# Terminology

1. Cache Method.

# Chapter 13 Caching Data

1. A close-up of a text

   Description automatically generated
2. Children ask the same questions again and again.
3. Application consisting of components and If components are stateless, they will ask same question.
4. Reaching to the same answer may require fetching data from DB and then revoke a remote service or perform a calculation.  
   That is time and resources spent arriving at the answer.  
   **If answer is not likely to change frequently ( or at all)**, then it is wasteful to go through the same channel to fetch it again.
5. **Caching** is a way to store frequently needed information so that it’s readily available when needed.
6. In this chapter, we look at **Spring Cache Abstraction**.  
   Although **Spring doesn’t implement a cache solution**, it offers **declarative support** for caching that integrates with several popular **caching implementations**.

## 13.1 Enabling Cache Support

1. **Spring** **Cache Abstraction** comes in two forms.
   1. **Annotation-Driven Caching**.
   2. **XML-Declared Caching**.
2. The most common way to use **Spring’s Cache Abstraction** is to annotate methods with **@Cacheable** and **@CacheEvict.**  
   Then in section 13.3, we will look at **how to declare cache boundaries in XML**.
3. Before we can apply **Caching Annotation** in our beans, we need to **enable Spring’s support for annotation-driven caching.**If we are using **Java Configuration**, we can **enable annotation-driven caching** by adding **@EnableCaching to one our configuration classes**.
4. d

### 13.1.1 Configuring a Cache Manager