Architectural Pattern vs Design Pattern

Before the comparison let us define them first.

Architectural Pattern

An **Architectural Pattern** expresses a fundamental structural organization schema for software systems. It provides a set of predefined subsystems, and their responsibilities, and includes rules and guidelines for organizing the relationships between them.

Design Pattern

In software engineering, a design pattern is a general repeatable solution to a commonly occurring problem in software design. A design pattern isn't a finished design that can be transformed directly into code. It is a description or template for how to solve a problem that can be used in many different situations.

There is a lot of confusion around what is the difference between Architectural pattern and Design pattern since they are closely related. Some of the common questions which we encounter are: Are they same or different? And if different, how different are they? Also some more questions arise like how they both fit in a project and then what is App Architecture?

Let’s talk about these and understand similarities and differences:

1. Architectural patterns are broader in scope than Design patterns. Design patterns provide very specific software related tasks whereas Architectural patterns are solutions for business problems. In other words Architectural pattern focuses more on the abstract view of idea while Design pattern focuses on the implementation view of idea.

2. Implementation of Design patterns are defined at granular level whereas Architectural patterns are defined at high level. For example different implementations of Factory or Builder patterns might look very similar in different projects. But the same Architectural pattern can vary a lot in different projects.

3. One Architectural pattern can be implemented by using many design patterns. There is one to many relationship between architectural pattern and design pattern.
   For example an implementation of MVP can be done in the following manner: Models can be built by using Factory and Builder patterns. Presenter can be built using the Observer and Facade pattern. Views can be built using Factory and Singleton.

   4. Architectural pattern is not App Architecture. Architectural pattern provides guidelines and rules to make application more maintainable, loosely coupled and extensible at project/solution level. Design patterns also do to some extent but more at module or component level. While App Architecture is complete Architecture of App which will have used Architectural pattern along with other design patterns and interfaces.

   5. Architecture : how components should behave and communicate in the system, set the physical location of components and finally choose the tools in order to create components. **Design** : while architecture deals more with the wide picture, design should drill down into the details relating to implementing certain components. Designing of components ends up with classes, interfaces, abstract classes and other OO features in order to fulfill the given component tasks.

   6. Architecture comes in the Designing phase and Design Patterns comes in the Building phase.

7. Architectural pattern is like a blueprint and design pattern is actual implementation.

Some popular Architectural patterns are:

- MVC
- MVP
- MVVM
- VIPER
- Three-tier/Multi-tier
- Dependency Injection Architectural pattern

There are 23 popular design patterns which are also known as Gang of Four (GOF) and divided into three categories: Creational Pattern, Structural Pattern and Behavioural Pattern.

Conclusion

Architecture

1. Architecture is the overall structure of software.
2. Architecture is the structure of the software system in its entirety.
3. Developer chooses different design patterns according to the architecture specification and requirement.
4. It defines the granularity of the component.

Design pattern

1. Design patterns are concerned with how the components are built.
2. It’s about a particular solution.