

September 29 , 2025

# B.R.A.D. DEMO 4



# TABLE OF CONTENT

01 Sales Pitch

02 Live Demo

03 Architectural  
Diagram

03 Quality  
requirements

04 Deployment model

05 WOW Factors

06 Testing

07 Website Usability &  
Experience

08 Meet Our Team

09 Q&A

# REPOSITORY LINK



# WHAT IS BRAD AND WHY IS IT NEEDED?



- **PHISHING, MALWARE-HOSTING, AND FRAUDULENT DOMAINS ARE INCREASING AT AN ALARMING RATE.**
- **CURRENT REPORTING SYSTEMS ARE SLOW, MANUAL, AND OFTEN LACK PROPER FORENSIC INSIGHTS.**
- **SECURITY PROFESSIONALS SPEND VALUABLE TIME PIECING TOGETHER EVIDENCE BEFORE ACTION CAN BE TAKEN.**

# **BRAD** **Bot to Report Abusive Domains**

- automates the forensic investigation of suspicious domains.
- BRAD safely visits reported domains, extracts content, detects malware, collects metadata, and produces structured reports.



# Who Is BRAD For

- Everyday users who want to safely report malicious sites.
- Businesses and organizations protecting their staff and customers.
- Cybersecurity analysts and investigators who need deeper insights.
- Domain registrars & DNS providers (like our partner DNS.Business) who require automated systems to identify and act on abusive domains.

# How does B.R.A.D work?

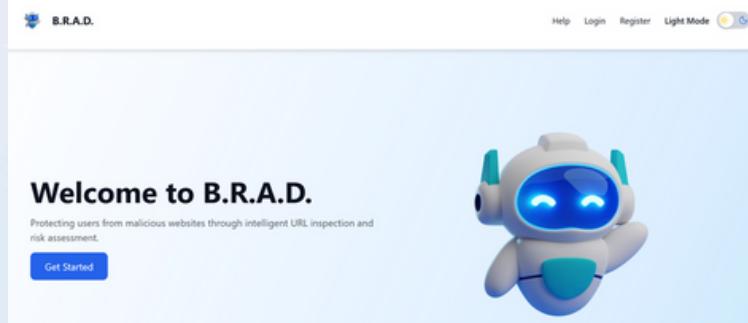
1



User receives a link they are not sure is safe to visit

2

Visit B.R.A.D website and login or create an account



3

SUBMIT



User submits the URL

4

B.R.A.D analyzes the URL by scraping the website and collecting useful information



↓



Investigators receive an email that there is a new report to analyse

5

Investigator claims a report then reviews the bot analysis,



User is able receives an email notifying them that their report has been resolved and they can view the verdict

6

User can view their report history and all resolved report's filtered analysis



# LIVE DEMO

We would like to show you everything we  
have implemented for B.R.A.D.

# How Does our bot work?

# Role based access Control and User responsibilities

## Reporter/General User

- When registering a user is automatically a reporter
- Can report URL and add evidence
- View report history and a very filtered analysis of their report

## Admin

- Manage and add users
- User statistics to monitor investigators and reports
- View all report but can't claim or give a verdict

## Investigator

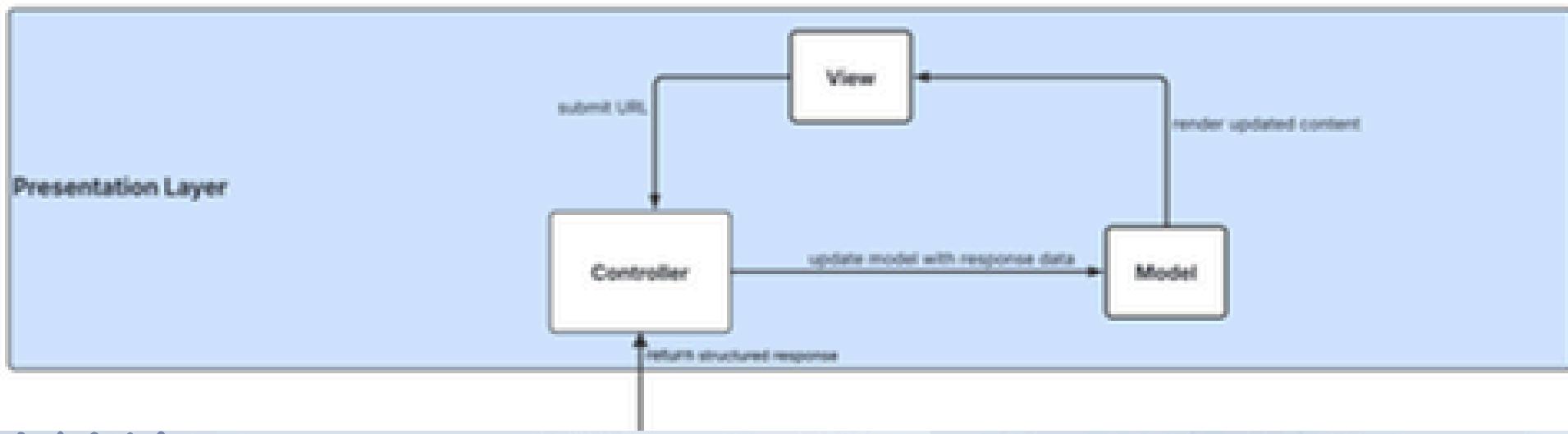
- Can only be added by an admin
- Responsible for claiming reports and using bot's analysis as well as domain similarity checks and evidence to mark a report as malicious or benign

