Department of Computer Science

COS 301 - Software Engineering

Not Like This

Architectural Specifications

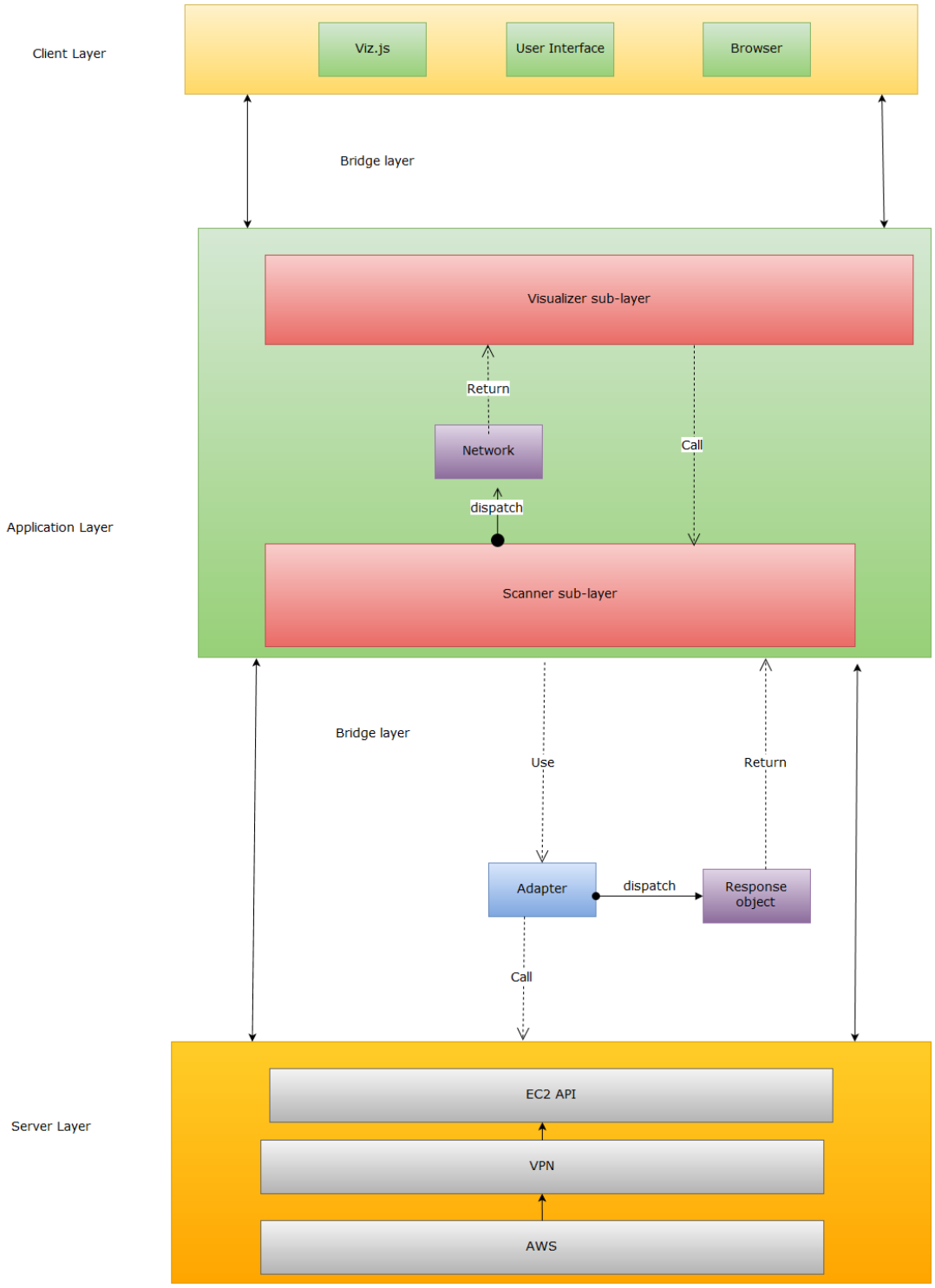
Authors:	Student number:
Jedd Shneier	13133064
Duncan Smallwood	13027205
Daniel King	13307607
Muller Potgieter	12003672

28 July 2016

Architecture Requirements

1.1 Architectural Scope

The project is implemented using a layered architecture. The main 3 layers being the client layer, application layer and the server layer. As seen in the below diagram.



1.2 Quality requirements

- Maintainability

  The code needs to be well documented and designed such that after the project has been completed it is easily maintained and possibly extended by the Amazon development team.

- Scalability

  This is one of the major quality requirements for the service. As stated from the start of the project. The system will have to work with very large networks in future and thus the algorithms within will need to scale well such that they can work with these large networks and so that performance costs are minimized when these large networks are used.

- Usability

  The system is going to be used by the Amazon clients who will not necessarily have programming knowledge thus the end product needs to be catered to these users such that they are able to use the program efficiently.

- Performance

  Performance is another major quality requirement that much of our attention needs to be focused on. All visualizations need to be rendered within 5 seconds. This was a requirement set by the client.

- Security

  As we are working with sensitive client info (login/account information), it is imperative that we make sure the system is secure and does not allow for this information to go to unwanted 3rd parties.

1.3 Access and integration channels

- The information regarding the networks is taken via the Amazon API services.
- The user should login via the existing Amazon site and the visualizer should integrate into the site itself.
- The scanner and visualization elements of the product are facilitated by an adapter class which handles the communication between these two objects.

1.4 Architectural constraints

- The user must have an existing and valid account with the Amazon services.
- The user must be using a virtual network

2. Architectural Patterns or styles

2.1 Layer (object-oriented design)

- Client layer – This handles the user interface as well as the visualization aspect of the program.
- Application layer – This layer handles the majority of the generation of the visualization. As well as the scanning of the actual network.
- Server layer – This is the Amazon server side API and will not be modified however it is used extensively by the higher layers.

3. Use of reference architectures and frameworks

- vis.js – This library is used extensively to create the visualization of the network, through the use of node and link generation.
- Amazon API – This is used to create the network object (the representation of the actual network)

4. Access and integration channels

- The product needs to integrate with the Amazon API as well as the actual website Amazon intends to use with the visualizer.
- The system will be accessed by this site and all verification is done on prebuilt user account systems.
- vis.js will need to be integrated within the HTML CSS and other markup for display on the site.

5. Technologies

- Java : used to integrate with the API, and to create the visualization
- CSS, HTML, Javascript : To create the site elements displayed by the browser
- vis.js : handles the generation of the visualizer