Architectural styles

Introduction

Our system embraces diverse architectures, incorporating data flow, data-centered, and resource-oriented architectures. The application follows a 3-tier structure and incorporates a notification architecture.

Data flow architecture

Within our database, the data flow architecture is evident, utilizing pipe and filter functionality to refine .osm data. Structural variation of the pipe and filter exists due to multiple inputs, resulting in loose coupling facilitated by independent filters. The ultimate output is on the client side.

Data centered architecture

Communication between the centralized data store and clients aligns with the repository subtype, featuring a widely used centralized data format - .csv in a database.

Resource-Oriented Architecture

Data manipulation occurs through the HTTP protocol, aligning with the principles of Resource-Oriented Architecture.

Virtual Machines

Our system is powered by the Java virtual machine.

Interpreter

Command scripts and JavaScript serve as interpreters within our system.

3-tier architecture

The presence of business logic functionalities in the Java backend contributes to our adherence to a 3-tier architecture.

Notification architecture

Events are dispatched to interested components within our notification architecture.

Client-server

The client-server relationship is characterized by loose coupling, with vertical scaling implemented.

MVC(Model-View-Controller)

The Springboot framework is employed, where the model interacts with the data layer, adhering to the MVC pattern.

Heterogeneous architectures

The inclusion of a search engine contributes to the overall heterogeneity of our system.