RGenius Analytics

Team name, members names/emails:

RGenius

Evan Murray, Team Lead (evanmurray2@gmail.com)
Nicholas Duncan, Scribe (ncduncan@ucsb.edu)
Yash Rane (yash_rane@ucsb.edu)
Nikita Tyagi (nikitatyagi@ucsb.edu)
Wei Tung Chen (weitung_chen@ucsb.edu)

About the problem:

- **What problem the project is solving?**
  The problem we must solve is finding new ways to analyze data collected by a network of routers and displaying the data in a way that shows the important trends and any issues that need to be addressed.

- **Why is the problem important?**
  The dashboard will enable engineers to quickly pinpoint router problems and outages, and allow analysis of all routers which assists greatly in the development and maintenance process. This could potentially lead to new use cases and software implementations to improve the customer's experience with the routers, and improve the routers themselves. Customers can also see distribution details of the devices, including geographic and usage details.

- **What is innovation, the science, and new core technical advance?**
  We will be creating new analytics that will clean and organize SmartRG's datapoints into usable information for both SmartRG employees and customers. The information this provides will provide new visual ways for the employees to assess their devices and for the customers to keep track of their own customers' usage that they don't have access to currently. The information in the database is currently behind a firewall, so the customers don't have access to anything so the web interface would provide them with a lot more information.
  As mentioned, all of SmartRG's information is in the cloud but behind a firewall, so we aim to add security through authentication so that they can shift their technology to True Cloud Computing, adding more flexibility and scalability.

Overall goals and objectives:

- Backend: Analyze the data from their databases and create a server that will call different libraries to display the appropriate data
- Web: Display the analytics on a cloud-based dashboard in an informative and creative way for both world and ISP views

Project specifics:

Specification:

- Develop modules for device analytics that will run in server using company's existing databases
    - Analytics modules: Heatmap of router temperatures, Sortable grid of router information, Map of router kernel crashes, Map of locations of router, and more to come
- Create a web services with user interaction and security (user auth)
- Display the analytics on dashboard for world and ISP perspective

Design:

- Will have a web interface for customers to see different analytics from different views
- Define API for interactions between frontend web interface and backend server
- Take data from the database and clean and analyze it in the server
- Define a set of analytics to implement using different charting and mapping libraries to display the data in a user-friendly way