# ARCHITECTURAL DIAGRAM (1/6)

Architecture Diagram

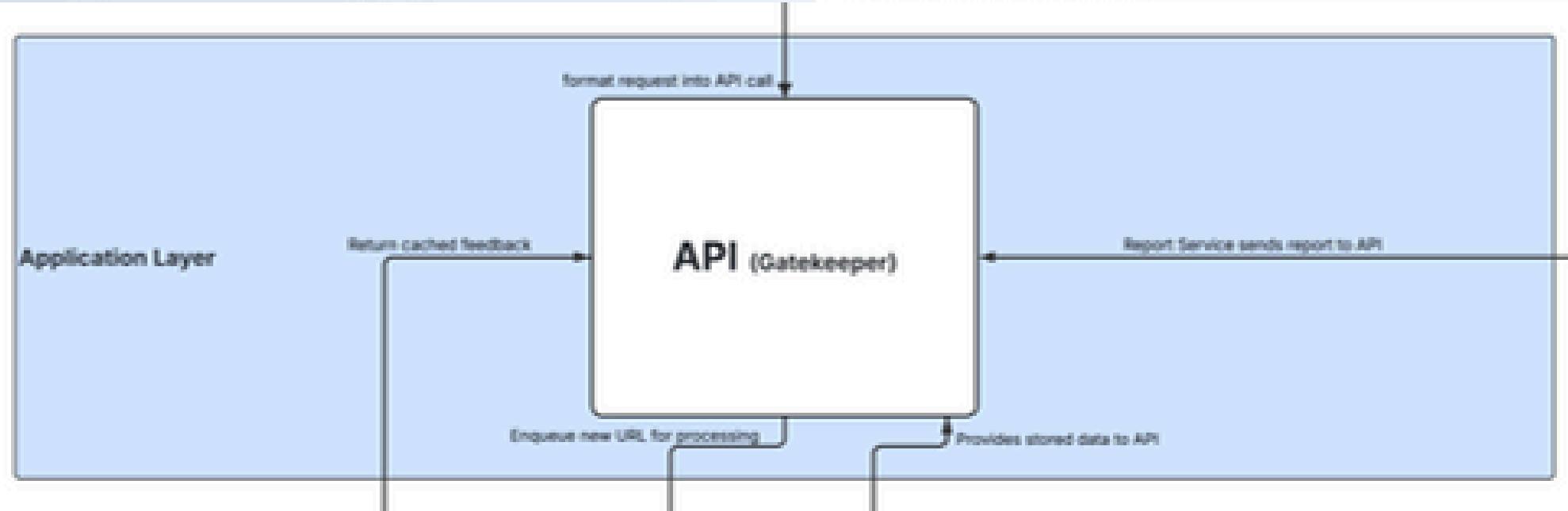


# PRESENSTATION LAYER (2/6)



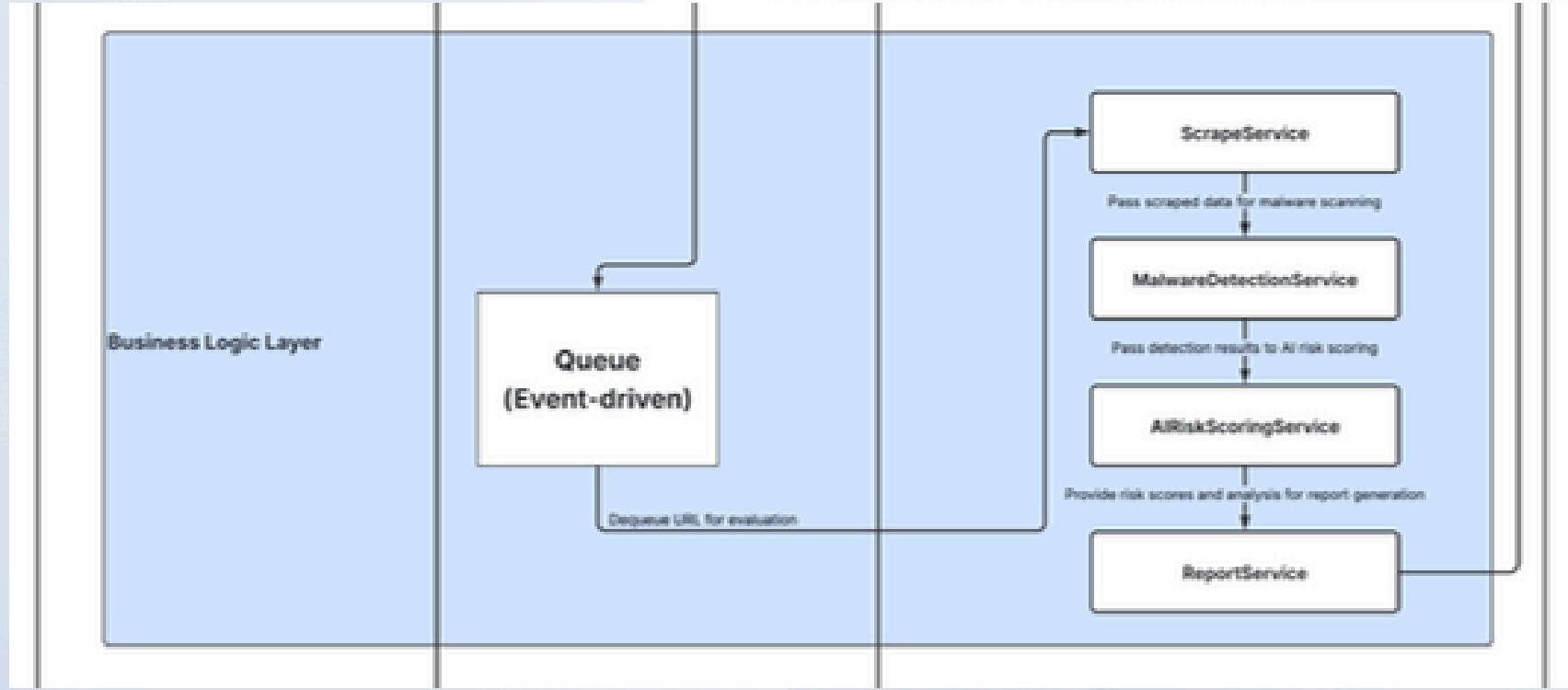
- WHERE USERS ARE ABLE TO INTERACT WITH THE SYSTEM
- **MODEL VIEW CONTROLLER APPLIED**
  - THE **VIEW** IS WHAT USERS ARE ABLE TO SEE
  - **CONTROLLER** HANDLES SUBMISSION OF URL'S AND EVIDENCE
  - **MODEL** SHOWS STATE OF DATA BACK TO USER
- MAKES OUR FRONTEND SIMPLE, RESPONSIVE AND EASY TO MAINTAIN

