

University of Pretoria
Software Engineering - COS 301

Architectural Design

Contrapositives
May 2019

Authors:

| | |
|------------------|------------------|
| Brendan Bath | u16023359 |
| Musa Mathe | u15048030 |
| Jessica da Silva | u16045816 |
| Natasha Draper | u16081758 |

Architectural design

Contrapositives

May 2019

1 3-tier + Component-based + Plug-in Architecture

1.1 3-tier

1.1.1 Presentation layer

- This will be the top level of the application which will provide graphic user interface by displaying information to the user in a more organized and simple manner in which the user can easily understand and navigate through application. This layer will serve as a means to communicate with the user by displaying list of services, performing calculations, and displaying the metrics of the instances.

1.1.2 Logic layer

1. This layer will serve as a middle dynamic content processing which will be responsible for coordinating the application, and making logic decision and performing calculations. We will be using python for the logic and flask to expose the logic as a restful api.

1.2 Data layer

1. The data layer will provide an API to the application layer that exposes methods of managing the stored data like keys and login details without exposing or creating dependencies on the data storage mechanisms.

1.3 Component-based

1. This will provide flexibility which will ensure separation of concerns which will help for defining, implementing and composing loosely coupled independent components into our system through out the development cycle to allow the client benefit both for short and long-term process since it will provide us continuous deployment of features since a component(e.g displaying metrics) will be developed as a separate feature on it's own then later will be converted to a service which then will be added to the running application. It also give us an opportunity to substitute a component which needs to be replaced by another component with either an updated version or an alternative without breaking the system in which the component operates.

1.4 Plug-in

1. This enable us to add features and functionality to our application by just installing a new bundle by adding multiple instances of a particular type of module that provides a well-defined unit of functionality to our application thus allowing for customization. Since we need to pull data from other third-parties services this will give us an ability to implement plug-in functionality using some form of shared libraries, which we can get data dynamically at run time thus extending our application e.g pulling data via a plugin using boto3. This will also reduce the size of our application.

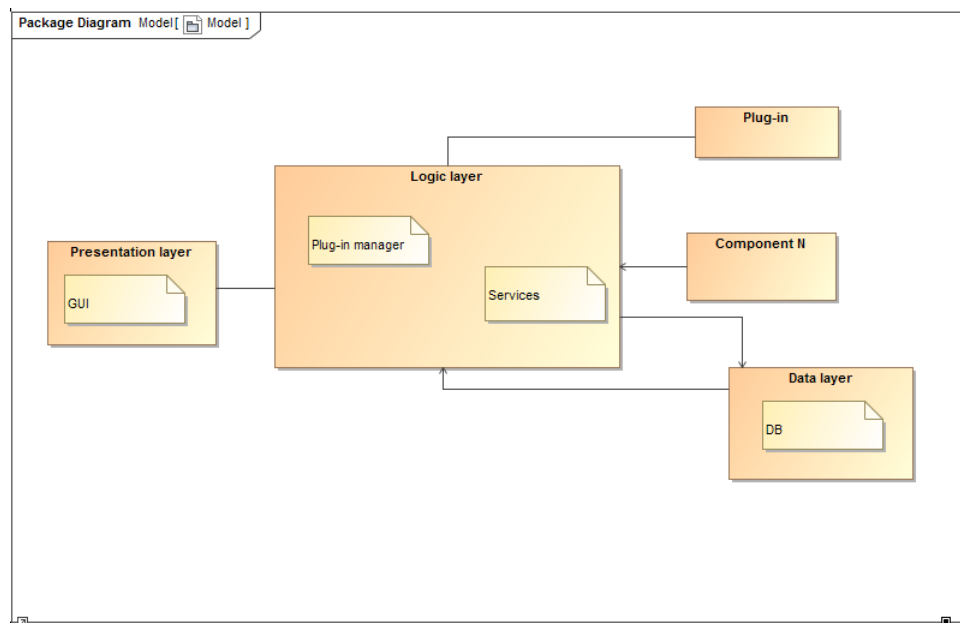


Figure 1: Amazon Dash Architecture consideration