# Project 11 FIT File Handling and Data Pipeline

## Project Team

Mark Telley

**Company Redback Operations**

# Project Overview

The FIT File Handling and Data Pipeline project aims to handle FIT files from the Wahoo KICKR Live, convert them to CSV format, and upload the data to a database. It will aim to provide real-time performance metrics through a rudimentary user interface using basic JS, HTML, and CSS as an MVP (only data points). The project will offer guidance to the web/application team on integrating the data within the game experience. A Python script will communicate with the KICKR, download, and convert FIT files to CSV, and integrate the data into the data warehouse project for storage. The project will provide a comprehensive solution for handling KICKR Live FIT files, making the data easily accessible for analysis and real-time performance metrics.

What is the project about?

What problem is the project solve?

* Efficiently handles FIT files from the Wahoo KICKR Live.
* Converts FIT files to CSV format to support analysis and data manipulation.
* Unlocks the ability to upload the converted data to a database for storage and accessibility.
* Supports the ability to develop user interface to display real-time performance metrics (see all the other T2 projects)
* Offers guidance to the web/application team on integrating the FIT file data within the game experience.
* Streamlines the process of retrieving, converting, and analysing FIT file data.
* Eliminates the need for manual data extraction and conversion.
* Enables users to monitor their performance during workouts and make informed training decisions + allows the data science and AI team to conduct analysis etc.
* Provides a comprehensive solution for handling and integrating KICKR Live FIT files.

What are the aims of the project?

Aims for Trimester

* Develop a Python script for communicating with the Wahoo KICKR Live via Bluetooth connectivity and Wahoo API, downloading, and converting FIT files to CSV, and uploading data to a database.
* Create a rudimentary user interface using basic JS, HTML, and CSS to display real-time performance metrics as an MVP.
* Provide guidance to the web team on integrating the data within the game experience.

What are the deliverables?

Long-term Deliverables:

* A comprehensive solution for handling KICKR Live FIT files, ensuring easy accessibility for analysis and real-time performance metrics.
* Implementation of a scalable and secure data pipeline that seamlessly integrates with the game experience.

Trimester Deliverables:

* Development and completion of a Python script capable of downloading, converting, and uploading FIT file data utilising the Wahoo API and Bluetooth connectivity with the KICKR Live.
* Creation of a rudimentary user interface as a minimum viable product (MVP) to display real-time performance metrics.
* Documentation providing clear instructions on the usage and deployment of the Python script.
* Provision of guidance to the web team on effectively integrating the FIT file data within the game experience, including instructions on how to access the data through the implemented data pipeline.

# User Manual

Three key elements make up the Project: Setting up a GCP and BigQuery environment, connecting to Bluetooth devices such as but not limited to Wahoo’s Kickr bike trainer and connecting to Wahoo via API to retrieve information.

## 1.1. GCP & BigQuery

The GCP project provides a sandbox environment and essential datasets to support various projects. From handling FIT files and corporate reporting to sentiment analysis and user ranking, the project covers a wide range of data-related activities. It also facilitates access to app analytics data, demographics analysis, fitness data, user data, and Wahoo Kickr data. Also note original fitness data is stored securely in Google Cloud Storage buckets. It's an opportunity for the team to explore and innovate while paving the way for future developments in data management and analytics.

*Refer to the readme.MD for comprehensive instructions.*

[*https://github.com/redbackoperations/data-analysis/blob/main/Trimester\_1\_2023/Project%2011%20FIT%20File%20Handling%20and%20Data%20Pipeline/GCP\_Bigquery/GCP\_BigQuery\_Documentation.md*](https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2011%20FIT%20File%20Handling%20and%20Data%20Pipeline/GCP_Bigquery/GCP_BigQuery_Documentation.md)

## Direction Connection to Wahoo Kickr

The BLE Cycling Power Data Collection sub project is a python script that captures cycling power data from a Bluetooth Low Energy (BLE) device and saves it in a CSV file. Utilising libraries like asyncio, bleak, and pandas, the script provides a straightforward process for data collection. After setting up the necessary libraries, connecting to the BLE device (Wahoo Kickr), and specifying the session length, running the script captures data such as timestamps, power output, energy accumulation, pedal power balance, torque, wheel and crank revolutions, force and torque magnitudes, as well as dead spot angles. The collected data is stored in a pandas dataframe and saved in a CSV file. Documentation also focuses on the use of the PyCycling package as a valuable resource for cycling-related data analysis.