# APPLICATION LAYER (3/6)



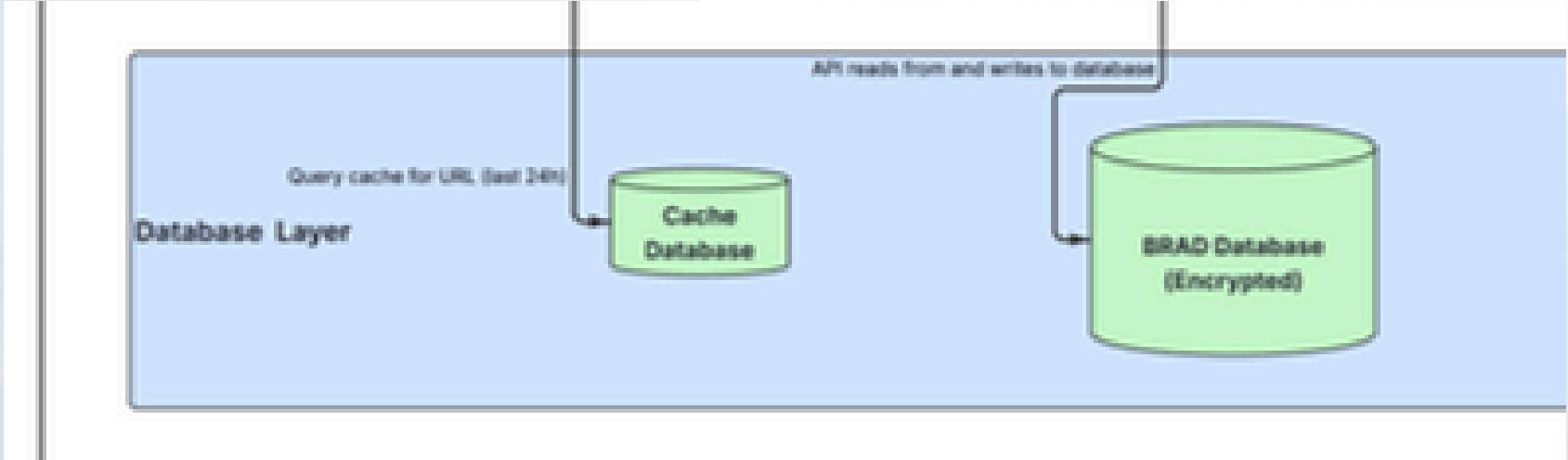
- **GATEKEEPER PATTERN IMPLEMENTED**
- API IS THE SINGLE ENTRY POINT INTO THE BACKEND
- ENFORCES ROLE BASED ACCESS CONTROL AND MFA
  - USED TO VALIDATE REQUESTS

# BUSINESS LOGIC LAYER (4/6)



- COMBINATION OF **EVENT DRIVE ARCHITECTURE AND PIPE AND FILTER PATTERN**
- THE QUEUE ENABLES EVENT DRIVEN PROCESSING
  - WHEN A NEW URL IS SUBMITTED IT IS ADDED TO THE QUEUE
  - USING **PIPE AND FILTER PATTERN**, THE URL FLOWS THROUGH A SEQUENCE OF SERVICES, EACH PERFORMING A SPECIFIC TASK AND PASSING IT TO THE NEXT

# DATABASE LAYER (5/6)



- CACHED DATA STORED FOR RECENTLY SUBMITTED URL'S
- ENCRYPTED RECORD STORED IN BRAD DATABASE
- ENSURES RELIABILITY, PERFORMANCE, COMPLIANCE

# ARCHITECTURAL DIAGRAM (6/6)

THE COMBINATION OF THESE LAYERS AND PATTERNS OUR ARCHITECTURE ENSURES:

- 1. SECURITY
- 2. COMPLIANCE
- 3. RELIABILITY
- 4. SCALABILITY
- 5. MAINTAINABILITY

WHILE KEEPING THE DESIGN CLEAN AND STRUCTURED

# QUALITY REQUIREMENTS(1/6)

IN DEVELOPING THE BOT TO REPORT ABUSIVE DOMAINS (BRAD) SYSTEM, WE HAVE SELECTED SEVERAL ARCHITECTURAL STYLES TO MEET KEY QUALITY REQUIREMENTS, INCLUDING **SECURITY, COMPLIANCE, SCALABILITY, RELIABILITY, USABILITY, AND MAINTAINABILITY**.

# QUALITY REQUIREMENTS(2/6)

## EVENT-DRIVEN ARCHITECTURE (EDA) QUALITY REQUIREMENTS ADDRESSED:

- **SCALABILITY:**

EVENTS TRIGGER DECOUPLED PROCESSES THAT SCALE INDEPENDENTLY.

- **PERFORMANCE:**

SUPPORTS REAL-TIME, ASYNCHRONOUS EXECUTION.

- **RELIABILITY:**

PERSISTENT QUEUES ALLOW TASK RECOVERY AFTER FAILURE.

# QUALITY REQUIREMENTS(3/6)

## GATEKEEPER PATTERN

### QUALITY REQUIREMENTS ADDRESSED:

- **SECURITY:** VALIDATES AND AUTHENTICATES ALL INCOMING TRAFFIC.
- **RELIABILITY:** SHIELDS SERVICES FROM OVERLOAD AND MALFORMED INPUT.
- **COMPLIANCE:** ENFORCES LOGGING AND DATA PROTECTION RULES (POPIA/GDPR).

# QUALITY REQUIREMENTS(4/6)

## CLIENT-SERVER MODEL

### QUALITY REQUIREMENTS ADDRESSED:

#### **USABILITY:**

FRONTEND PROVIDES CLEAR UI FOR REPORTERS AND INVESTIGATORS.

#### **SECURITY:**

BACKEND CONTROLS SENSITIVE OPERATIONS AND DATA.

#### **COMPLIANCE:**

ENFORCES LOGGING AND DATA PROTECTION RULES (POPIA/GDPR).

# QUALITY REQUIREMENTS(5/6)

## LAYERED ARCHITECTURE

### QUALITY REQUIREMENTS ADDRESSED:

- **MAINTAINABILITY:**  
EACH LAYER CAN BE CHANGED INDEPENDENTLY.

- **SECURITY:**  
SENSITIVE LOGIC IS ISOLATED IN PROTECTED BACKEND LAYERS.

- **RELIABILITY:**  
FAULTS ARE CONTAINED WITHIN INDIVIDUAL LAYERS.

