# Why do we have an extra service layer at the Application Scope?

1. **Separation of Concerns**:

* The Infrastructure layer's CachedFormRequestService handles the low-level caching mechanics and Cosmos DB interactions
* The Application layer's FormRequestCachingService provides a higher-level abstraction specifically for the application's needs
* This separation makes the code more maintainable and testable

1. **Application-Specific Caching Logic**:

* The Application layer service can implement caching strategies that are specific to the application's use cases
* It can handle application-specific cache key patterns and invalidation rules
* It can add application-specific metadata (like the CachedResponse wrapper)

1. **Domain-Driven Design (DDD) Principles**:

* The Application layer is responsible for orchestrating the use cases
* By having the caching service here, we can better control how caching interacts with our application's business logic
* It allows us to implement caching strategies that align with our domain requirements

1. **Flexibility and Extensibility**:

* If we need to change caching strategies or add new caching features, we can do it at the Application layer without touching the Infrastructure layer
* We can easily add new caching patterns or modify existing ones without changing the underlying infrastructure

1. **Testing and Mocking**:

* Having the caching service in the Application layer makes it easier to test application-specific caching behavior
* We can mock or substitute different caching implementations for testing purposes