*Refer to the readme.MD for comprehensive instructions.*

<https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2011%20FIT%20File%20Handling%20and%20Data%20Pipeline/Kickr_Connection/README.md>

## Direction Connection to Wahoo Kickr

The Wahoo API/FIT Handling is a comprehensive solution that utilises the Wahoo API to facilitate interactions with Wahoo products and services. It offers several functionalities to enhance the integration and handling of FIT files:

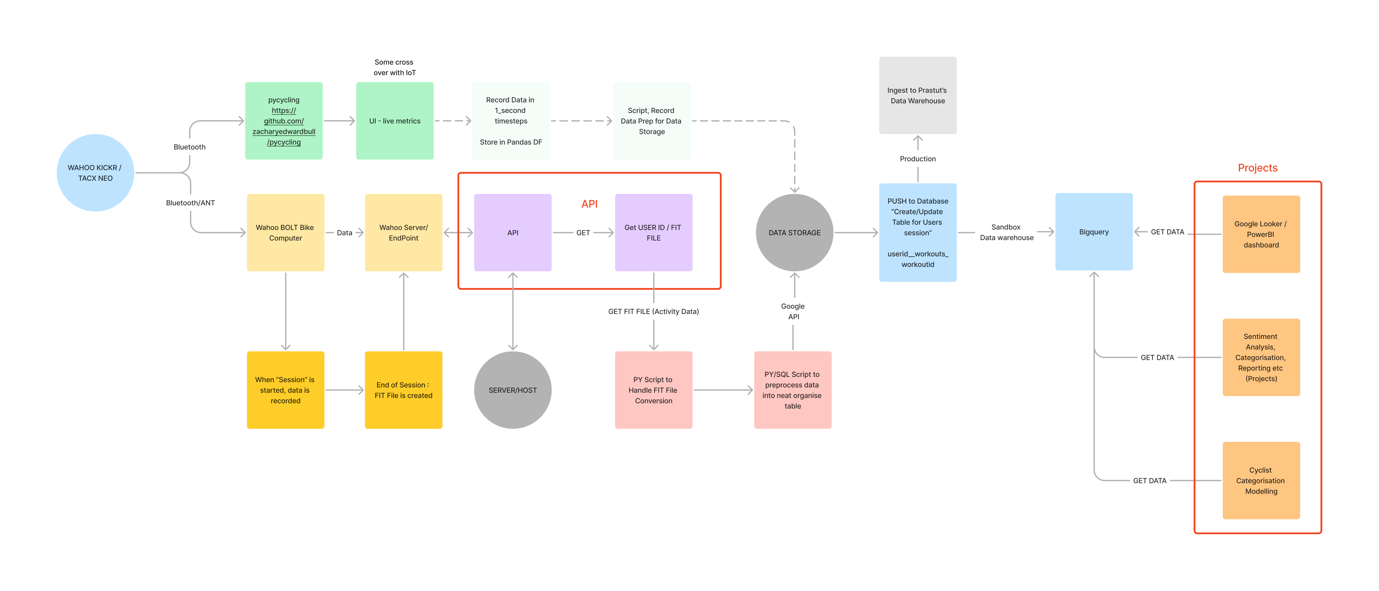
1. Authentication: The API allows developers to authenticate with Wahoo by providing the necessary client ID and client secret. These credentials can be obtained from the Wahoo Developer Portal.
2. User Details: The API enables the retrieval of user details, providing access to relevant information associated with the authenticated user.
3. Select Workout: Users can select specific workouts using the API, making it convenient to fetch and work with targeted exercise data.
4. Workout Summary / FIT File: The API allows users to retrieve workout summary information and obtain the associated FIT file. FIT files are widely used in the fitness industry to store detailed workout data, and accessing this information opens possibilities for analysis and further processing.
5. FIT Conversion and Handling: The script includes functionality to convert and handle FIT files. By using the "fitparse" and "csv" libraries, FIT files can be transformed into CSV format, allowing for easier manipulation and analysis of the data.
6. Data Warehouse / Table: The script also provides an **example** of integrating the FIT file data into a data warehouse or table. Documentation demonstrates connecting to a data warehouse, creating a table, inserting data, and closing the connection. This functionality streamlines the process of storing and organising the FIT file data for future analysis and retrieval.
7. API Endpoints: The Wahoo API offers various endpoints that facilitate communication with the Wahoo system. These endpoints include authentication, user details retrieval, workout listing, workout summary retrieval, and more.
8. Security Considerations: The script emphasises the importance of client secret security, ensuring that sensitive information is handled with care and stored in secure locations. Documentation discusses token refresh mechanisms to ensure seamless authentication without compromising user data.
9. Database Handling: While providing an example of data insertion into a SQL Server database, the script acknowledges the need to address potential SQL injection issues and handle data types properly. Adhering to best practices, such as using parameterised queries and data type considerations, is crucial when interacting with databases to ensure data integrity and security.

