# 1 Introduzione

## 1.1 Obiettivo

L’obiettivo del documento è quello di illustrare l’architettura del sistema “Swatcher” fornendo una descrizione dettagliata delle componenti dalle quali è composto e le loro modalità di interconnessione.

## 1.2 Scopo del Sistema

Swatcher è un sistema di video-sorveglianza remoto, accessibile tramite un’applicazione mobile Android. Il sistema è composto da una componente client rappresentata da un’applicazione Android, e da una componente server rappresentata da una videocamera ad accesso remoto. Il sistema consente all’utente un’interazione in tempo reale costituita dalla visualizzazione dello streaming video, dalla cattura di foto e registrazione di video. Il sistema, inoltre, è in grado di rilevare movimenti nell’ambiente sorvegliato ed avvertire l’utente dell’evento rilevato mediante notifica alla componente client.

## 1.3 Definizioni Acronimi Abbreviazioni

| Termine | Definizione |
| --- | --- |
| streaming |  |
| snapshot |  |
| gallery | video + img |
| media |  |
| schermata gallery |  |
| AWS | Apache Web Server integrato all’interno di Raspberry-Pi |
| FS | File System di archiviazione contenuti lato server |
| MWS | Motion WebCam Server. Sottosistema lato server che si occupa di gestire i comandi provenienti dall’App Client. Contiene al suo interno il modulo EVM. |
| EVM | Event Manager. Modulo presente all’interno del sottosistema MWS. L’EVM è in grado di intercettare gli eventi provenienti dalla WebCam e di gestirli inviando una notifica al NOM o salvando su disco i media prodotti a seguito dell’evento intercettato(SNapshot e/o Video). |
| NOM | Notification Manager |

# 2 Architettura del Sistema

## 2.1 Componenti Principali del Sistema

Il sistema Swatcher è composto da 2 sottosistemi interagenti tra di loro:

1. Un sottosistema denotato come “Client” costituito da uno Smartphone dotato di sistema operativo Android.
2. Un sottosistema denotato come “Server” costituito da un single-board computer Raspberry Pi.

I 2 sottosistemi appena elencati sono stati definiti come Client e Server anche se il funzionamento complessivo del sistema non sempre rispetta un modello Client/Server puro: esistono determinate condizioni in cui la componente Raspberry-Pi si comporta come un Peer interagente con ulteriori Sottosistemi/moduli(Notification Manager). Lo scenario appena descritto si manifesta al verificarsi di un evento che viene intercettato e gestito autonomamente dall’Event Manager, un modulo o sottosistema appartenente alla componente ‘Server’. In tutti gli altri scenari di funzionamento il sottosistema Raspberry-Pi si comporta effettivamente come server dell’Applicazione Android residente sullo Smartphone.

La fruizione dei servizi resi disponibili dalla piattaforma server è possibile soltanto attraverso dispositivi dotati di sistema operativo Android; non è stato previsto uno sviluppo del sistema per poter interagire anche con dispositivi dotati di sistema operativo iOS.

## 2.2 Elenco completo delle componenti

In questo paragrafo vengono elencate in forma tabellare tutte le componenti del sistema riportandone nome e/o acronimo col quale ci si riferirà nel resto del documento.

## 2.2 Struttura e Collegamenti delle componenti del sistema

Nell’immagine seguente viene riportata una prima descrizione dell’architettura complessiva del sistema.

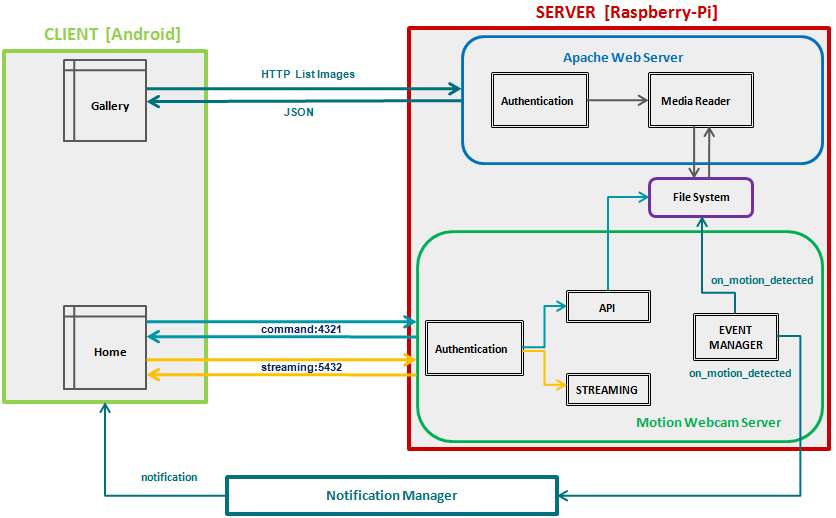


Figura : Una prima vista Architetturale del Sistema Swatcher

In questa prima overview Architetturale il Sistema Swatcher è stato composto nelle sue componenti principali. Oltre ai già citati moduli client e server notiamo la presenza di una terza componente denotata come “Notification Manager”. Struttura e modalità di interazione del Notification Manager verranno descritte successivamente.

### 2.2.1 Sottosistema Client

Il sottosistema Client è costituito da un app. deployata su un qualsiasi smartphone Android. Nella prima Overview del sistema (Figura 1) sono state esplicitate le componenti software “Home” e “Gallery” che sono due videate fondamentali dell’app. Swatcher quindi usate per comprendere le principali interazioni e modalità di comunicazione che il sottosistema client intraprende con la restante parte del sistema (Server).

La videata o Android Activity “Gallery” permette di richiedere al server l’intero listato dei file multimediali che sono stati archiviati nel file system della componente server; ciò avviene attraverso una richiesta http inoltrata dall’app al sottosistema Server. A seguito di tale richiesta l’anteprima di dettaglio dell’intera collezione presente in remoto viene passata all’app in formato JSON.

La videata o Android Activity “Home” permette l’esecuzione di due funzionalità principali:

1. La visualizzazione dello streaming proveniente dalla WebCam
2. La possibilità di scattare istantanee (Snapshot) memorizzandole opportunamente in remoto lato server.

Entrambe le funzionalità sopra elencate sono realizzate mediante l’opportuno scambio di comandi tra l’app mobile e l'MWS.