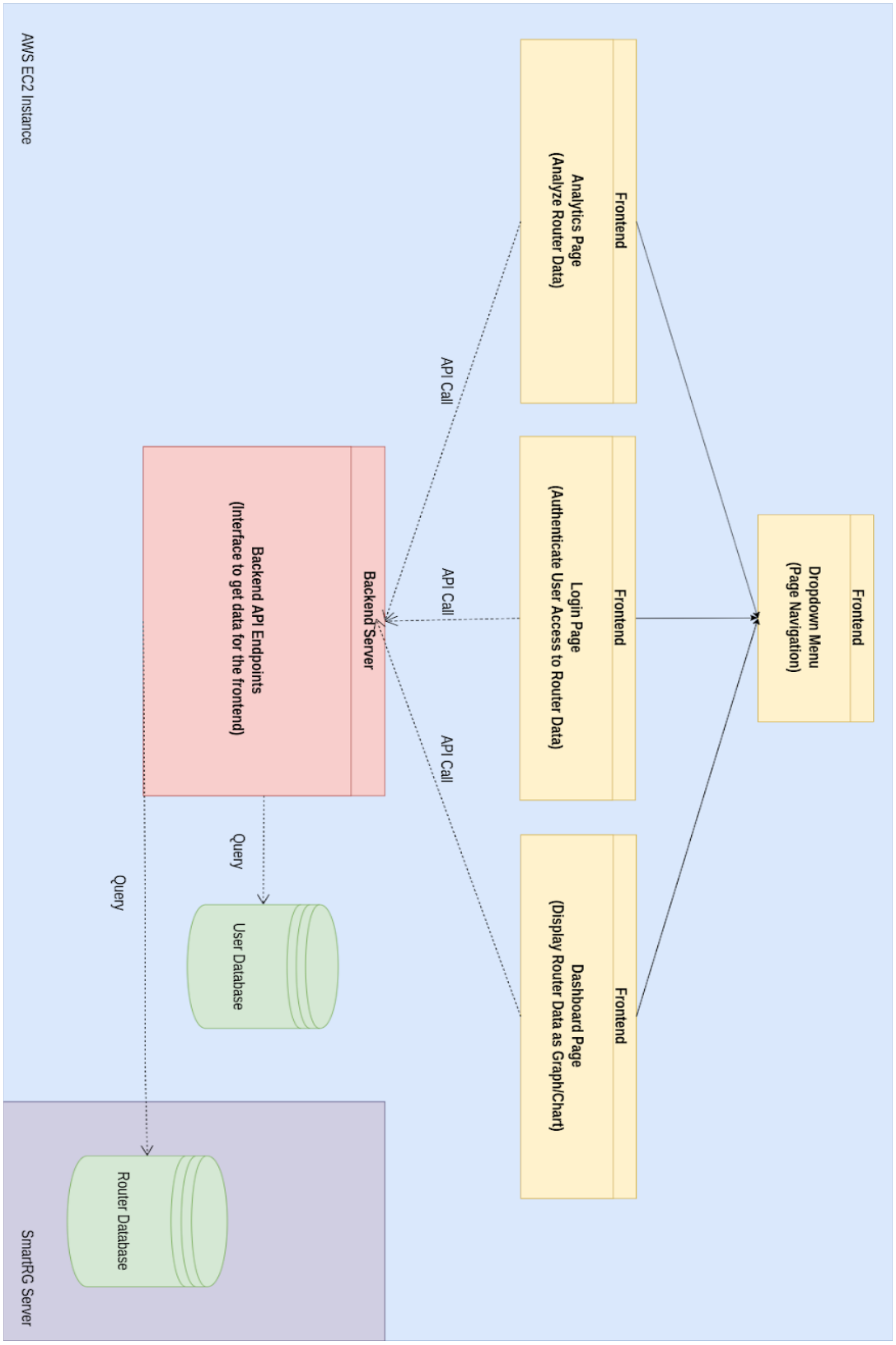
Background:

- Currently, all SmartRG routers check in once daily with a cloud server and push identifying information and some metrics into a database. The data is transferred in json format. The cloud server converts and pushes that data into a mongo database. The 'customer' information comes from a SalesForce database that is currently a static csv file on the cloud server (mac customer mapping). Currently 400 routers emit health metrics and geographical data. But, there are over 100k routers out there, so those need to be upgraded somehow to emit the health data to be used in the analytic modules.
- They have tried basic analytics using MongoDB Charts which can be used for all devices or specific devices, so these could be applied with filters to see data for specific customers or any of the other fields the checking provides.
- As of now, their server is behind a firewall so their data can only be accessed through their company's IP addresses, so it is not True Cloud Computing.

Assumptions:

- Eventually, all of their routers will be emitting the health and geographical data so that the analytics and models we create can provide helpful insight for almost all, if not all their devices.
- We will be constantly designing, developing, and testing according to the Agile Development Cycle so there are bound to be changes throughout the course of the project.
- We will add security and deploy both the backend and frontend on AWS so that the company can shift toward True Cloud Computing.

System Diagram



User Stories

1. *Use case*: As a developer, I can develop and run all of our codebase on the cloud, so users can access our website through an ip address.
   Acceptance Test: Given an AWS environment, when running our application, then both the frontend and backend code are either running on the cloud or running in a virtual environment that can be easily moved to a cloud server, and users can access our website directly.

2. *Use case*: As a User Information Database, I can store user information such as permission and login credentials so that users can properly access their account.
   Acceptance test: Given user account information, when a request is made to store or retrieve data during user login or create account, then the data is added or retrieved from the database.
   Github Commit:
   https://github.com/evanmurray2/RGenius-Analytics/commit/54854ea8d56898c3faea6b0778a49d9280f9c758
   Test Commit:
   https://github.com/evanmurray2/RGenius-Analytics/commit/445e4dba2424acc2e5fe63084b4ee808092e06e3

3. *Use case*: As an HTTP Request, I can retrieve data from a cloud server so that it can be used in the frontend.
   Acceptance test: Given an API request to the server, when the frontend requests a specific endpoint, then the data is transmitted in JSON format.

