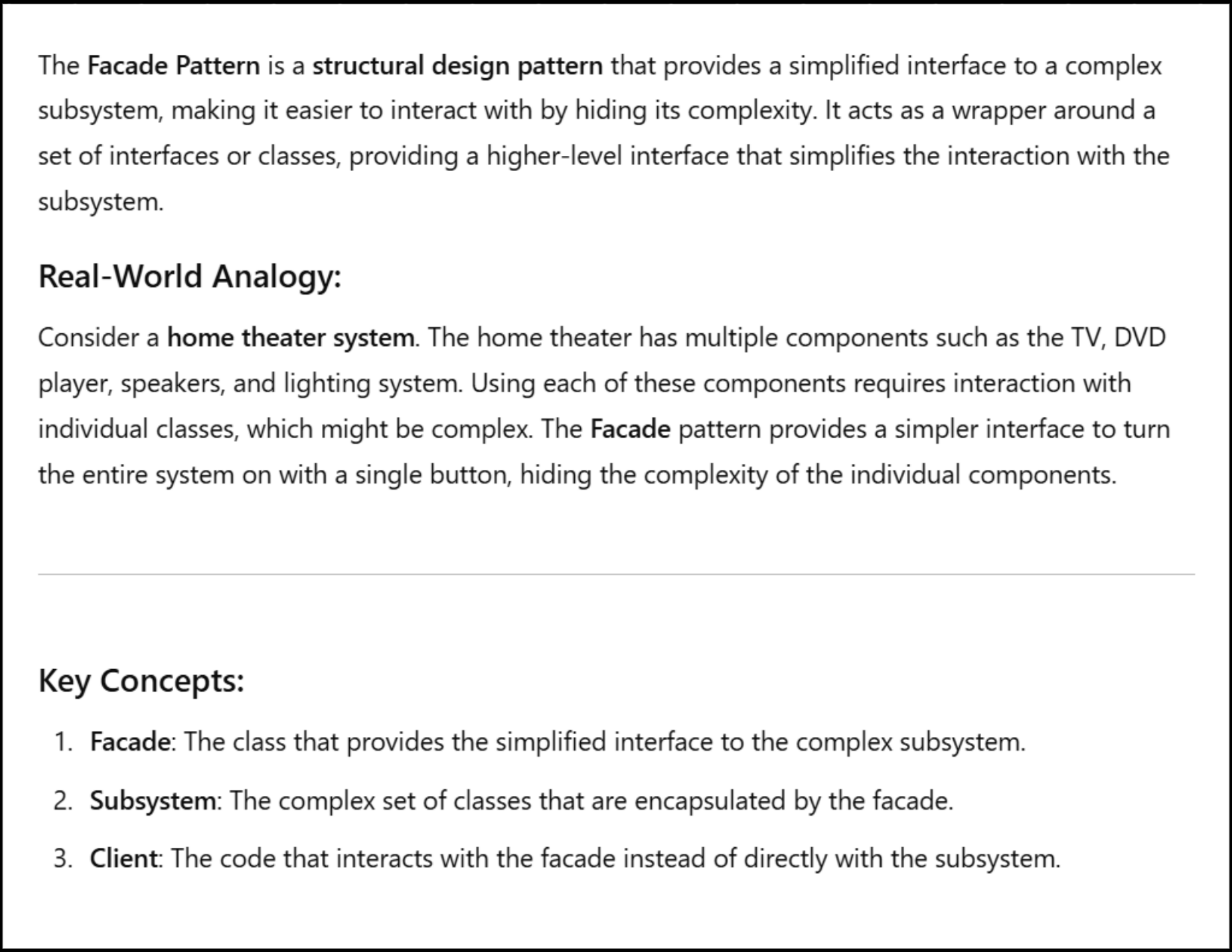
[Facade Design Pattern]



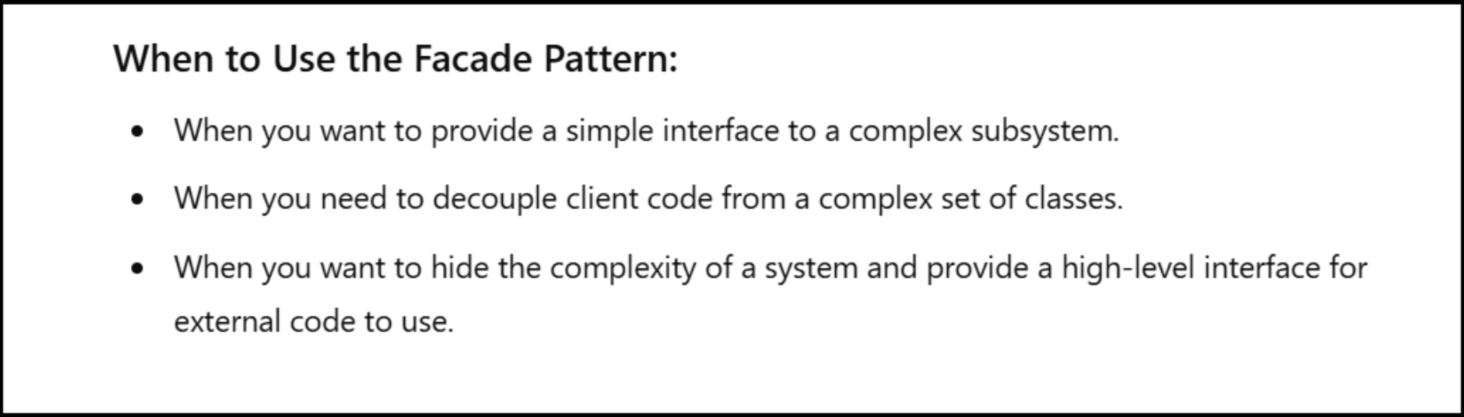
The **Facade Pattern** is a **structural design pattern** that provides a simplified interface to a complex subsystem, making it easier to interact with by hiding its complexity. It acts as a wrapper around a set of interfaces or classes, providing a higher-level interface that simplifies the interaction with the subsystem.

Real-World Analogy:

Consider a **home theater system**. The home theater has multiple components such as the TV, DVD player, speakers, and lighting system. Using each of these components requires interaction with individual classes, which might be complex. The **Facade** pattern provides a simpler interface to turn the entire system on with a single button, hiding the complexity of the individual components.

Key Concepts:

1. **Facade**: The class that provides the simplified interface to the complex subsystem.
2. **Subsystem**: The complex set of classes that are encapsulated by the facade.
3. **Client**: The code that interacts with the facade instead of directly with the subsystem.



When to Use the Facade Pattern:

- When you want to provide a simple interface to a complex subsystem.
- When you need to decouple client code from a complex set of classes.
- When you want to hide the complexity of a system and provide a high-level interface for external code to use.