

Assignment 4

Web services and SOA

Johan Holmgren

1. Introduction

In this assignment, you will learn about web services and service oriented architecture. Using a P1353/P1354 or Q1604 camera, you will build a SOA style system consisting of a motion detection logger, a web service that provides motion logs, and a client that is able to request motion logs from the camera (via the web service) and display the motion log in a nice way using an online charting service.

All of the P1353/P1354 and Q1604 used in this assignment have the ACAP application *Axis Video Motion Detection (VMD) 3*¹, which creates motion detection events when motion is detected. In this assignment, we strongly advice that you make use this application in order to detect motion.

During supervised times you are able to physically access the cameras, and during non-supervised times you are able to remotely access two of the cameras (however only from the laboratory used in the assignment): one camera of model type P1353/P1354 and one camera of model type Q1604².

Please note that there is a risk that you disturb each other if you work on the same cameras simultaneously. Therefore, you will have to respect each other when working remotely on the cameras. However, as you are all sitting in the same room, you should be able to synchronize your work by simply talking to each other.

2. Preparation

Read carefully

- The documentation on event management in the AXIS Embedded Development SDK, which you find in your Ubuntu system (`~/Desktop/axis/acap/docs/start_here.html`). In particular, you are recommended to read the text in the document `~/Desktop/axis/acap/docs/html/axevent_8h.html`
- The VAPIX documentation on Axis Video Motion Detection 3, which you can find in the Important resources folder in the Assignments module in the course homepage on Canvas on the Axis partner pages online. You find the documentation on Video motion detection under **Applications** in the VAPIX library documentation.

¹ VMD 3 is preinstalled on all of the cameras used in this assignment.

² You can find the ip addresses of the connected cameras using the following command:
> `sudo nmap -n -sP 192.168.20.0/24`

Please note that much more useful information is available in the example applications included in the ACAP SDK, and in the ACAP and VAPIX documentation.

3. The assignment

As mentioned above, you will in this assignment build a system consisting of three applications:

- An application that logs the motion detected by the camera (below referred to as *Motion detection logger*).
- An application that provides motion history from the motion over HTTP (below referred to as *Motion log web service*).
- A client application that requests a motion history from the web service, and displays the log history in a nice, graphical, way for the user, using Google charts or any other available online service (below referred to as *Client*).

A system architecture, which illustrates how the different applications in your system are connected, is given in Figure 1. Please note that this architecture allows for some flexibility regarding where each of the components should be located. For example, the motion logger could run on the camera or on another computer that listens to motion detection events that are sent out by the camera. Hence the data might be stored either on the camera or externally, and the motion log web service could run either on the camera or on another server.

