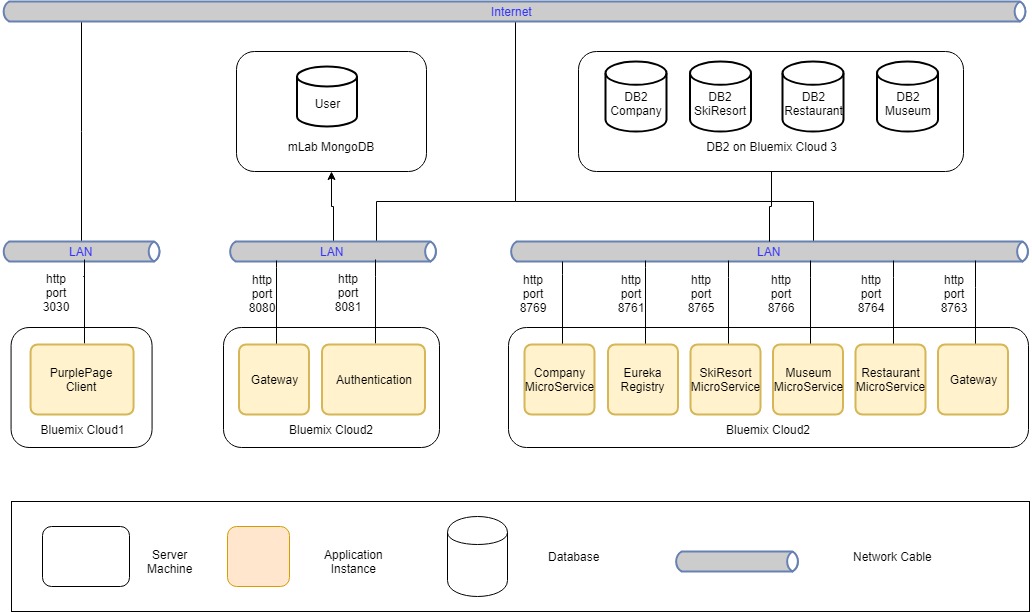
# Allocation View

## Deployment view of a system

### Section 1: The Primary Presentation



### Section 2: The Element Catalog

Totally, there are three could instance deployed on bluemix in our project.

PurplePage Client is the frontend of our project, it is responsible to interact with users and deployed on Bluemix Cloud1.

Both Bluemix Could2 and Cloud3 are in charge of some microservices. For the Cloud2, there are Gateway, Authentication and Company Microservice which are implemented by NodeJS. They all connect to the mLab MongoDB that has two tables, user table and companies table. Authentication service shares the same database with login and signup functions. For this part, they have already deployed on bluemix but we design it properly so that it can be deployed in any system, environment, cloud platform.

The last part, including Microservice registry, request gateway, three microservices are all deployed on Bluemix Cloud3. All the Eureka clients consisting of three microservices and gateway will send the heartbeat to notify the action of registering and in Eureka server will check whether the service is still alive periodically.

### Section 3: Context Diagram

Totally the same as Section 1.

### Section 4: Variability Guide

In the Registry Center, we should config the eureka server host and port number. Furthermore, we need to set some properties to manage the usability of both server and clients. Besides, the source code needs to config the database connection in the main *application.yml.* Therefore, the DAO layer can retrieve Eureka data from a database located in different servers or hosts.

As mentioned before, we decide to deploy the applications on bluemix, so that we need to utilize pipeline in bluemix to integrate the compile and deploy procedures. Some parameters like service bound name, runtime memory and service start timing should be set properly according to the account’s authority.

### Section 5: Rationale

The project the frontend part to interact with users and the NodeJS is an effective language to implement this functionality. Moreover, in order to imitate the actual development environment and requirements, we decide to utilize two server language to implement all the logic layers. Therefore, we try to use bluemix to implement two different kinds of microservice to integrate the project. Moreover, microservice, in terms of a kind of SOA architecture, need a registry to manage all the microservices. Eureka has set the connections between the server and clients by allowing the client to send a heartbeat to the server and requiring the server to check the status of clients lifecycle in an exact time interval. All three parts are deployed on Bluemix and are able to be distributed with domains and accessed by users anytime and any platforms.