Android Evidence Database

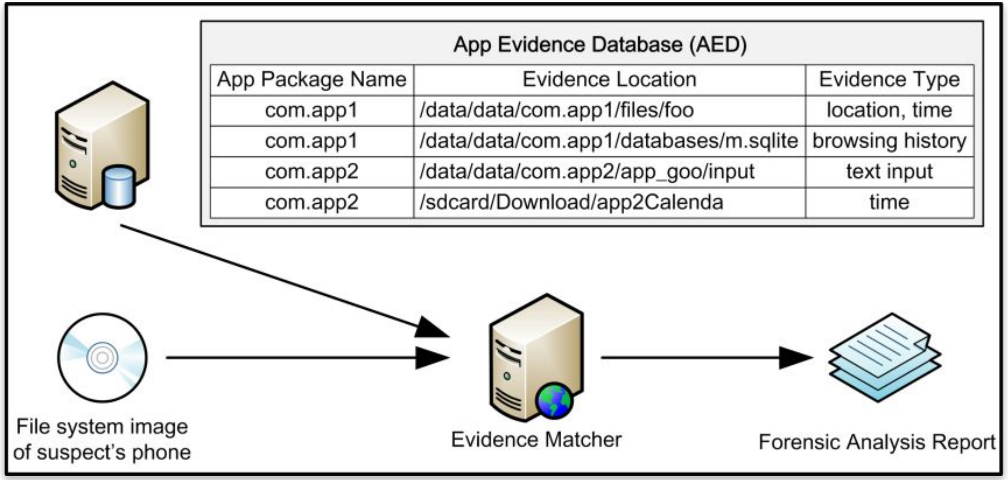
For Forensic Analysis

Team: sdmay19-38
Advisors: Dr. Neil Gong & Dr. Yong Guan
Clients: NIST Center of Excellence in
Forensic Sciences - CSAFE at Iowa State University

Project Plan

Problem Statement

- Current digital forensic investigation techniques are slow, tedious, inaccurate and may not yield complete results.
- Investigators need to manually search phones for evidence
- There is no standard for where applications store data.



App Evidence Database (AED)

App Package Name	Evidence Location	Evidence Type
com.app1	/data/data/com.app1/files/foo	location, time
com.app1	/data/data/com.app1/databases/m.sqlite	browsing history
com.app2	/data/data/com.app2/app_goo/input	text input
com.app2	/sdcard/Download/app2Calenda	time

Functional & Non-Functional Requirements

Functional

1. App Store Crawlers
   - Collect Application Metadata
   - Collect Apk Files
   - Store all collected data in the database
2. Application Post-Processing
   - Store forensic report data in the database
3. Website
   - Query database
   - Filter query results
   - Download APK files

Non-Functional

1. System must be able to scale
   - Due to the large quantity of stores/applications
2. Each Crawler must process its’ website weekly
   - New applications and versions will be added

Constraints & Considerations

- Use a NoSQL database
- System must operate twenty-four seven
- System must be easily adaptable
- Data must not be tampered with after collection
- Legal issues
  - App Stores' TOS, University Regulations
- Cost Analysis
  - Software
    - Python, Javascript, MongoDB
  - Storage
    - LSS Drive

Market Survey

- **On the market**
    - Forensic tools
        - Security loopholes
        - Malicious intent
    - Database
        - Individual Stores
        - APK Files

- **Our Project**
    - Database
        - Metadata
        - Apk Files
        - App stores
        - Versions
    - Website/ UI