Full Documentation: <https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2011%20FIT%20File%20Handling%20and%20Data%20Pipeline/Wahoo_Cloud_API/readme.md>

## Pipeline Ideation

The below aims to provides an initial data pipeline that could service the Data/AI team:

* The Blue = GCP Environment with a future to switch to the data warehouse (3.1).
* The Green = Directly connecting to the Bluetooth device (3.2).
* The Yellow / Purple / and Red = FIT File and Wahoo API handling (3.3).



# Completed Deliverables

**GCP**

* Set up a Google Cloud Account and established a sandbox BigQuery database for project use.

**Wahoo Device Connection**

* Successfully recorded data from a Wahoo Kickr during a live session
* Thoroughly documented the steps involved and tested code.

**Wahoo API and FIT File Handling**

* Obtained a Wahoo API development account for the data pipeline project.
* Developed a working Wahoo API to retrieve information.
* Finalised the FIT handling script, ensuring efficiency and error-free operation.
* Thoroughly documented the steps and tested code.

Refer to the GITHUB LOCATION FOR ALL UPDATES: <https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%2011%20FIT%20File%20Handling%20and%20Data%20Pipeline>

# Roadmap

1. Collaboration with IoT Team:
   * Goal: Retrieve user-generated data from IoT devices.
   * Milestone: Enable the generation of FIT files upon exercise completion.
   * Action: Work closely with the IoT team to establish seamless data integration and mirror Wahoo's CLOUD API data model.
2. Integration with other Redback Teams and support data and analytical requirements:
   * Goal: Utilise the retrieved data from different teams to drive analytics and visualisation efforts across the entire company.
   * Milestone: Incorporate the IoT-generated data into the existing project framework Project 11)
   * Action: Align with the to identify the specific data requirements and adapt the project scope accordingly.
3. Completion and Deployment of Data Pipeline:
   * Goal: Establish a robust and efficient data pipeline using the Data Warehouse.
   * Milestone: Successfully deploy the pipeline for seamless data processing and storage.
   * Action: Collaborate with the other teams to design and implement the pipeline architecture, ensuring scalability, security, and efficiency.
4. Expansion of Data Sources:
   * Goal: Capture additional data sources to enrich the project insights.
   * Milestone: Incorporate data from app/website usage, corporate reporting, social interactions, and user ranking.
   * Action: Engage with respective teams (e.g., UX, Cyber, Mobile) to define data collection mechanisms, integrate APIs or data connectors, and design data processing workflows.
5. Refinement of the Project Roadmap:
   * Goal: Align the project with Trimester 2 objectives and priorities.
   * Milestone: Evaluate the project roadmap and adapt it to future goals.
   * Action: Engage in strategic discussions with stakeholders to identify Trimester 2 project requirements, dependencies, and resources.

By following this roadmap, we can as a team and company aim to enhance our data capabilities by incorporating real user-generated data, expanding the scope of our analytics and visualisation project, and establishing a robust data pipeline that supports other teams in the Company. This roadmap promotes cross-team collaboration and ensures that the project remains aligned with the evolving goals and objectives of Redback.

# Open Issues

**Wahoo developer Account**

. Formal application from Deakin needs to be confirmed/completed.

. Data model from IoT side of things needs to be established to mirror Wahoo’s.

# Lessons Learned

1. Data Model Agreement:

Lesson: It is crucial to establish a clear and agreed-upon data model between IoT devices and external platforms like Wahoo or Garmin.

Recommendation: Future teams should prioritise defining and aligning the data model early on to avoid compatibility issues and ensure seamless data integration.

1. Streamlined Communication Channels:

Lesson: Effective communication channels are vital for project coordination and progress tracking.

Recommendation: Future teams should establish streamlined communication channels to facilitate regular updates, feedback sharing, and efficient collaboration. Utilise tools like project management software and regular meetings to keep the team informed and aligned.

1. Thorough Documentation:

Lesson: Comprehensive documentation is essential for knowledge transfer and project continuity.

Recommendation: Future teams should prioritise thorough documentation of project scope, technical details, and any challenges encountered. Documenting processes, code repositories, and project resources will aid future team members and ensure smooth transitions.

1. Embrace Agile Methodologies:

Lesson: Agile methodologies enable adaptability and iterative development.

Recommendation: Future teams should consider adopting agile project management practices, such as scrum or Kanban, to promote flexibility, regular feedback, and continuous improvement. Agile approaches help teams navigate changing requirements effectively.

1. Testing and Quality Assurance:

Lesson: Early testing and quality assurance are crucial for identifying and addressing issues promptly.