4. *Use case*: As a user, I can login to my account with the appropriate user permissions so that I can access the dashboard services.
   Acceptance Test 1: Given a valid username and password, when a login request arrives, then an authentication token is returned that will be added to all future requests requiring authentication (test: create an authenticated user and see if they can access a protected endpoint)
   Acceptance Test 2: Given an invalid username and password, when a login request arrives, then an error message is returned declaring login attempt invalid (test: create a false username and password and ensure not authenticated/can't access protected endpoint)
   Github Commit:
   https://github.com/evanmurray2/RGenius-Analytics/commit/775b56afa94843fb961c57af416a4eebb024f558

Test Commit:
https://github.com/evanmurray2/RGenius-Analytics/commit/4063b45c5cc69142c35b3d6911775ba0e2d92458

5. *Use case*: As a router I can upload my data each day so that it can be stored and used to compute analytics.
   Acceptance Test: Given a Smart/RG router, when a router sends its daily update data, then database will accept the data and store it in mongodb and show new data points each day as the routers check in.

6. *Use case*: As a SmartRG employee, I can access the world view and analytics so that I can see all of the routers' data.
   Acceptance Test: Given the username and password successfully logged in to the website with the correct user permissions, when the employee accesses the world view and analytics, then they will be able to see all the geographic and health data for all the routers on the dashboard and all the updated and desired analytics for all the routers on the analytics page.

7. *Use case*: As an ISP/ISP customer, I can view my analytics dashboard so that I can see the status of my deployed routers.
   Acceptance Test: Given a user that has logged in successfully, when they go to the dashboard page of the website, they will see a view denoting various analytics like downtime, temperature, kernel crashes etc. (test: demonstrate a logged in user navigating to the analytics page of the website and showing different analytics views)
   Github Commit:
   https://github.com/evanmurray2/RGenius-Analytics/commit/b8d7cbcfe3a3cc0f1d17828c91cc818d5059ec69
   Test commit:
   https://github.com/evanmurray2/RGenius-Analytics/commit/cb5836cfd5632a9af1f973b36cc91b35a1d275cc

8. *Use case*: As a user, I can view a map of all the kernel crashes given all my routers so that I can see where there are issues and trends.
   Acceptance test: Given user info, when the user logs in and goes to their dashboard, then there will be a map option of viewing kernel crashes with possible filters.

9. *Use case*: As a user, I can view a table of all my routers so that I can filter and see details about any specific one.

Acceptance test: Given user info, when the user logs in and goes to their router table page, then they will see all routers listed by page with filters and if they click on a specific router, they can see more information about it.

10. *Use case*: As an API, I can handle multiple requests so that server resources are maximally utilized.
    Acceptance test: Given an API, when multiple requests are received, then multiple threads carry out the requests.

11. *Use case*: As a SmartRG employee, I can verify a customer account once it has been created so that only valid customers can access their data.
    Acceptance Test: Given valid company details for an account, when a customer pays for the new service an account can be created with specified permissions group (test: verify that an account is generated in backend database when supplied with credentials)
    Github Commit:
    https://github.com/evanmurray2/RGenius-Analytics/commit/565b95015138c84fd3a251358f2b155b4d912911
    Test Commit:
    https://github.com/evanmurray2/RGenius-Analytics/commit/e710c73292380fac7bea08847c090394beedc3fc

12. *Use case*: As a user, I can view regular updates on my dashboard so that I have the most recent and accurate data.
    Acceptance test: Given the daily check-in process, when the database is updated with the recent data, then the frontend receives it and at least one of the charts will show updated data automatically.
    Github Commit:
    https://github.com/evanmurray2/RGenius-Analytics/commit/2b656eea3342a9039a2847a9c3b6f6c78a9e5a7a
    Test Commit:
    https://github.com/evanmurray2/RGenius-Analytics/commit/c837d8d8cfd41432e48e047093dc7dc713aa93c6

Appendix

Technologies Used:

- Cloud technologies- AWS, Docker
- Frontend frameworks- React
- Backend frameworks- Django, Python

- Database libraries- MongoDB
- Libraries for displaying data- Chart.js, Mapbox, React-Bootstrap