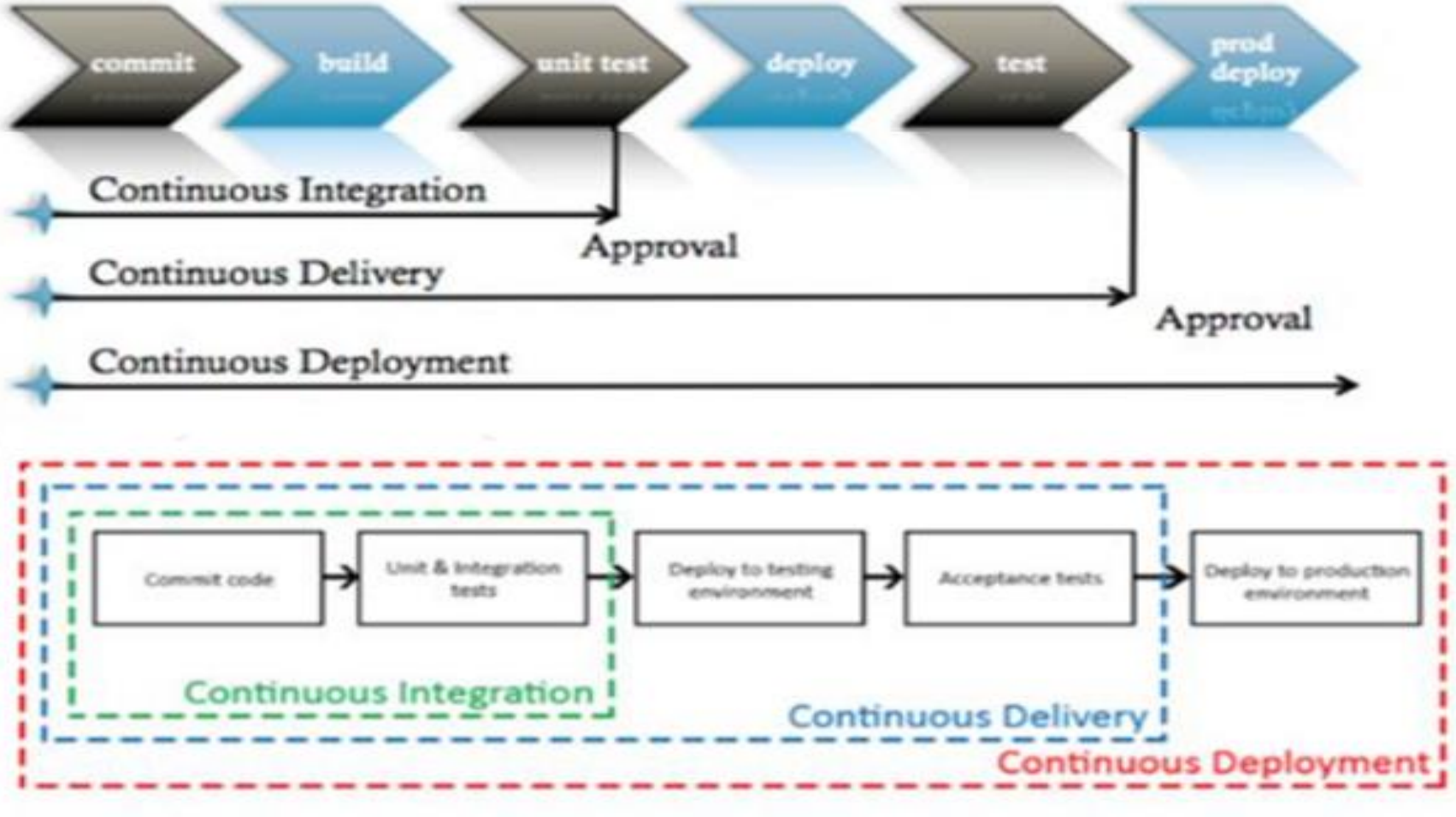
Capstone Project

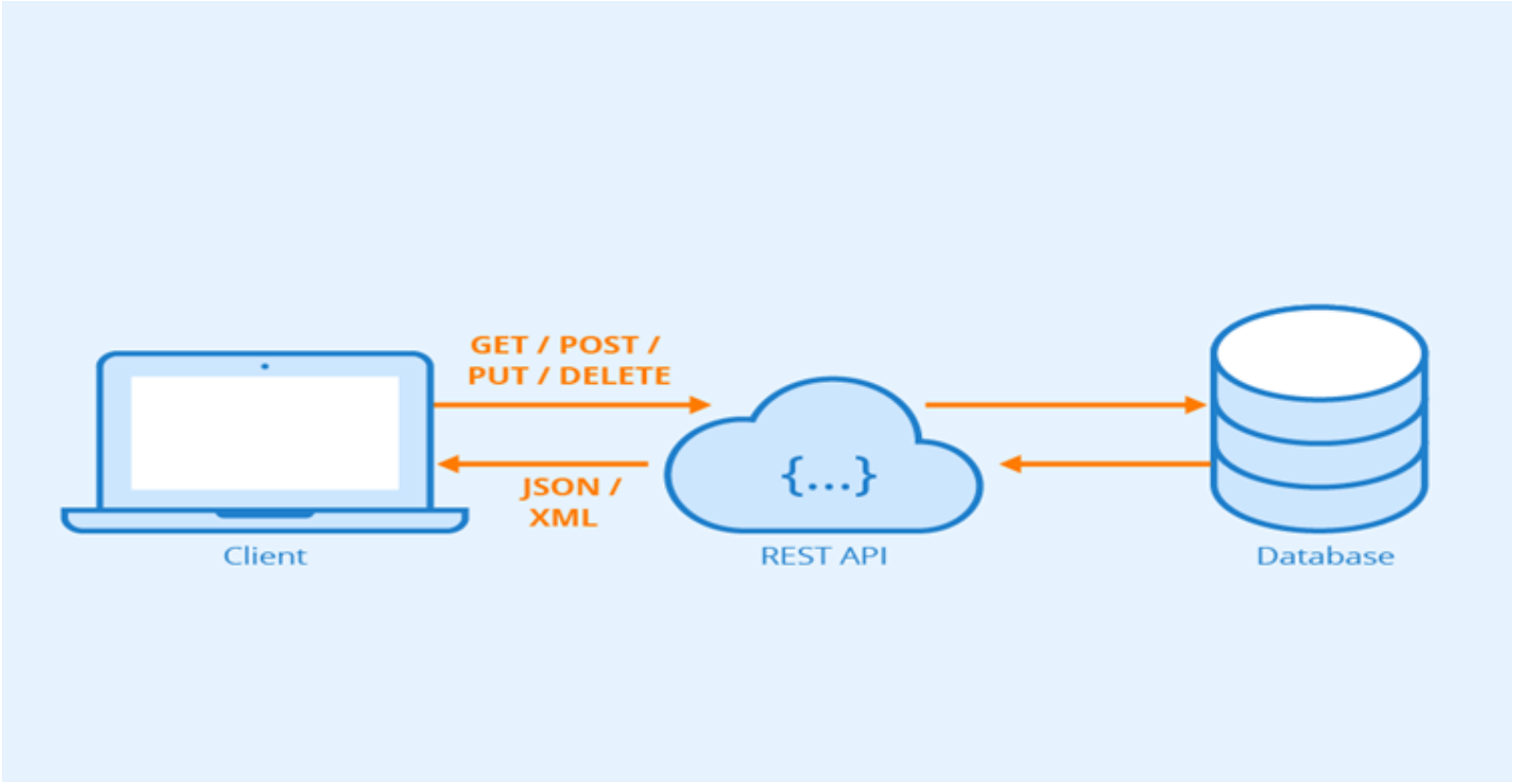
GlobeHopper App

CI/CD


commit
build
unit test
deploy
test
prod deploy
Continuous Integration
Approval
Continuous Delivery
Approval
Continuous Deployment
Commit code
Unit & Integration tests
Deploy to testing environment
Acceptance tests
Deploy to production environment
Continuous Integration
Continuous Delivery
Continuous Deployment

CI/CD Implementation

API – Application Programming Interface



Python - Flask

- Flask is a lightweight WSGI (Web Server Gateway Interface) framework that contains tools and libraries used to develop a web application in a fast and efficient way.

- **Install:** ***pip install Flask***

- Welcome to Flask — Flask Documentation (2.1.x) (palletsprojects.com)

Database - MySQL

- MySQL is a widely used relational database management system (RDBMS). It is free and open-source. It is ideal for both small and large applications.

