# TEAM 2
# COMPREHENSIVE ANALYSIS AND CLASSIFICATION OF MALICIOUS PE FILES USING MACHINE LEARNING

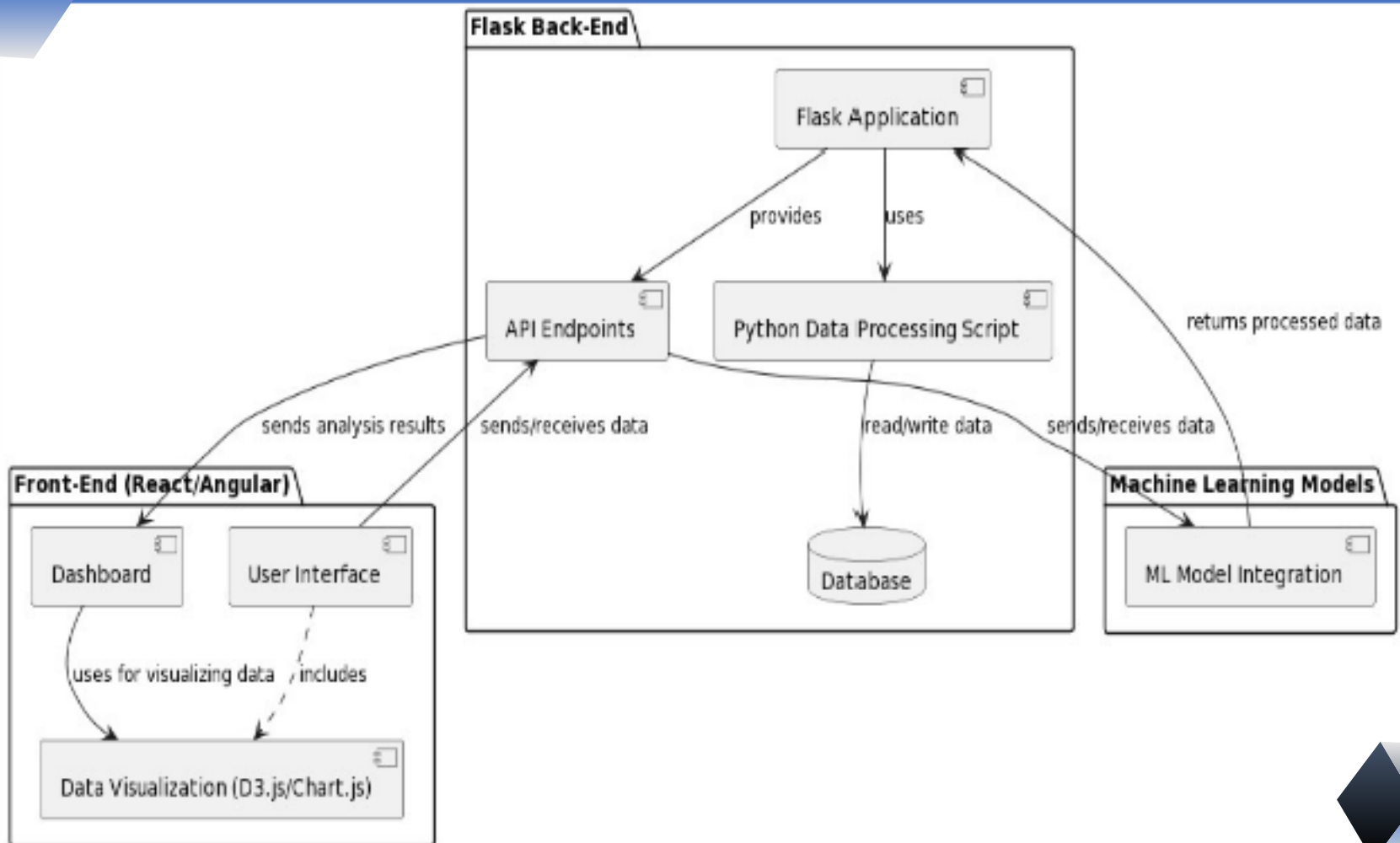
# Table of contents

# 01

# Purpose statement

The primary objective of our project is to address t critical challenge of cybersecurity, specifically focusing on the detection and analysis of malicious Portable Executables (PEs). PEs, being a prevalent vector for malware and cyberattacks, pose a significant threat to digital security. Our aim is to develop a sophisticated tool that combines dynamic and static analysis methods with advanced machine learning techniques to accurately identify and classify these malicious PEs.

