Authorization and Social Network Services

The user will authenticate from both the mobile application and the desktop application, therefore services will be needed to support this functionality: login, logout, profile related services. Integration with other authentication frameworks is also being discussed and, in this situation, services will be used for this functionality as well.

For the Social Network Services, the user will be able to post his achievements to social networks and for this, appropriate services will be implemented, extending a potential authentication framework implementation.

3.4.6. Desktop Application

The Desktop Application will need to communicate with the Cloud Integration Layer in order to be in sync with data collected from the sensors and to also send specific updates from actions that may be performed by the user from the Desktop Application.

The Desktop Application will provide notifications which will assist the user in monitoring working conditions and the several types of stress without having to check his smartphone while at work.

The Desktop Application will extend the functionalities, and therefore, services from the mobile application.

3.4.7. AAL Connector

Fit4Work will integrate with other potential AAL Space services through an AAL Connector using a selected AAL middleware framework. It is currently planned to share specific context events related to the Fit4Work user and their environment with other AAL solutions present in the user space (see D5.2. AAL Middleware Specification). The AAL Connector will intercept certain events triggered during normal Fit4Work processes and convert them into a format used by the chosen AAL middleware integration framework.

4. Conclusions

Fit4Work is based on a service-oriented architecture (SOA). This approach provides that several components from the system communicate with each through services. This is usually done over the network, and in most cases this network is the Internet.

The advantages of using a SOA approach are many, below a few are listed:

- **Independence**: services are not platform dependent, therefore several heterogeneous components (from a platform point of view) may be connected without major restrictions;
- **Modularity:** the SOA approach allows modular testing, since services are usually simple and focused on a particular functionality; it also allows work to be done on several components at once, due to the fact that, as long as the inputs and the outputs are well defined, developing the component is possible without having the entire system ready;
- **Scalability:** a service may be migrated to more powerful hardware, if the situation requires it and this process will be transparent for the users of the application;
- **Availability:** similar to the principle explained above, an instance, or several, of the same service may be available in hot stand-by, therefore in a situation in which the initial service fails, downtime is minimal, possibly even no downtime at all;

The Fit4Work architecture is relying on services and on a central unit called the Cloud Integration Layer. Because of this approach, it is possible to connect several modules to the system in a way that requires minimal interactions from developers of those individual modules. The focus on a modular architecture is crucial in order to support easy and fast integration between all partners in the project, but also, to support the development of a possible Fit4Work product, having in mind future commercialization.

An analysis was done in order to select the best candidate for the Cloud Integration Layer out of a list of pre-selected candidates. This is explained in chapters 3.2 and 3.3, together with the outlining selected candidate and the arguments for choosing the best one among them.

The service categories are described, starting from the general architecture overview in deliverable 5.1, and they are supporting the user and system requirements defined.

Regarding the data being exchanged by the services, details are to be found also in deliverable 5.1.

5. Bibliography

Apache Foundation. (2016). Retrieved from https://struts.apache.org

Carjan, C., Mitrea, A., Neagu, C., Lustrek, M., Cvetkovic, B., Franco, O., et al. (2016). *System Architecture Definition.* Fit4Work Project report.

Pivotal. (2016). Retrieved from http://spring.io

Scalatra Team. (2016). Retrieved from http://scalatra.org

Talend. (2016). Retrieved from https://www.talend.com/products/talend-open-studio/