# Proposal: Migrating from .NET API to FastAPI with Redis

## 1. Introduction

This document outlines a proposal to replace the current .NET backend API with a modern and lightweight FastAPI backend in combination with Redis for improved performance, scalability, and cost-efficiency.

## 2. Why FastAPI?

FastAPI is a modern, high-performance web framework for building APIs with Python. It is ideal for building scalable and efficient APIs quickly and securely.

Benefits of FastAPI:

* - Asynchronous support (fast, non-blocking I/O)
* - Automatic interactive API documentation (Swagger, ReDoc)
* - Easy to develop and maintain
* - Python ecosystem integration (pandas, numpy, ML libraries)
* - Strong community and growing popularity

## 3. Why Redis?

Redis is an in-memory data structure store commonly used as a cache. It's used to reduce database load and improve response times.

Benefits of Redis:

* - Fast access to frequently used data
* - Reduces calls to BigQuery or other data sources
* - Helps avoid repeated processing for same input
* - Minimizes cloud billing and improves performance

## 4. Proposed Architecture

1. UI (Angular or existing .NET Frontend)  
2. FastAPI backend with REST endpoints  
3. Redis for caching results  
4. BigQuery or other data sources

## 5. Benefits of Migrating from .NET to FastAPI

* - Reduced development time and easier debugging
* - Improved performance for I/O-bound operations
* - Better integration with Python-based data tools
* - Lower hosting and maintenance costs
* - Scalability using asynchronous capabilities

## 6. Redis in This Context

Every user request is first checked in Redis. If cached, response is sent immediately. Otherwise, Python script fetches from BigQuery, result is cached in Redis and returned to user.

## 7. Summary

Migrating to FastAPI with Redis will make the backend more efficient, scalable, and easier to maintain. This will also reduce the load on BigQuery, lowering cloud costs and improving user experience.