# QUALITY REQUIREMENTS(6/6)

## PIPE AND FILTER PATTERN

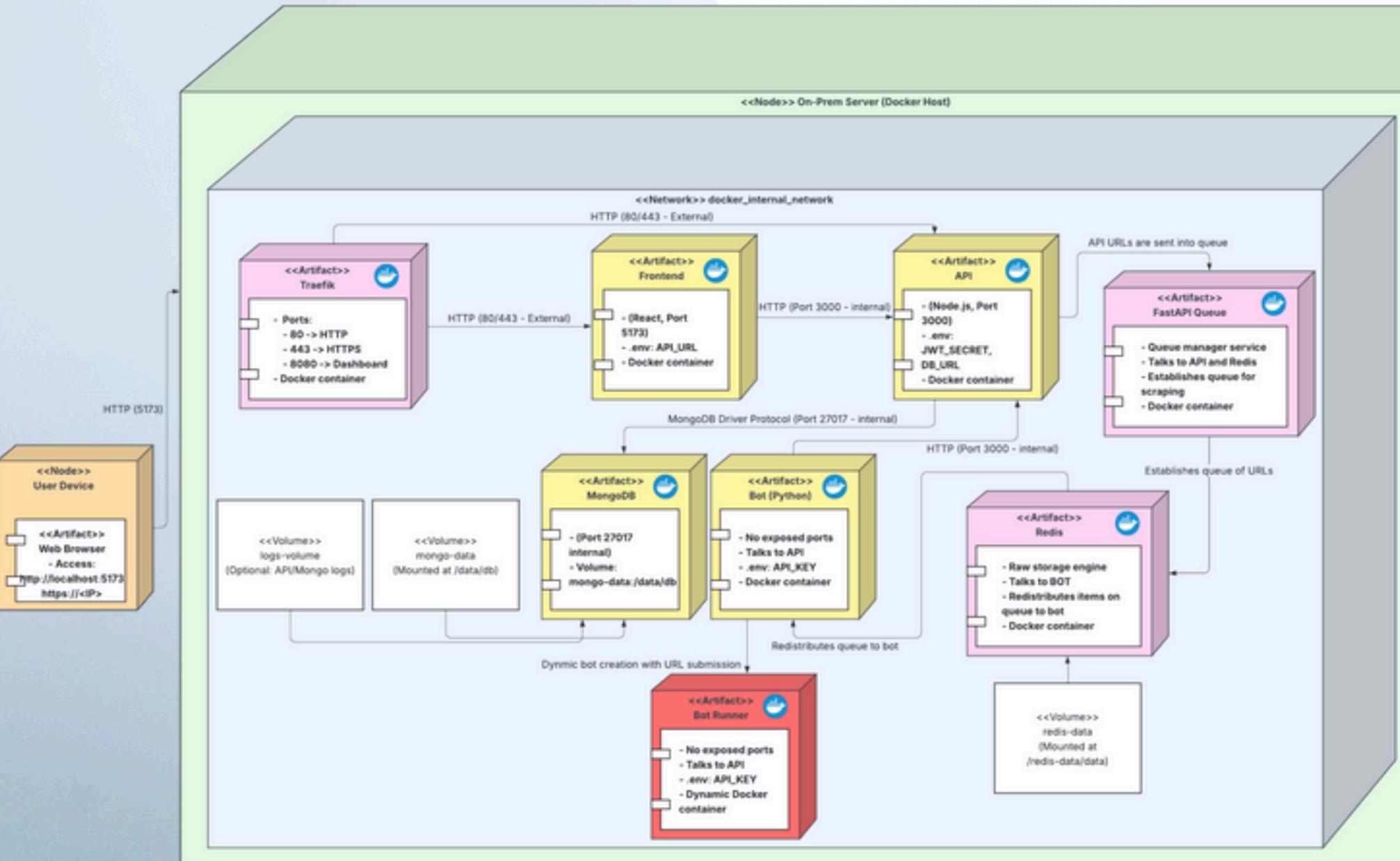
### QUALITY REQUIREMENTS ADDRESSED:

- **MAINTAINABILITY:**  
FILTERS CAN BE UPDATED WITHOUT BREAKING THE FLOW.
- **RELIABILITY:**  
PIPELINE RESUMES FROM FAILED STAGES WHERE POSSIBLE.
- **PERFORMANCE:**  
PROCESSING STEPS CAN BE PARALLELIZED AND OPTIMIZED.

