# Article information:

Implementing Clean Architecture — To use or not to use MediatR? | by Plainionist | Medium
[https://medium.com/@plainionist/implementing-clean-architecture-to-use-or-not-to-use-mediatr-c16e93766de9](https://medium.com/%40plainionist/implementing-clean-architecture-to-use-or-not-to-use-mediatr-c16e93766de9)

# Article summary:

1. MediatR is a popular library in .NET used to decouple components, but its usage in Clean Architecture can violate the Dependency Rule.

2. The classic approach of providing adapters to integrate third-party libraries without violating the Dependency Rule is not easy to realize in the case of MediatR.

3. Reinventing the wheel by solving the problem without MediatR at all is also an option, especially for smaller projects where a subset of the full functionality of a library or framework is used.

# Article rating:

May be slightly imbalanced: The article presents the information in a generally reliable way, but there are minor points of consideration that could be explored further or claims that are not fully backed by appropriate evidence. Some perspectives may also be omitted, and you are encouraged to use the research topics section to explore the topic further.

# Article analysis:

The article "Implementing Clean Architecture — To use or not to use MediatR?" by Plainionist on Medium discusses the usage of the popular .NET library, MediatR, in Clean Architecture projects. The author questions whether using MediatR violates the Dependency Rule and explores alternative solutions.

The article provides concrete examples of how MediatR is used in Clean Architecture projects, such as dispatching commands and publishing domain events. The author visualizes the dependencies between classes and concludes that using MediatR violates the Dependency Rule in both cases.

The author acknowledges that some software engineers may argue for a pragmatic approach and accept MediatR as part of the application logic. However, the author presents reasons why they personally would decide differently, such as the potential for breaking changes or end-of-life for third-party libraries like MediatR.

The article suggests two options to avoid violating the Dependency Rule: using adapters or finding an alternative library that does not require markup interfaces. The author explores several alternative libraries but notes that they still require implementing certain library interfaces.

Finally, the article suggests "reinventing the wheel" as a last option to solve problems without using MediatR. The author provides examples of how to implement alternatives using request-specific interfaces or event buses.

Overall, the article presents a well-reasoned argument against using MediatR in Clean Architecture projects. However, it could benefit from exploring counterarguments more thoroughly and presenting both sides equally. Additionally, some claims made by the author are unsupported by evidence or missing points of consideration. For example, while it is true that third-party libraries can experience breaking changes or end-of-life, it is also true that maintaining custom code can be costly and time-consuming.

# Topics for further research:

* Pros and cons of using MediatR in .NET projects
* Dependency Rule in Clean Architecture
* Alternative libraries to MediatR for dispatching commands and publishing events
* Adapters in software design
* Request-specific interfaces in software design
* Event buses in software design

# Report location:

<https://www.fullpicture.app/item/ebb325d14cec31f7de24a34399e59b54>