Potential Risks & Mitigation

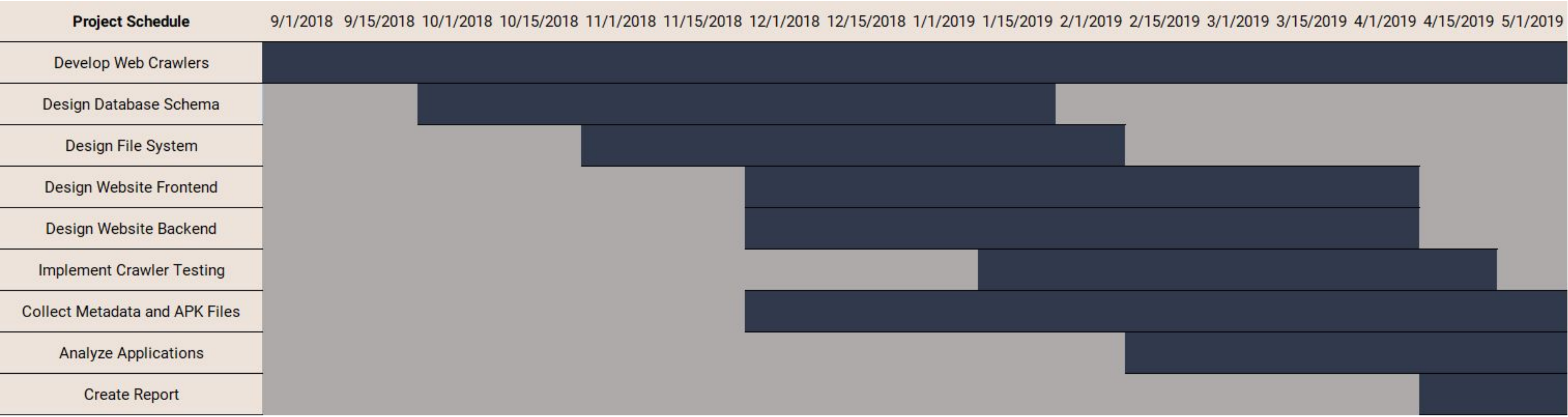
Anticipated Risks

- Resource Acquisition
- Domain Knowledge
- Downloading illegal apps from 3rd party stores

