

AN EFFICIENT SOFTWARE ARCHITECTURE FOR VIDEO MANAGEMENT SYSTEM

Mitesh Patel

M. Tech in Embedded and VLSI Systems.
U. V. Patel College of Engineering, Kherva, Gujarat. India.

ABSTRACT

Video Surveillance System is rapidly growing domain as organizations demands to safe guard physical and capital assets.^[1] As the database increases and number of features increases in video management systems the whole system's efficiency decreases. Video management systems contains many modules like live, playback, carousel etc. This paper addresses software architecture specially designed for video management system.

I. INTRODUCTION

Algorithms and data structures constitute the major design problems when sizes of video surveillance software increases.^[2] When Video Management systems constitutes many components the organization of the overall system leads into major design problems. The overall design of video management software can be presented in number of ways including descriptive terms, informal diagrams, modules interconnection languages, framework and templates for systems that serve the demands of particular domains and models of component.

II. DESIGN FLOW

Design flow of Video Surveillance System software encompasses various stages. From GUI to database, data travels to and from database to GUI. Data mainly travels in two forms, in structures and JSON form.

From GUI data is filled in Structures and is passed to controller section to call for specific API. Controller calls common communication framework where structure is serialized into light-weight data form JSON.

After that RESTAPI is called with requested JSON and particular API command and URL. RESTAPI with the help of CURL library calls CGI and gives JSON as a command line argument. Rest CGI calls Client to connect with Server using TCP socket. As server receives data and API command it transfers to Service that is running continuously and listens to service if any data is coming then it gives relative response and acts on data.

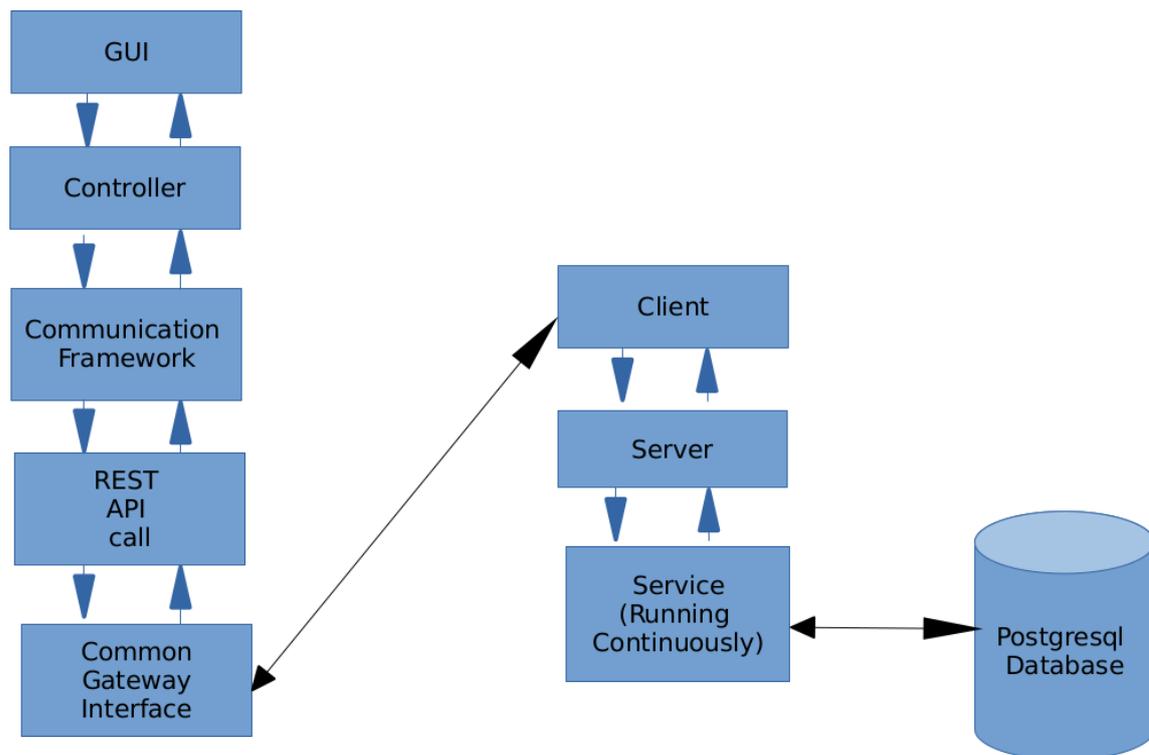


Figure 1. Architecture

III. QT BASED GRAPHICAL USER INTERFACE

Qt is framework designed in C++. Framework includes extensive C++ libraries which helps engineers to develop and build highly performing application and GUI for several desktop and mobile OS. [3] GUI can be made with any of the framework available in the markets. But the reason we go with the Qt is, it is faster and supports many embedded devices. Qt is framework written in C++. Qt supports Model/View architecture to handle data flow. Models represents classes that stores data. Data can also be retrieved from model as and when required. Views are readymade classes in Qt similar to widgets that helps to view data to user. Types of models and views can be varies depending upon how one wants to represent data. It may be list-view, table-view, or can be a simple combo-box. Example for Model of List-view is set to object of list view class. As and when data is changed in Model view shows those changes to user in display. Model is connected to external world of Qt to Controller's API which returns data. Models also passes data as an arguments to API to controller.

IV. CONTROLLER

Controller connects GUI to external world. As and when GUI needs to get data, update, get list or delete some data, it calls respective controller functions which in turn calls Send and Receive function which sends data in structure format and receives data in structure format. Controller returns status of whether API is successful or not. With structured format data, controller also sends API command for specific API.

V. COMMUNICATION FRAMEWORK

Communication framework constitutes main block of whole video management software architecture system. Communication Framework contains definition of Send and Receive function. Send and Receive function serializes data. The structured format data sent by GUI is converted into JSON (Java Script Object Notation). JSON is light weight compared to structure format. Hence it can be easily passed to RESTAPI and over network. URL (Universal Resource Allocator) is already defined in

REFERENCES

- [1]. C. Lakshmi Devsena, R Revathi, M. Hemlatha. "Video Surveillance Systems- A Survey". IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 4, No 1, July 2011 ISSN (Online): 1694-0814.
- [2]. Advances in Software Engineering And Knowledge Engineering, Vincenzo Ambriola, Genoveffa Tortora. ISBN 981-02-1594-0
- [3]. Parul, Jagandeep Sidhu. "Study and Design and Development of Integrated Development Environment using Qt C++ Framework". International Journal of Advanced Research in Computer Science and Software Engineering. Volume 4, Issue 11, November 2014. ISSN 2277 128X
- [4]. Min Choi, Young-Sik Jeon and Jon Hyuk Park. "Improving Performance through REST Open API Grouping for wireless Sensor Network". International Journal of Distributed Sensor Networks. Volume 2013, Article ID 958241.
- [5]. Kamal Kathuria, Chaynika Kapoor, Apoorva Adlakha. "Common Gateway Interface". International Journal of Science and Research. Volume 3 Issue 10.
- [6]. Limi Kalita. "Socket Programming". International Journal of Computer Science and Information Technologies. Vol 5(3) ,2014. ISSN:0975-9646.
- [7]. <http://doc.qt.io/qt-5/qtcpsocket.html>. Qt Official Site.