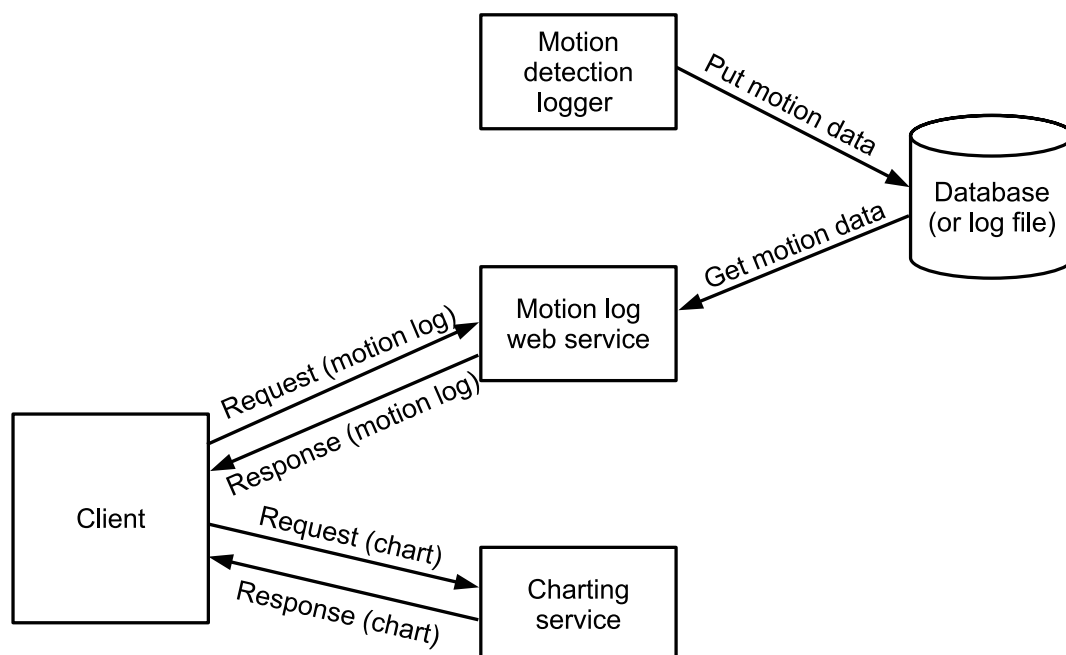


Figure 1: System architecture

Below we specify the requirements for each of the three applications in your system.

Motion detection logger

The purpose of this application, which might be an application that runs on the camera, is to listen to motion detection events that are produced by the application VMD3, and log information about the detected motion somewhere, for example, in the camera's file system or in an external web server.

Hint: In order to subscribe to the motion detection events (produced by VMD3) using ACAP, you first need to call the `ax_event_key_value_set_add_key_values(...)` function in the following way:

```
ax_event_key_value_set_add_key_values(key_value_set, NULL,  
    "topic0", "tns1", "RuleEngine", AX_VALUE_TYPE_STRING,  
    "topic1", "tnsaxis", "VMD3", AX_VALUE_TYPE_STRING,  
    "active", NULL, NULL, AX_VALUE_TYPE_BOOL, NULL);
```

Motion log web service

The purpose of this application is to provide information from the motion detection log using basic HTTP messages. The application should be able to receive an HTTP request that specifies a time interval for which it should provide motion information, and it should respond with a HTTP message that specifies when motion has been detected using a format that you define.

Client application

This application is the interface towards the user. The user should be able specify the time interval for when the client application should display motion history. For the chosen time interval (perhaps an hour) the application should then request motion history from the camera and display the returned information using a chart retrieved from Google Charts or any other online charting tool. For example, you could use a timeline chart. The chart should clearly show when the camera has detected motion as well as when it has not detected motion during the specified time interval.

Hint: As Google Charts supports connections using JavaScript, HTML together with JavaScript might be a good option for building your client application. However, please note that the use of JavaScript puts some limitations on how you build your web service, as JavaScript applications are prevented from making cross domain HTTP requests. A possible workaround is to use a callback where your web service outputs some valid JavaScript code that specifies function call (with the content to process as argument). Furthermore, you need to implement the called function, where you parse the message, in your client application³. Another solution that might work is to make use of Cross-origin resource sharing (CORS)⁴.

4. Examination

Each of the groups should write a short report where you describe your solution. In the report, you should, at least:

³ See, e.g., <https://en.wikipedia.org/wiki/JSONP> for more information on this type of workaround.

⁴ See, e.g., https://en.wikipedia.org/wiki/Cross-origin_resource_sharing

- Briefly describe your system, including both an overall system description and more detailed descriptions of each of your applications, including what technique(s) you used for each of your applications. You need to justify your choices of techniques.
- Discuss what part(s) of the assignment you found particularly interesting and challenging.

In addition, we would appreciate if you provide feedback on the assignment, for example, if there is anything you think should be added, removed or updated.

Your should report should:

- Include a reference list if it refers to external sources.
- Be written in your own words (i.e. slightly modified or simply translated texts written by someone else is not allowed).
- Be well-structured with relevant headings and correct paragraph groupings.
- Be well-written (with correct spelling and sentence construction).⁵

The final report shall be a word (.docx or .doc) or a pdf document. The report and the source code shall be submitted as one zip-file⁶ using the course page on Canvas. The submission deadline is specified on the submission page for this assignment. If you miss the specified deadline (or if your report is not approved after one resubmission), the second and third opportunities to submit will be around the second and third opportunities to write the final exam respectively (these submission deadlines are also specified on Canvas). One submission per group is sufficient (i.e. one report and one set of applications). After the submitted report has been approved you should meet the assignment supervisor for a short discussion on your assignment (the supervisor will announce the time for the discussion). In order to pass this assignment, you need to pass both the report part and the discussion part of the examination.

⁵ Hence, you need to proof-read your report several times before you submit it.

⁶ Please note that all other archiving file formats than zip are forbidden.