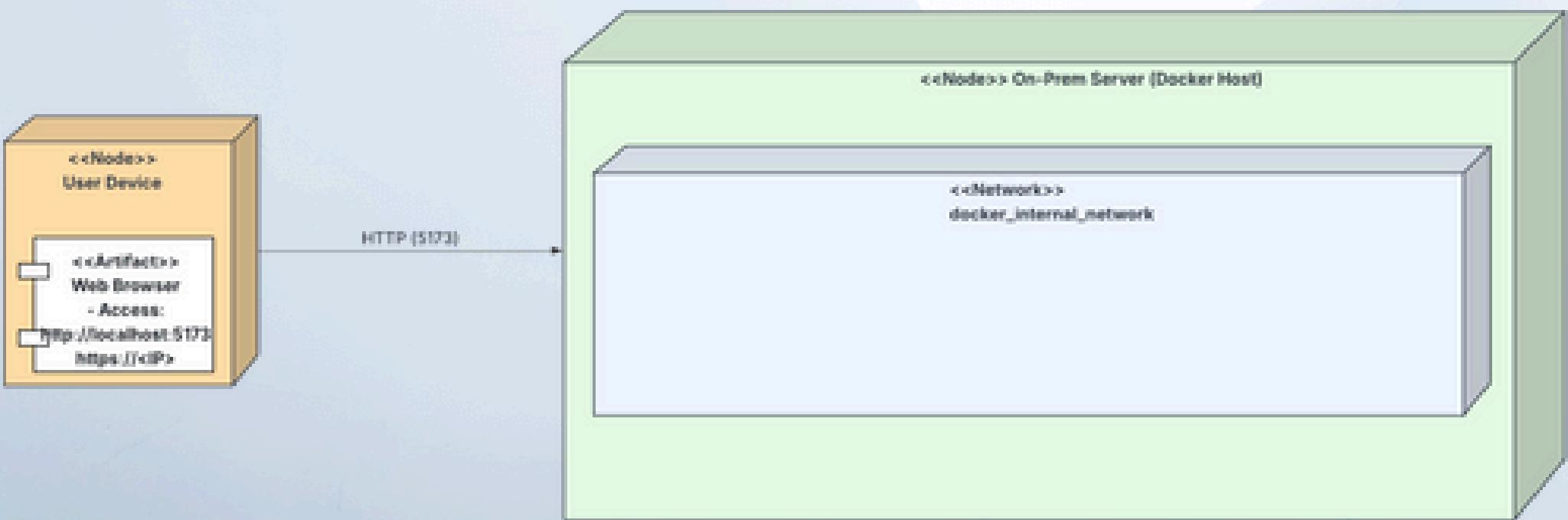
# QUALITY REQUIREMENTS TESTING



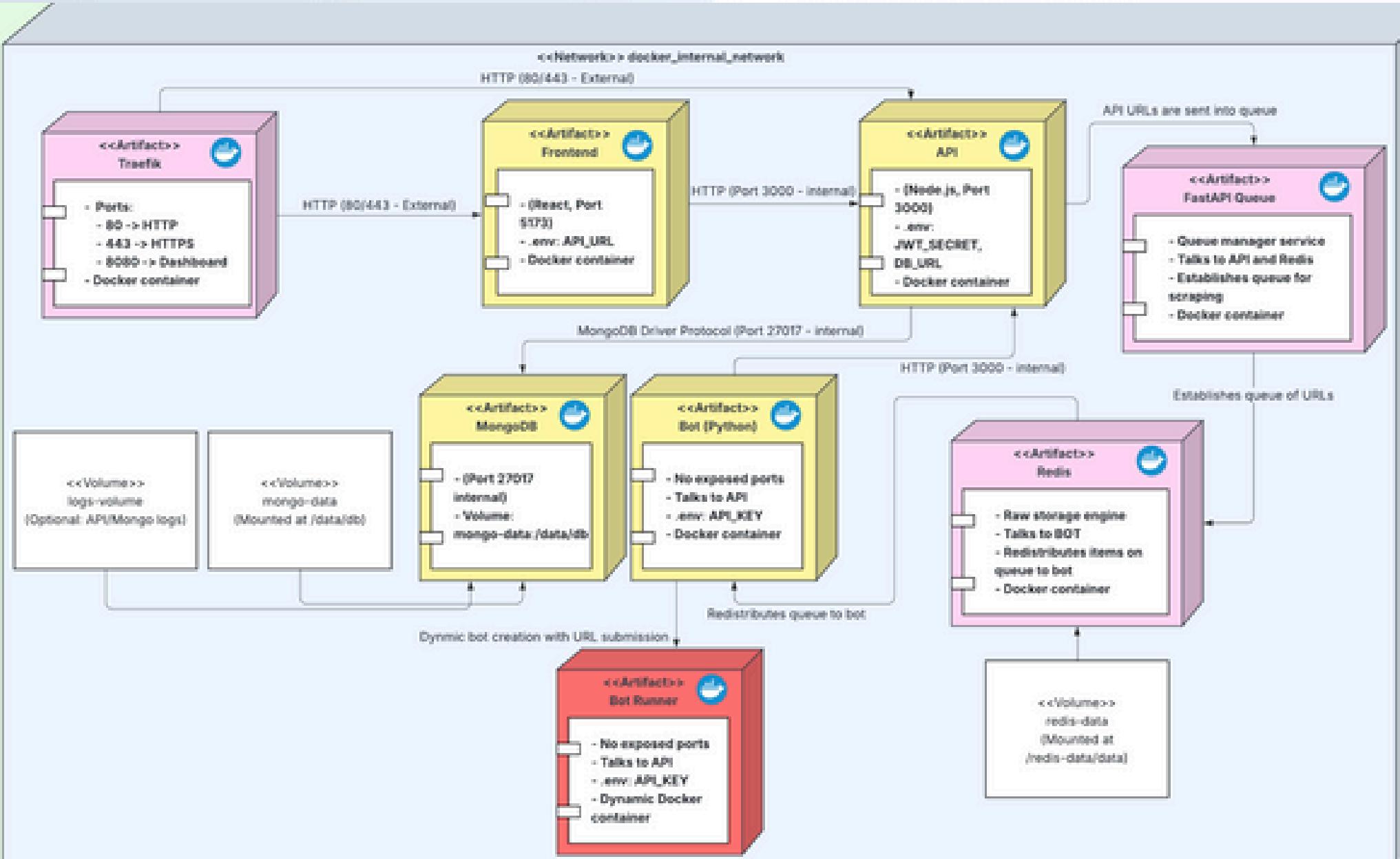
# DEPLOYMENT MODEL (1/3)



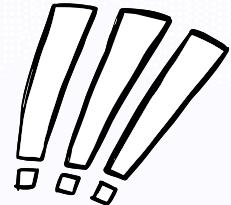
# DEPLOYMENT MODEL (2/3)



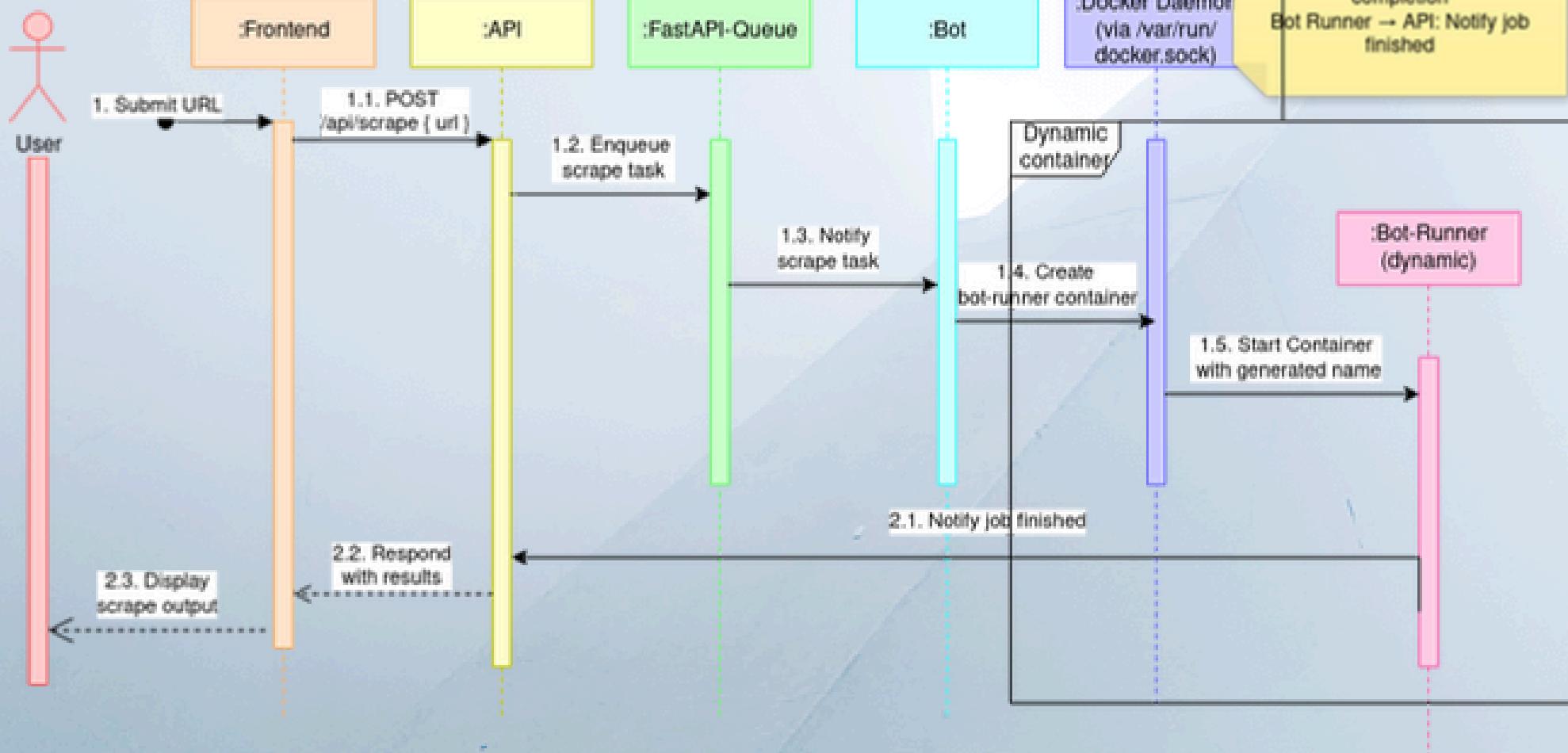
# DEPLOYMENT MODEL (3/3)