Actual Risks

- Legal
- Time Management
- Crawler Rate Limiting
- Public Access
- Storage

Schedule & Milestones

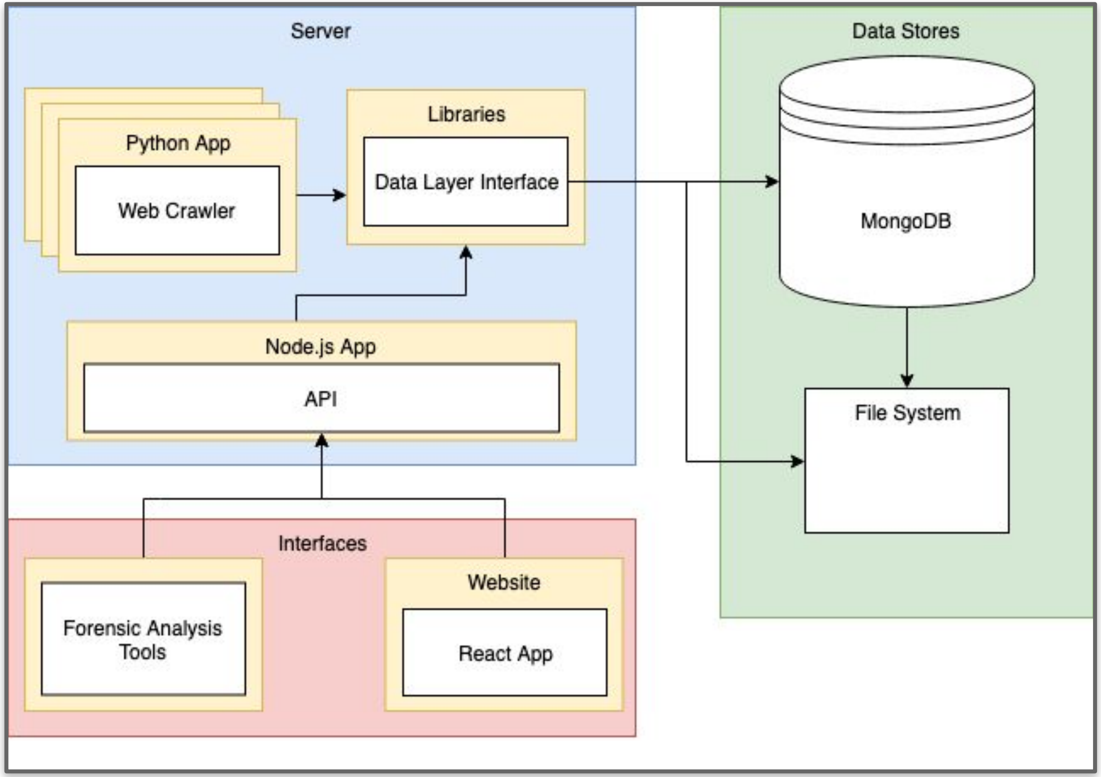


Milestones

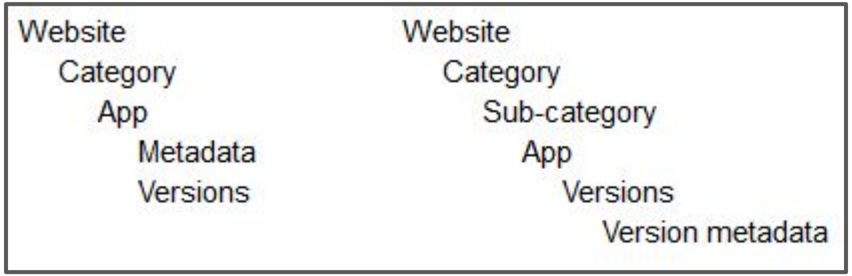
1. Develop Baseline Crawlers
2. Begin Collecting App Data
3. Design Web Tool
4. Integrate Additional Functionality into Web Tool

System Design

Detailed Design

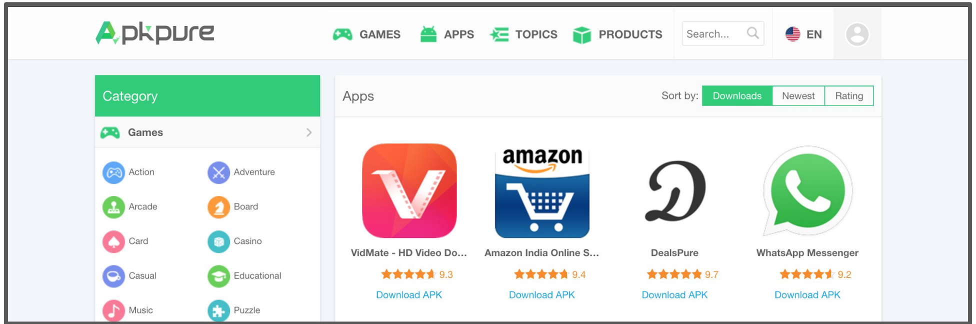


Detailed Design – Crawler



- Website
  - Category
    - App
      - Metadata
      - Versions

- Website
  - Category
    - Sub-category
      - App
        - Versions
          - Version metadata



APKPure

GAMES | APPS | TOPICS | PRODUCTS | Search... | EN

Category

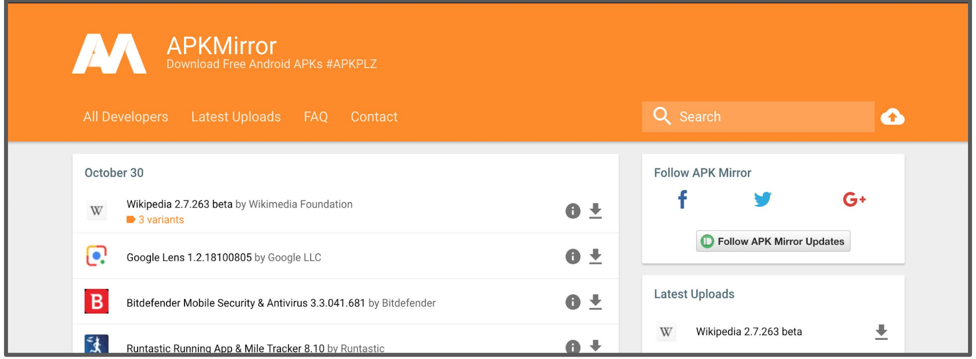
Games

Action, Adventure, Arcade, Board, Card, Casino, Casual, Educational, Music, Puzzle

Apps

Sort by: Downloads | Newest | Rating

VidMate - HD Video Do...	Amazon India Online S...	DealsPure	WhatsApp Messenger
★★★★★ 9.3	★★★★★ 9.4	★★★★★ 9.7	★★★★★ 9.2
Download APK	Download APK	Download APK	Download APK