Recommendation: Future teams should emphasise early testing and quality assurance throughout the development process. Implement automated testing frameworks, conduct regular code reviews, and involve stakeholders in the validation process to ensure high-quality deliverables.

# Product Development Life Cycle

Team Collaboration: We prioritise working as a team to deliver tangible value to the project and the company.

Scheduled Stand-ups: We conduct two weekly stand-up meetings (start, and end of the week) to synchronise our progress and address any issues or additional tasks required for the project. Ad hoc discussions also take place via the Teams app for quick decision-making.

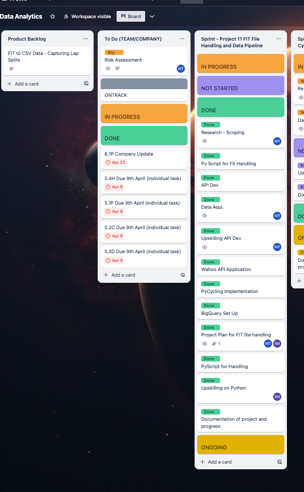
Task Planning and Progress Tracking: We utilise Trello boards to plan and track our tasks, updating our progress accordingly. We frequently create pull requests (PRs) and ensure timely merging by the team lead.

## New Tasks

We come up with new tasks along the way while we are working on existing planned tasks or from each stand-up meeting time. Any new tasks will be created in the [Trello](https://trello.com/login?returnUrl=%2Fb%2FNSuF3z83%2Fdata-analytics) board.

## Definition of Done

A DoD list is normally clearly defined in each Trello card, so the card assignee will be able to know exactly when a task is treated as completed by meeting all the DoD items. Additionally, we also have different status labels on each task on the Trello board to indicate their completeness.



## Task Review

All tasks’ updates are reviewed by the Team Lead prior to being committed/merged in the Github Repo

## Testing

Testing was conducted manually – this involved creating test cases and working through them. All task work provided has been tested, and retested.

## Branching Strategy

We never directly push any changes into the company’s `main` branch. To make any changes, we either create a new branch based off the latest `main` branch or fork the company’s `main` branch into our own repo. After we’ve finished the changes, we create a PR against the company’s `main` branch, and have it reviewed and merged by the team lead. We also ensure to resolve conflicts (if there’s any) before merging back to the latest `main` branch.

# Product Architecture

## UML Diagram

Refer to point 3.4 Pipeline Ideation

## Tech Stack

|  |  |
| --- | --- |
| Google Cloud Platform Tutorial: From Zero to Hero with GCP | . Google BigQuery  . Google Storage Cloud (Buckets)  . Google Looker Studio (Visualisation)  . Google Colab (Python Coding) |
|  | . Wahoo Cloud API |
| GitHub Logos and Usage · GitHub | . Version control  . Project management and documentation |
| Code Languages: | . SQL  . Python  . C (Curl request) |

# Source Code

Refer to the following links  
  
FIT Handling and API:

<https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2011%20FIT%20File%20Handling%20and%20Data%20Pipeline/Wahoo_Cloud_API/Wahoo_FIT_Handling.ipynb>

Bluetooth Connection:

<https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%2011%20FIT%20File%20Handling%20and%20Data%20Pipeline/Kickr_Connection/Wahoo_Kickr_Connection>

<https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2011%20FIT%20File%20Handling%20and%20Data%20Pipeline/Kickr_Connection/Wahoo_Kickr_Connection/main.py>

# Login Credentials

**BigQuery / GCP refer to project** [**documentation**](https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2011%20FIT%20File%20Handling%20and%20Data%20Pipeline/GCP_Bigquery/GCP_BigQuery_Documentation.md)**:**

To gain access to the GCP Project / BigQuery, please contact your team lead. They will need to coordinate with:

* Scott Blackburn (Senior Technical Officer, Cloud Computing & AI, School of Information Technology)
* GCP Project: SIT-23t1-fit-data-pipe-ee8896e

**Wahoo Developer:**

**Request Use of the Cloud API**

The Cloud API uses the public Wahoo server and authorised user data. Because of this, Wahoo Fitness is currently limiting the use of the API to those who request it, as well as providing more information about the scopes involved and the purpose of the application.

When you apply request to Wahoo, it will show up on your Developer Portal as pending approval. Be sure to include as much information as you can about your application so Wahoo can be confident in approving your use of the Cloud API.

*Refer to project documentation* [*here*](https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%2011%20FIT%20File%20Handling%20and%20Data%20Pipeline/Wahoo_Cloud_API)

# Appendices

Refer to the Github Repo for all project documentation.

<https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%2011%20FIT%20File%20Handling%20and%20Data%20Pipeline>