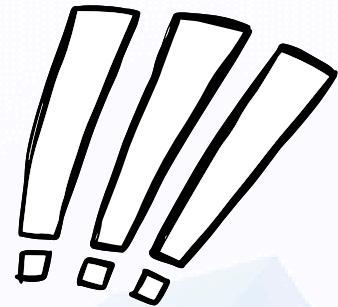
# WOW FACTORS(1/2)



## LIVE SANDBOXING



# WOW FACTORS(2/2)



## STATISTIC DASHBOARDS

TO HELP GIVE INVESTIGATORS AND ADMIN A BETTER OVERVIEW OF REPORTS AND INVESTIGATION PROCESS

### ADMIN

- AVG BOT ANALYSIS TIME
- AVG INVESTIGATOR ANALYSIS TIME
- AVG RESOLUTION TIME
- REPORT DISTRIBUTION
- REPORTS OVER TIME
- INVESTIGATOR STATS
- TOP DOMAINS REPORTED

### INVESTIGATOR

- REPORT DISTRIBUTION
- REPORTS OVER TIME
- TOP REPORTED DOMAINS

# Testing and CI/CD Pipeline

**Actions**

New workflow

All workflows

[github/workflows/deploy-no-docker.yml](#)

**CI/CD Pipeline**

CI/CD Pipeline

Management

Caches

Annotations

Runners

Usage metrics

Performance metrics

**CI/CD Pipeline**

[deploy.yml](#)

73 workflow runs

- Merge pull request #203 from COS301-SE-2025/ui**  
CI/CD Pipeline #73: Commit [431ff1f](#) pushed by [carinda-smith](#) dev
- Merge pull request #202 from COS301-SE-2025/deployment-proxy**  
CI/CD Pipeline #72: Commit [a6d1c7e](#) pushed by [EthanVletter](#) dev
- Merge pull request #200 from COS301-SE-2025/settings**  
CI/CD Pipeline #71: Commit [a588a0b](#) pushed by [EthanVletter](#) dev
- Merge pull request #198 from COS301-SE-2025/ui**  
CI/CD Pipeline #70: Commit [fe0b949](#) pushed by [EthanVletter](#) dev
- Merge pull request #196 from COS301-SE-2025/stats-frontend**  
CI/CD Pipeline #69: Commit [1052d56](#) pushed by [carinda-smith](#) dev
- Merge pull request #194 from COS301-SE-2025/notifications**  
CI/CD Pipeline #68: Commit [c7acd1b](#) pushed by [carinda-smith](#) dev
- Merge pull request #192 from COS301-SE-2025/ui**  
CI/CD Pipeline #67: Commit [3a50305](#) pushed by [carinda-smith](#) dev

# Frontend Unit Testing

```
> ✓ TERMINAL
PS C:\Cos 301 Capstone\BRAD\frontend> npm test
PASS  src/__tests__/_ReporterDashboard.test.jsx
PASS  src/__tests__/_register.test.jsx
PASS  src/__tests__/_login.test.jsx
PASS  src/__tests__/_UserSettingsPage.test.jsx
PASS  src/__tests__/_InProgressReportsPage.test.jsx (5.516 s)
PASS  src/__tests__/_ResolvedReportsPage.test.jsx (5.541 s)
PASS  src/__tests__/_PendingReportsPage.test.jsx (5.556 s)
PASS  src/__tests__/_HelpPage.test.jsx
PASS  src/__tests__/_InvestigatorDashboard.test.jsx
PASS  src/__tests__/_ReportsPage.test.jsx
PASS  src/__tests__/_ManageUsersPage.test.jsx
PASS  src/__tests__/_AdminDashboard.test.jsx

Test Suites: 12 passed, 12 total
Tests:       34 passed, 34 total
Snapshots:   0 total
Time:        12.375 s
```

Every time a new frontend component or page is created, a corresponding unit test file must be created. The test file is located a centralized `__tests__` folder and named to match the component (e.g. `__tests__ /login.test.jsx` for `/app/login/page.jsx`).

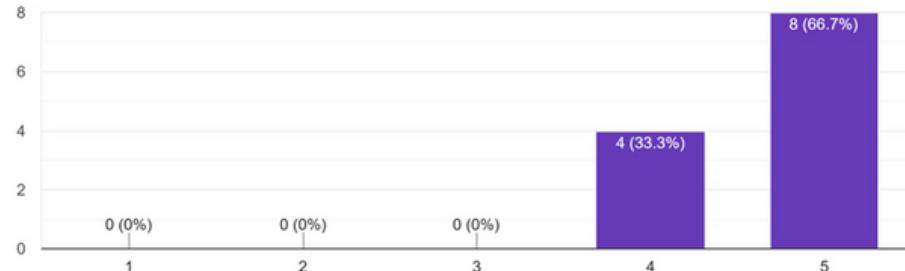
- FRAMEWORKS AND TOOLS:

- **JEST:** USED AS THE PRIMARY TEST RUNNER AND ASSERTION LIBRARY FOR ALL FRONTEND UNIT TESTS.
- **REACT TESTING LIBRARY:** USED FOR RENDERING COMPONENTS, SIMULATING USER INTERACTIONS, AND QUERYING DOM ELEMENTS IN A WAY THAT REFLECTS REAL USER BEHAVIOUR.