APKMirror
Download Free Android APKs #APKPLZ

All Developers | Latest Uploads | FAQ | Contact | Search

October 30

- Wikipedia 2.7.263 beta by Wikimedia Foundation — 3 variants
- Google Lens 1.2.18100805 by Google LLC
- Bitdefender Mobile Security & Antivirus 3.3.041.681 by Bitdefender
- Runtastic Running App & Mile Tracker 8.10 by Runtastic

Follow APK Mirror

Follow APK Mirror Updates

Latest Uploads

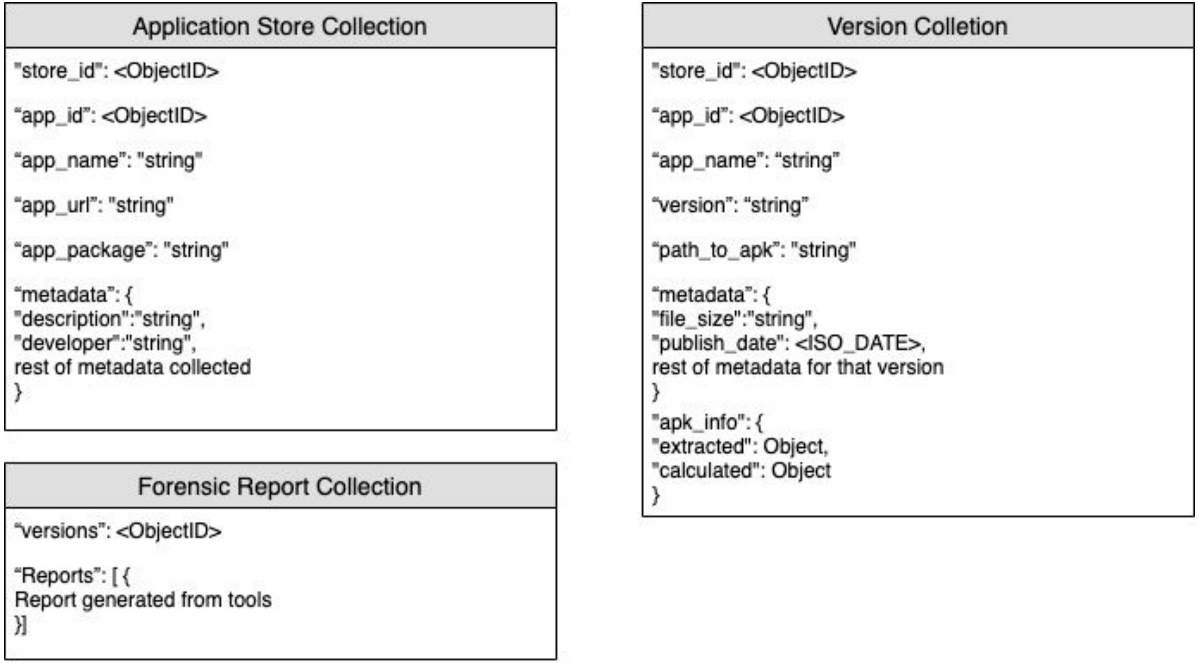
Wikipedia 2.7.263 beta

Metadata Collected (Not all inclusive)

- App Name
- Package Name
- Developer
- Hash Values
- Description
- Apk File
- App Version

Detailed Design - Database Model

- **3 Collections**
  - Application Store
  - Version
  - Forensic



Application Store Collection

```
"store_id": <ObjectID>

“app_id”: <ObjectID>

“app_name”: "string"

“app_url”: "string"

“app_package”: "string"

“metadata”: {
"description":"string",
"developer":"string",
rest of metadata collected
}
```

