# HassFramework Suggested Improvements

## Gemini Pro 1.0

// Improved Error Handling with more specific errors

enum HassError: Error {

case invalidURL

case networkError(Error) // Capture underlying network errors

case decodingError(Error)

// ...other errors

}

// Dependency Injection for better configuration

class HassRestClient {

private let baseURL: URL

private let session: URLSession

private let authToken: String

private let decoder: JSONDecoder

private let encoder: JSONEncoder

init(baseURL: URL, authToken: String, session: URLSession = .shared,

decoder: JSONDecoder = .init(), encoder: JSONEncoder = .init()) {

// ...

}

// Example using async/await

async func fetchDeviceState(deviceId: String) async throws -> DeviceState {

// ... use URLSession in an async manner

let data = try await session.data(for: request)

let deviceState = try decoder.decode(DeviceState.self, from: data)

return deviceState

}

}