# Website Usability & Experience Questionnaire

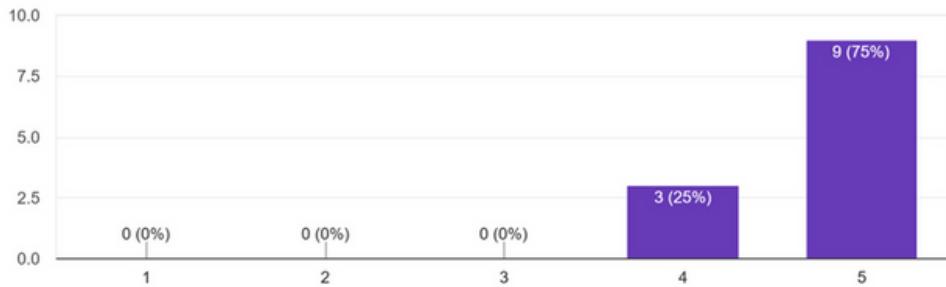
## Ease of navigation (finding what you need)

12 responses



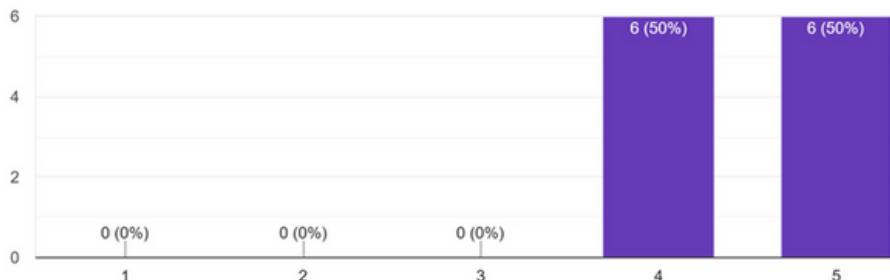
## Ease of understanding website content

12 responses



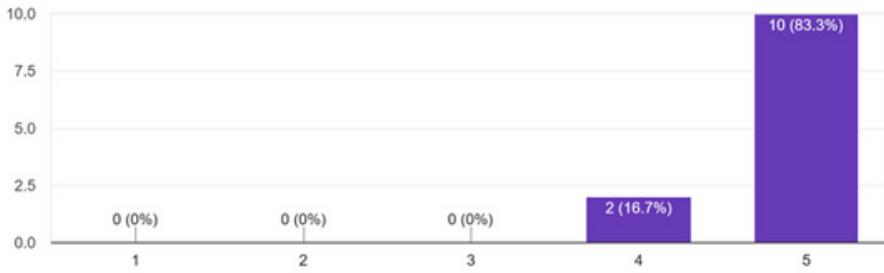
## Clarity of menu/sidebar structure

12 responses



## Readability of text (size, font, spacing)

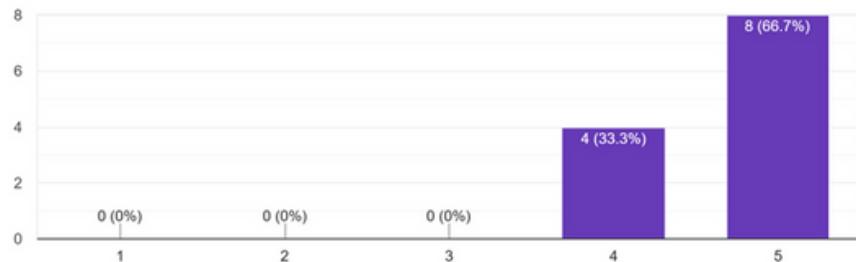
12 responses



# Website Usability & Experience Questionnaire

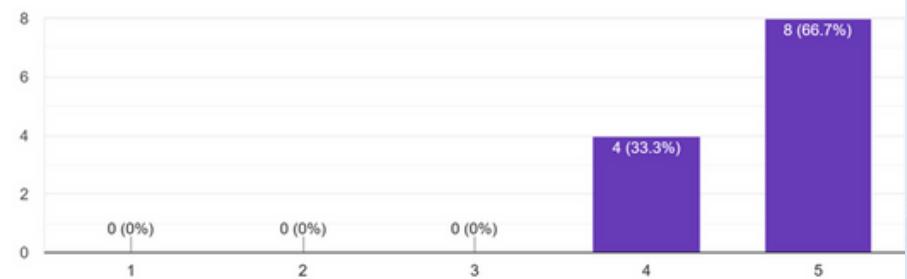
## Visual appeal of website design

12 responses



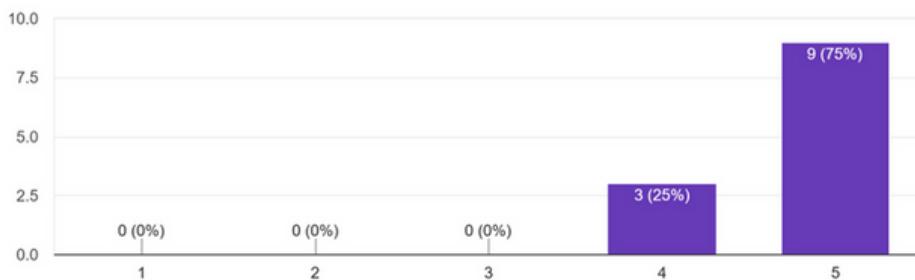
## Use of colors (contrast, consistency)