Forensic Report Collection

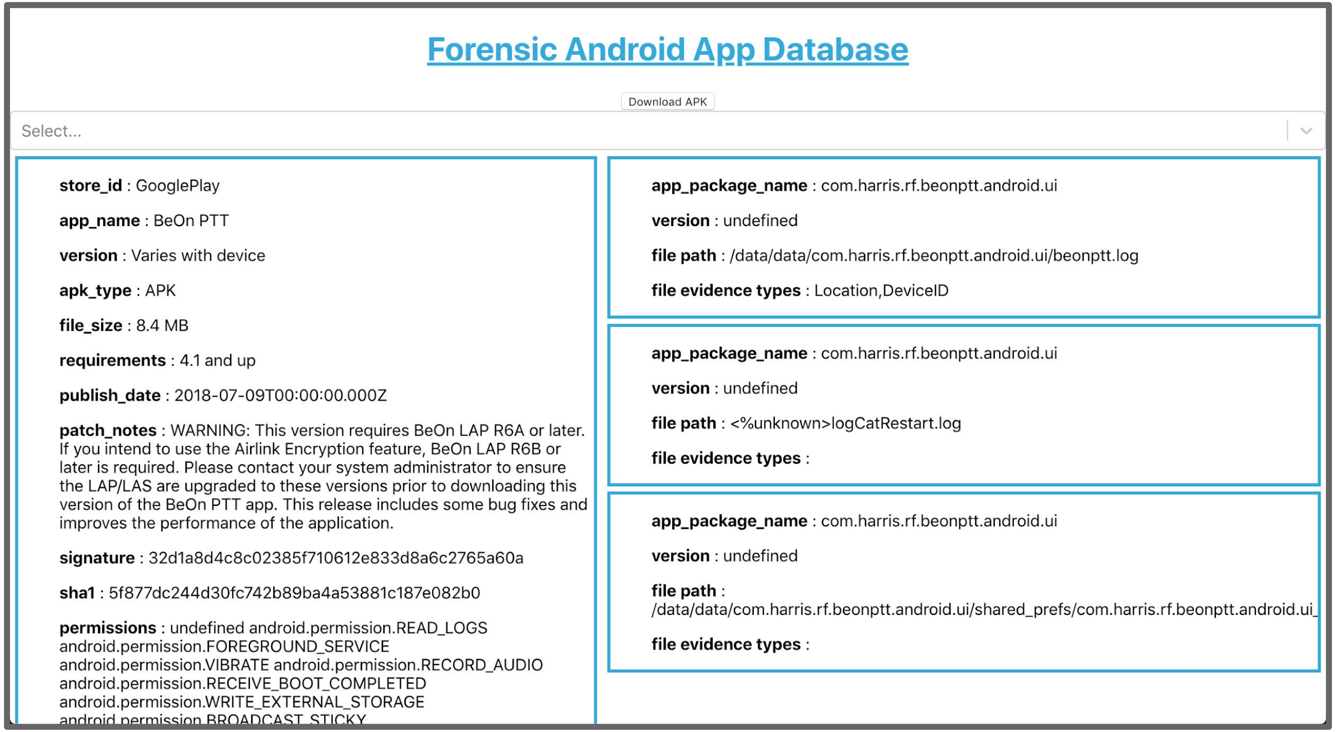
```
“versions”: <ObjectID>

“Reports”: [ {
Report generated from tools
}]
```

Version Colletion

```
"store_id": <ObjectID>

“app_id”: <ObjectID>

“app_name”: “string”

“version”: “string”

“path_to_apk”: "string"

“metadata”: {
"file_size":"string",
"publish_date": <ISO_DATE>,
rest of metadata for that version
}
"apk_info": {
"extracted": Object,
"calculated": Object
}
```

Detailed Design – Website / UI

- React
  - Javascript
  - Components
  - HTTP requests



Forensic Android App Database

store_id : GooglePlay

app_name : BeOn PTT

version : Varies with device

apk_type : APK

file_size : 8.4 MB

requirements : 4.1 and up

publish_date : 2018-07-09T00:00:00.000Z

patch_notes : WARNING: This version requires BeOn LAP R6A or later. If you intend to use the Airlink Encryption feature, BeOn LAP R6B or later is required. Please contact your system administrator to ensure the LAP/LAS are upgraded to these versions prior to downloading this version of the BeOn PTT app. This release includes some bug fixes and improves the performance of the application.