# Methodology


Flask Back-End
Flask Application
provides
uses
API Endpoints
Python Data Processing Script
returns processed data
sends analysis results
sends/receives data
read/write data
sends/receives data
Front-End (React/Angular)
Dashboard
User Interface
Database
Machine Learning Models
ML Model Integration
uses for visualizing data
includes
Data Visualization (D3.js/Chart.js)

| | |
|---|---|
| Introduction and Dataset Information | The dataset comprised 15,000 PE files evenly split between benign and malicious classifications for the training set, 5,000 PE files for the initial model evaluation test set, and a private test set of 10,000 PE files. |
| Static Analysis and Data Acquisition | This phase dealt with the exploration of the dataset's structure and attributes, focusing on data consistency, relevance, and comprehensiveness. |
| Exploratory Data Analysis (EDA) | The analysis included checking data balance, handling missing values, and analyzing numerical values to identify patterns, skewness, and potential outliers. |
| Dataset Creation and Preprocessing | This involved dropping redundant columns, handling missing data, and segmentation. |
| Dynamic Analysis Using API Sequences and File Activities: | This included data acquisition, integration, training, and evaluation of models like XGBoost, LightGBM, and LSTM on API sequences and file activities. |

## Feature Extraction and Representation

The project extracted features such as guest signers, section attributes, entropy, resources, version information, exports, directories, checksum match, and imports from DLLs like kernel32. dll, user32. dll, oleaut32. dll, and advapi32. dll. Feature representation focused on transforming raw data into a structured format for analysis and modeling.

## Feature Selection and Dimensionality Reduction

Employed methods like ExtraTreesClassifier and SelectFromModel to identify the most informative features and reduce dataset dimensionality.

## Decision Trees and Random Forest Modeling

These models were used for classification, with various approaches and hyperparameter tuning employed to optimize performance.

## Model Evaluation and Comparison

The project compared different machine learning models like Decision Trees, Random Forest, Multi-Layer Perceptron, Gaussian Naive Bayes, and Multinomial Naive Bayes on a test set to evaluate their performance.

## Conclusion

The project successfully used both static and dynamic analysis methods to detect malicious PEs, highlighting its potential efficacy in cybersecurity applications.

# 03

# Objectives

## Machine Learning Integration

This integration aims to enhance the tool's capability to accurately detect and classify malicious Portable Executables (PEs), providing an intelligent layer for proactive threat mitigation.

## Dataset Creation and Preprocessing

The objective here is to create a meticulously curated dataset, comprising a balanced representation of benign and malicious PE files. Rigorous preprocessing and feature engineering techniques will be applied to ensure the dataset's quality and relevance.

## Dynamic and Static Analysis:

The project aims to employ a comprehensive approach by incorporating both dynamic and static analysis methods. Dynamic analysis, utilizing execution reports of PE files, and static analysis, leveraging intrinsic file attributes, collectively contribute to a holistic understanding of PEs

# 04

# Practical Demo

99% accurancy

# 05

# Our Team

Back End: Majid Rahmanov, Ibrahimkhalil, Nihad Shirinli, Yusif Touri

Front End: Mirmusa Feyziyyev, Turan Jabbarli, Ismayil

Machine Learning: Majid Rahmanov, Maghrur

Research: Nemat Rahimli, Ismayil

Report: Nemat Rahimli

# 06

# Conclusion

Literature review and reference to sources used:

1. Front End: W3Schools, Geeks4Geeks
2. Back End:
3. Machine Learning:

Project plan reflection:

1. What was learned: designing responsive web pages , UI design , Web page functionality
2. Time management : Team collaboration and time management, to be ready until deadline
3. Obstacles: lack of knowledge in development , short time

Personal reflection:

1. Technical issues: Deployment errors , shortage of web side development
2. Tools: React , Bootstrap ,
3. What could be better next time? – We will develop this project more and improve user interface to be more functional

# Thanks!