- Install: MySQL :: Download MySQL Installer

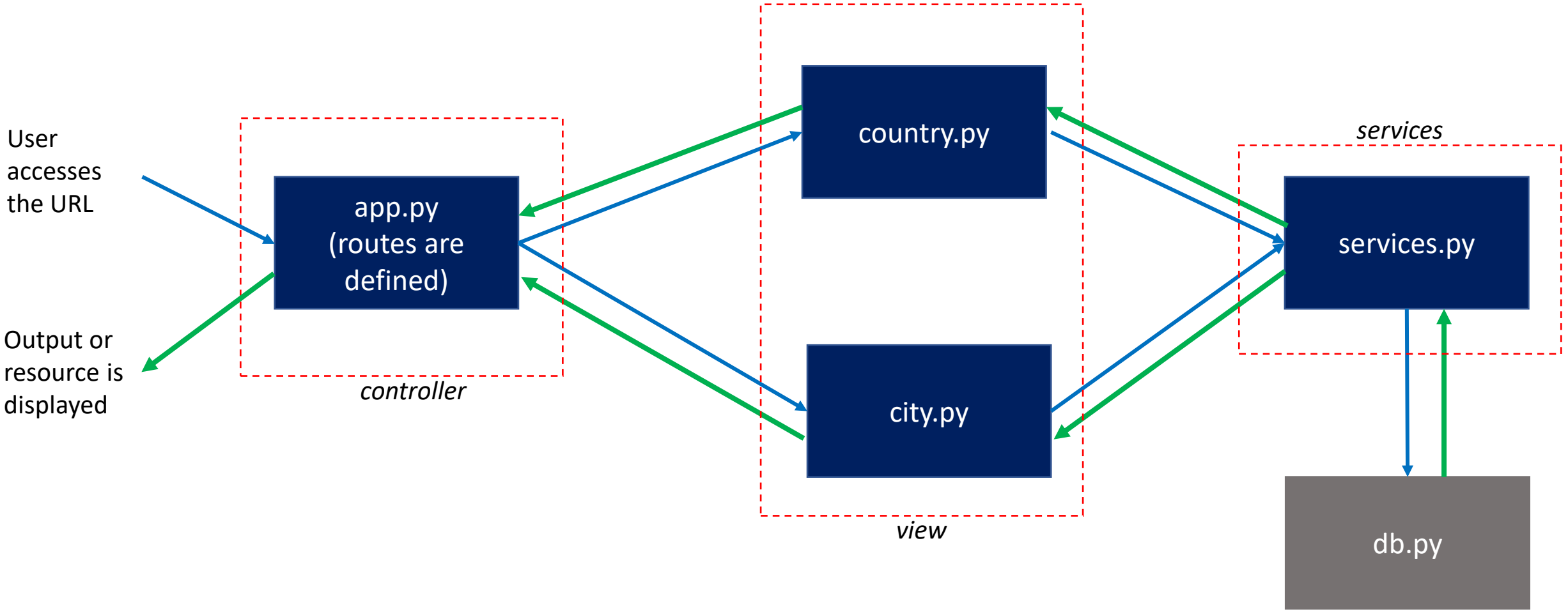
- Create a new user with valid credentials on the localhost for the application

- **Install: *python -m pip install mysql-connector-python***

API Endpoints

- Ability to search for all countries of the world as a Traveler - **/countries**
- Ability to search for all countries in a given continent as a Traveler - **/countries/North America**
- Ability to get details about the capital city of the country as a Traveler - **/countries/<country>/1**
- Ability to add/update/delete Country and City information as a Travel Agent
    - **GET – /countries**
    - **POST - /countries**
    - **PUT - /countries/<countryId>**
    - **DELETE - /countries/<countryId>**
    - **GET – /cities**
    - **POST - /cities**
    - **PUT - /cities /<cityId>**
    - **DELETE - /cities /<cityId>**

Design


User accesses the URL
Output or resource is displayed
app.py (routes are defined)
controller
country.py
city.py
view
services
services.py
db.py
request
response

Automation with TOSCA

- Create TestCases for the APIs in GlobeHopper

Solution - GlobeHopper App

Step 1: JIRA - Have all the Functional requirements as JIRA Tickets

Step 1 (a) : JIRA – Tickets should be traceable to GitHub

Step 2: GitHub - Create a GlobeHopper repo for the Web App

Step 3: VSCode - Create Project in VSCode and create files as per the design

Step 4: MySQL - Create Database in MySQL

Step 4(a): Connect to MySQL from application.

Step 5: Python-Flask framework - Code the first API (search for all countries)

Step 5(a): Postman - Test the API

Step 5(b): Commit the code into GitHub.

Solution - GlobeHopper App

Step 6: Jenkins – Create job to build the project from GitHub and push for code inspection to SonarQube

Step 7: SonarQube – Verify, validate and test the code quality

Step 10: CI/CD – Develop, build and test for code quality for the rest of the API's **(Repeat Steps 5, 6 and 7)**

Step 11: TOSCA – Automation of API testing