signature : 32d1a8d4c8c02385f710612e833d8a6c2765a60a

sha1 : 5f877dc244d30fc742b89ba4a53881c187e082b0

permissions : undefined android.permission.READ_LOGS android.permission.FOREGROUND_SERVICE android.permission.VIBRATE android.permission.RECORD_AUDIO android.permission.RECEIVE_BOOT_COMPLETED android.permission.WRITE_EXTERNAL_STORAGE android.permission.BROADCAST_STICKY

app_package_name : com.harris.rf.beonptt.android.ui

version : undefined

file path : /data/data/com.harris.rf.beonptt.android.ui/beonptt.log

file evidence types : Location,DeviceID

app_package_name : com.harris.rf.beonptt.android.ui

version : undefined

file path : <%unknown>logCatRestart.log

file evidence types :

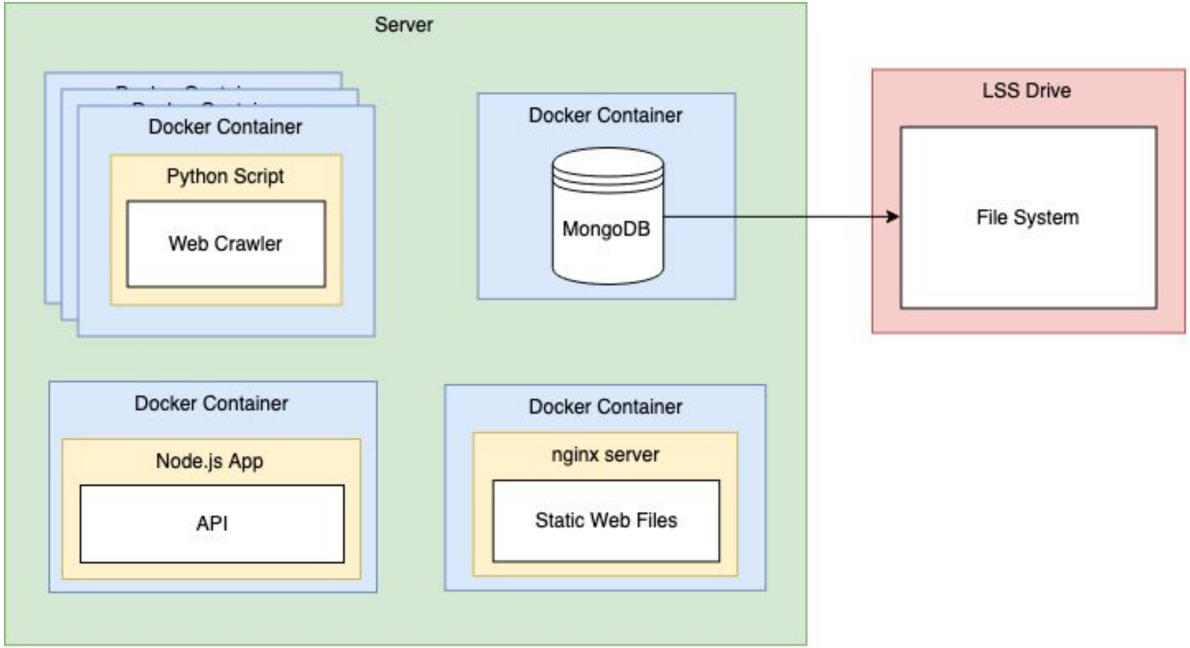
app_package_name : com.harris.rf.beonptt.android.ui