12 responses



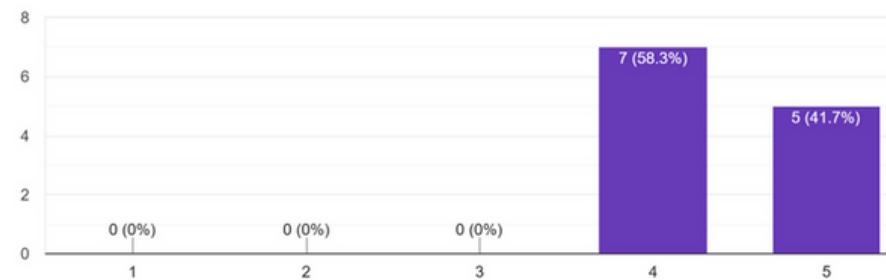
## Dark/Light mode usability

12 responses



## Loading speed of the website

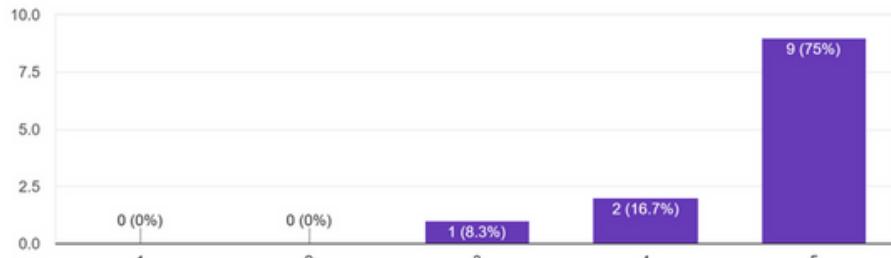
12 responses



# Website Usability & Experience Questionnaire

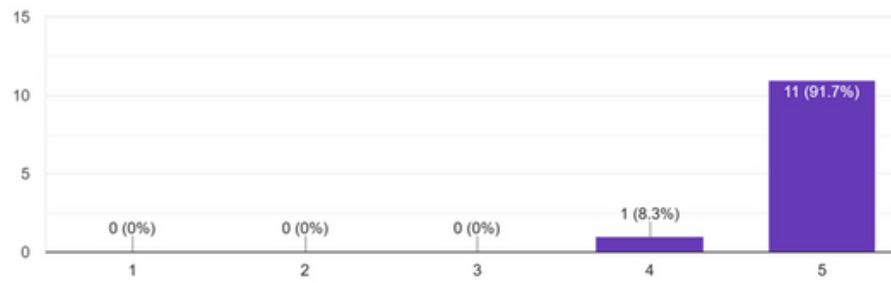
## Usefulness of FAQs section

12 responses



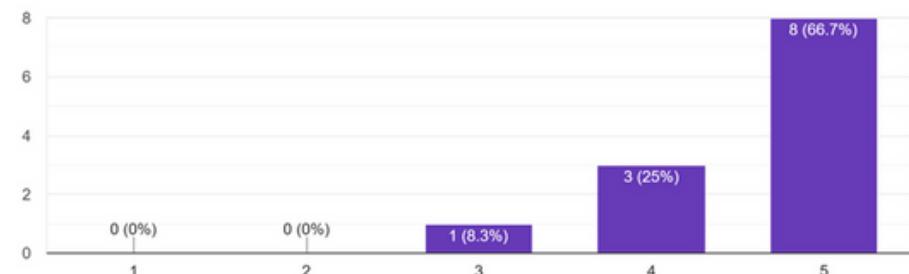
## Ease of completing tasks (e.g., sign up, checkout)

12 responses



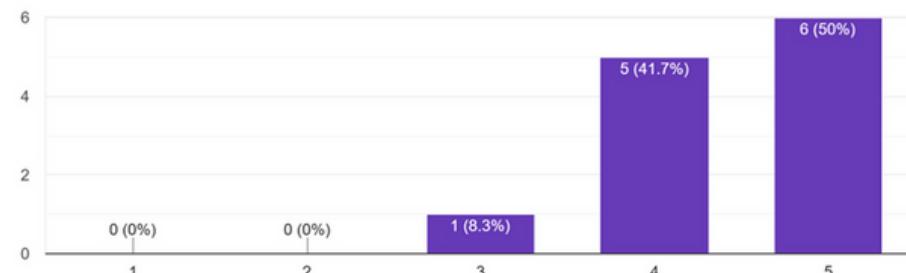
## Help menu / support options clarity

12 responses



## Error messages clarity (if applicable)

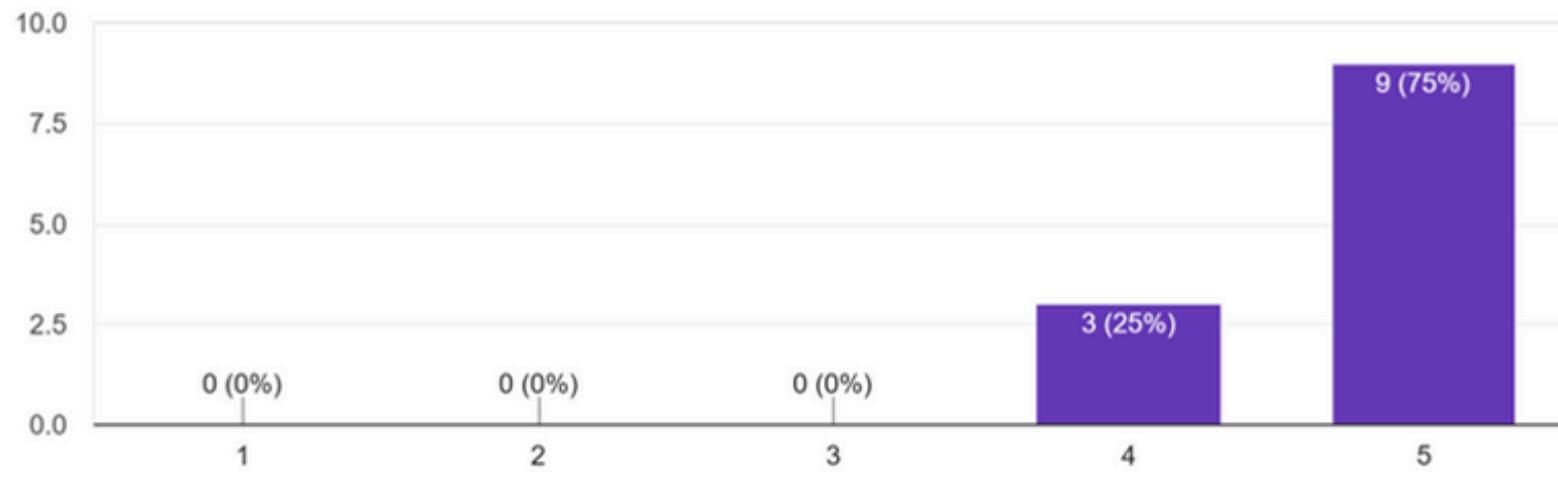
12 responses



# Website Usability & Experience Questionnaire

## Overall satisfaction with the website

12 responses



## Additional Comments / Suggestions

2 responses

The colour grading makes it easy to follow and gives an overall aesthetic appeal to the appearance of the website. I prefer the dark mode. I think that the light mode, is very bright. It's fairly easy to understand what you can accomplish with BRAD.

Personal preference. I like side bars or top bars to be locked.

# Meet Our Team



**Ethan Vletter**

Project Manager,  
DevOps



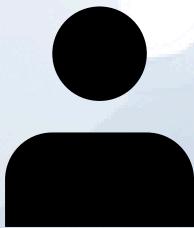
**Megan Pretorius**

UI/UX, API



**Tebatso  
Mahlathini**

API



**Salome Kalaka**

Architecture , API



**Carinda Smith**

BOT,API

# QGA

# THANK YOU