

Дефинирање на архитектурни стилови во апликацијата

Апликацијата се состои од **хибридна архитектура**, која ја сочинуваат архитектура за проток на податоци, централизирана архитектура, клиент-сервер архитектура, слоевата архитектура, микропроцесорски системи, GUI како MVC, ресурс ориентиран стил (користејќи rest controller) и SaaS (за cloud делот).

За почеток започнавме со **архитектурата за проток на податоци**, конкретно концептот со цевки и филтри. Како влез зедовме податоци од OpenStreetMap. Податоците ги изпроцесиравме преку концептот на цевка и филтер и ги зачувавме во крајна дестинација .csv документ што всушност претставува наша база. Базата која се состои од филтрирани објекти од интерес на апликацијата е пример за **централизирана архитектура** на податоци кои го осигуруваат интегритет на податоците достапни за корисниците. Комуникацијата се одвива преку интернет.Shell скриптата како интерпретер кој работи на ниво на оперативен систем е претставник на слоевитата архитектура. Го искористивме како подлога за филтрирање на податоците што помогна во флексибилноста на кодот и негова реискорисливост со цел филтрирање на други целни групи и објекти. Комуникацијата се одвива преку ХТТП протоколот исто така слоевата архитектура.

Клиент – сервер архитектура

Апликацијата е заснована на добро познатата парадигма на клиент / сервер кои престојуваат претежно во сервер за апликации и тоа се пристапува од далечински клиент со тенка мрежа. Комуникацискиот протокол помеѓу серверот и клиентот е протоколот за пренос (HTTP), кој од страната на серверот се анализира и обработува од компонента на серверот за апликации кој е попознат како веб-контејнер. Клиентот е одвоен од веб апликацијата т.е зборуваме за loose coupling. Функционалноста е јасно поделена,

SaaS (Cloud)

Апликацијата е хостирана на CLOUD, што значи дека серверот не е во организациската мрежа туку е некаде на интернет. Клиентот пристапува далечински кон апликацијата.

Микро-сервиси

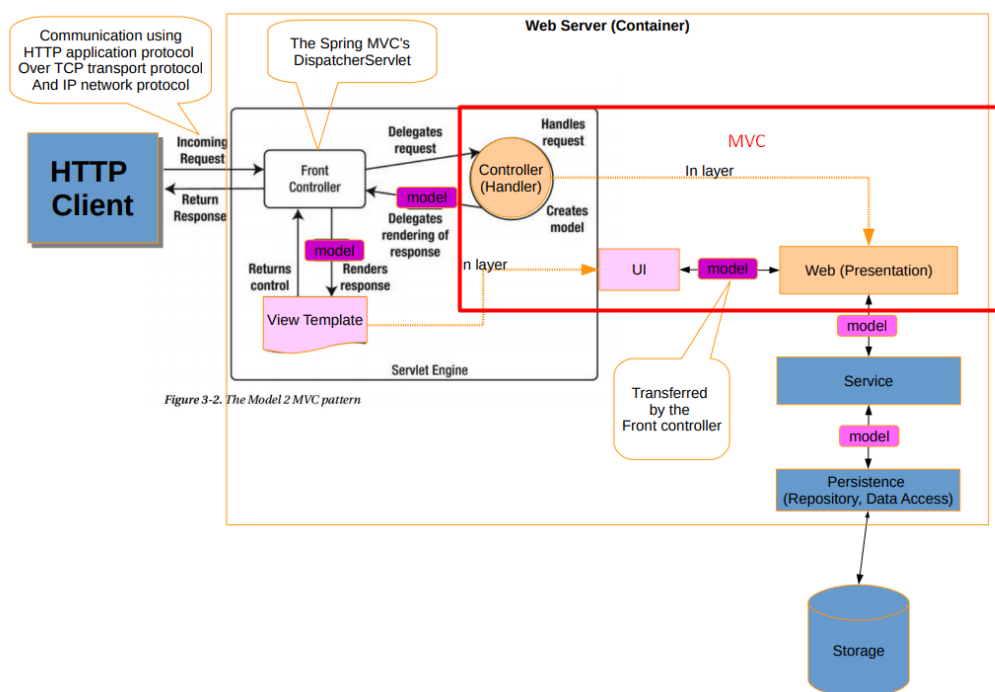
Микро-сервисите се Loosely coupled архитектура кои секоја засебно извршува некоја задача, како што се функцијата за наоѓање бар според име и функцијата за наоѓање бар според растојание.

Слоевита архитектура& GUI како MVC

Апликацијата е организирана во слоеви. Соодветно секој слој си извршува своја засебна функција. Апликацијата е составена од презентациски, сервис, репозиторен и дата слој кој што соодветно презентацискиот е одговорен за основната комуникација со клиентот,

сервисниот ја извршува бизнис логиката во апликацијата, репозиторниот слој е задолжен за преземање податоци од базата кои што се потребни за погорните слоеви и препраќање кон нив соодветно и дата слојот е задолжен за чување на состојбата на апликацијата т.е. тоа претставува дата базата. Презентацискиот слој е организиран со MVC шаблонот каде што е поделен на model, view, controller соодветно, контролерот ги прима барањата креира соодветен модел и го враќа како одговор до front-end апликацијата(користејќи rest контролери).

На сликата подолу е прикажана интеграцијата на MVC во слоевитата архитектура, приказот е на пониско ниво и прикажува интеграција на MVC во Spring слоевитата апликација (презентацискиот слој организиран како MVC). Барањата најпрво пристигнуваат до dispatcher controller кој користејќи го handler mapping одбира кој контролер ќе се справи со барањето и го препраќа барањето до него кој потоа се справува со барањето и генерира одговор.



Spring MVC [1]

Користена литература:

[1]- Craig Walls - Spring in Action, 5th Edition (2018, Manning Publications)