version : undefined

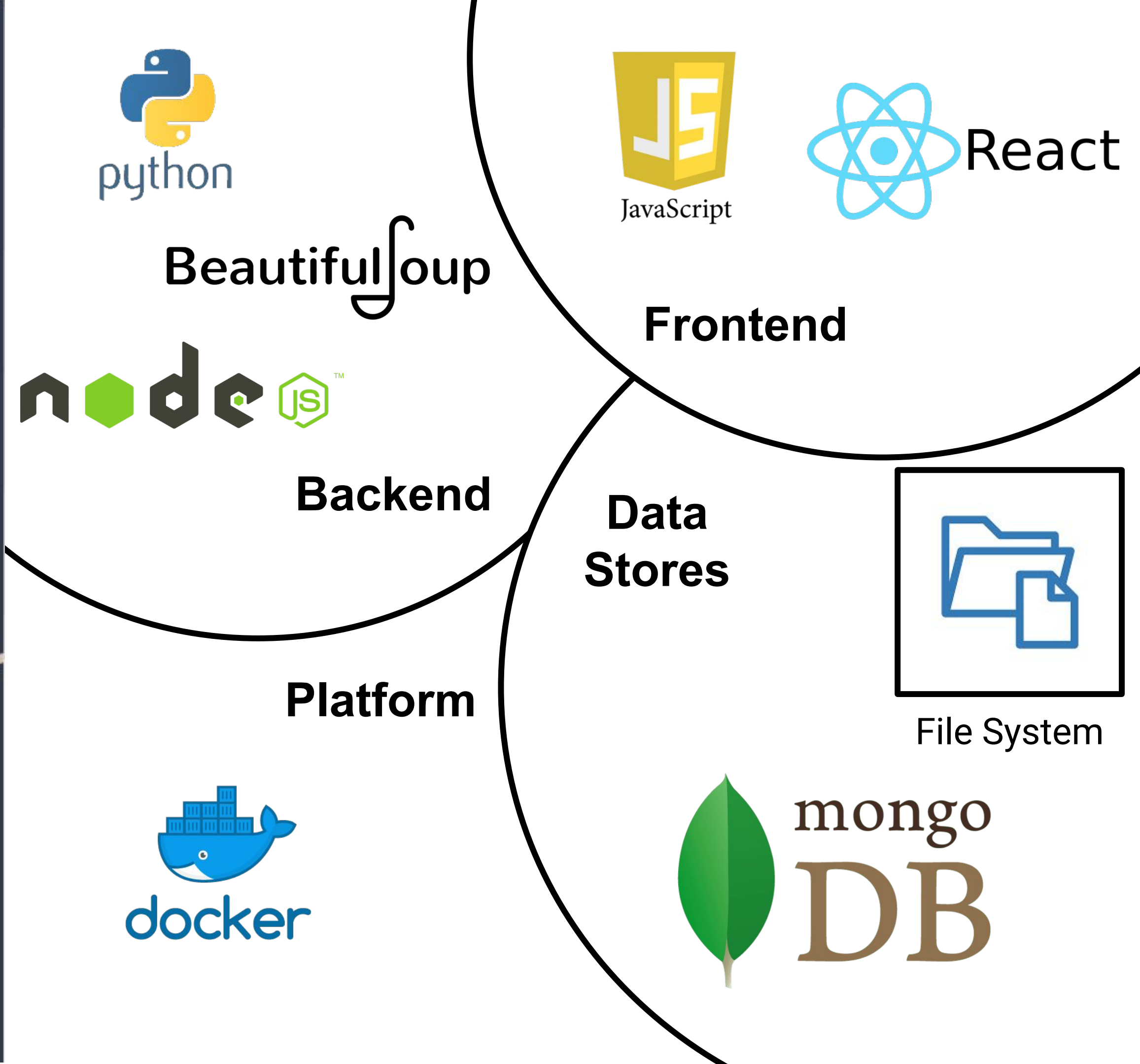
file path : /data/data/com.harris.rf.beonptt.android.ui/shared_prefs/com.harris.rf.beonptt.android.ui_

file evidence types :

Detailed Design – Implementation Diagram



Utilized Platforms and Technologies



Backend

Frontend

Data Stores

Platform

File System

Functional Testing

Unit Testing:

Web Crawlers

Backend

Integration Testing:

Web Crawlers → Database:

User API → Database

Database → File System

System Testing

Conclusion

Project Status

- **7 Completed Crawlers**
  - APKPure
  - APKMirror
  - UptoDown
  - F-Droid
  - Aptoide
  - Slideme
  - Google Play
- **Working database, frontend and backend.**
- **Mentions**
  - IEEE Symposium for Security & Privacy
  - Houston Forensic Science Center
  - AAFS in Baltimore

Member Contributions

Connor - Crawler Implementation

Emmett - System and Database design. Database Backend and Docker implementation

Jake - Crawler Implementation

Matt - Crawler and Frontend Implementation

Mitch - Crawler Implementation

Next Steps

- Continue developing additional crawlers to support more stores
- Continue to refine web portal for users to access and filter the information
- Set up production website for targeted user access
- Add paid applications
- Implement more security
- Collect reviews

Thank you for your time. Questions?