

Table of Contents

Advantages and Disadvantages of Layered Architecture	3
When to Use Layered Architectures	3
Advantage and Disadvantage of MVC	3
Layers	4
Tiers.....	5
Advantages and Disadvantages of Web Services	5
Advantages and Disadvantages of SOAP Web Services	6
Difference between REST and RESTful:	6
When to use REST?	7
Advantages and Disadvantages of REST Web Services.....	7
Advantages of REST API:	7
Disadvantages or Challenges in REST:	7
Advantages and disadvantages of Spring Boot.....	8
Difference Between @Controller and @RestController	8
Internationalization	10
Advantages and disadvantages of i18n	10
Content negotiation.....	10
Advantages and disadvantage of DevTools	10
Layered Architecture:	11
Tier Architecture	11
MVC:	12
Web Services:	12
SOAP	12
REST.....	13
Spring Boot.....	13
Internationalization I18N	14
Spring Data JPA:.....	15
Spring JMS:	15
P2P Model JMS:	16
Pub-Sub Model JMS:	16
# Layered architecture advantage/disadvantage?	17

# Tiers advantage/disadvantage?	17
# MVC advantage/disadvantage?	17
# Web Services advantage/disadvantage?	18
# SOAP advantage/disadvantage?	18
# REST advantage/disadvantage?	19
# Spring boot (Spring boot starter project) advantage/disadvantage?	19
# InternationalizationI18N advantage-disadvantage	20
Spring Data JPA advantage-disadvantage?	20
# JMS advantage-disadvantage?	20
# P2P advantage-disadvantage?	21
# Pub-Sub model advantage/disadvantage?	21

Advantages and Disadvantages of Layered Architecture

The following are the benefits and drawbacks that exist with this software pattern:

Advantages

- The framework is simple and easy to learn and implement.
- There is reduced dependency because the function of each layer is separate from the other layers.
- Testing is easier because of the separated components; each component can be tested individually.
- Cost overheads are fairly low.

Disadvantages

- Scalability is difficult because the structure of the framework does not allow for growth.
- They can be difficult to maintain. A change in a single layer can affect the entire system because it operates as a single unit.
- There is interdependence between layers since a layer depends on the layer above it to receive data.
- Parallel processing is not possible.

When to Use Layered Architectures

When developing simple, small applications, it is advisable to implement a layered architecture because it's the most simple framework. However, some developers are of the opinion that because they can be difficult to maintain, it is better to apply them to larger projects.

In spite of this, the framework can be used for applications that need to be built quickly because it's easy to learn and implement. It is also good in cases where the developers do not have a lot of knowledge of software architectures or when they are undecided on which one to use.

Some real-life applications of this architecture are in web applications and the [OSI model](#). The [J2EE programming model](#) also implements the layered architecture.

Advantage and Disadvantage of MVC

Advantages

- A main advantage of MVC is separation of concern. Separation of concern means we divide the application Model, Control and View.
- We can easily maintain our application because of separation of concern.
- At the same time, we can split many developers work at a time. It will not affect one developer work to another developer work.

- It supports TTD (test-driven development). We can create an application with unit test. We can write won test case.
- Latest version of MVC Support default responsive web site and mobile templates.
- We can create own view engine. Its syntax is very easy compared to traditional view engine.
- Easier support for new types of clients
- Development of the various components can be performed parallelly.
- It helps you to avoid complexity by dividing an application into the three units. Model, view, and controller

Disadvantages of using MVC

- Difficult to read, change, unit test, and reuse this model
- The framework navigation can some time complex as it introduces new layers of abstraction which requires users to adapt to the decomposition criteria of MVC.
- No formal validation support.
- Increased complexity and Inefficiency of data
- The difficulty of using MVC with the modern user interface
- There is a need for multiple programmers to conduct parallel programming.
- Knowledge of multiple technologies is required.
- Maintenance of lots of codes in Controller

Layers

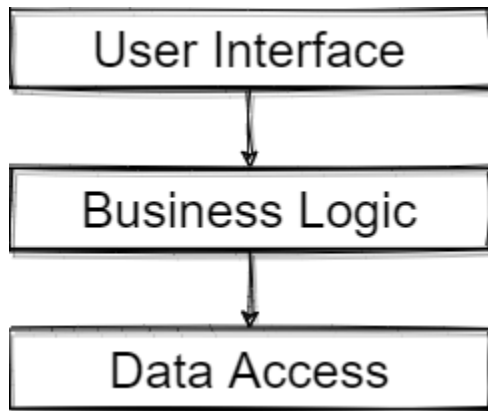
The term layer refers to a logical separation of code. In other words, it's a coherent set of related functionalities.

Structuring applications in layers is a way to organize our codebase better. For this reason, portability and maintainability are drivers that push us to use this kind of structure.

Indeed, we find an application of this concept in various types of architecture, such as *Layered Architecture* and *Hexagonal Architecture*.

As an example, the following is a representation of a layered architecture:

AD



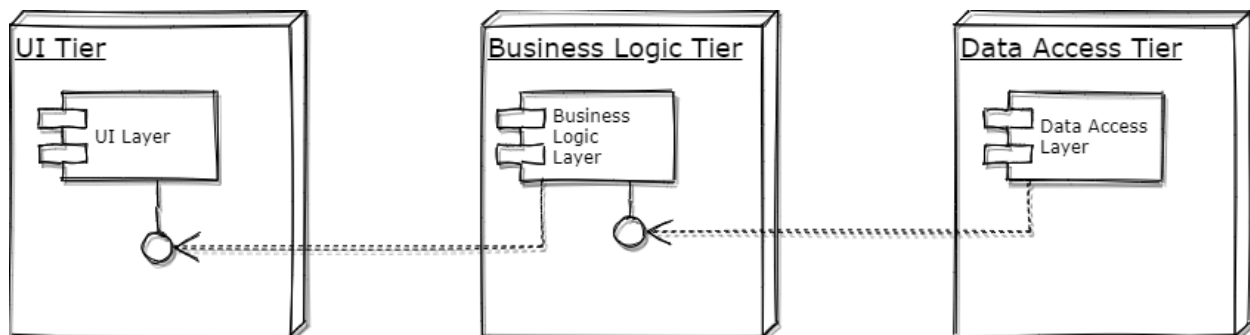
Here, the User Interface Layer, Business Logic Layer, and Data Access Layer each **have their own responsibility and are logically separated**.

Layers have nothing to do with their physical separation. When we talk about layers, we don't specify where we'll deploy them.

Tiers

On the other hand, the term tier refers to the physical structure. Tiers define where to deploy the layers, without necessarily a one-to-one mapping. So, when we talk of an application's tiers, we mean its topology.

As an example, let's see how we can physically divide the layered application of the previous paragraph:



Above, we see a representation of an *n-tier Architecture* where each tier physically hosts a layer. This is the classic *3-tier Architecture*.

Advantages and Disadvantages of Web Services

Advantages

- Web Services are standards-based.
- Web Services' loose coupling leads to increased modularity and flexibility.

- Web Services reduces integration costs.
- Web Services simplify Business to Business Integration (B2Bi).
- Web Services enable new business models.
- Web Services leverage existing technology and skill sets.

Disadvantages

- Web Services are immature.
- Every user or client who uses web services know that it is not available hundred percent all the time.
- Web services is not secure. Web services are available to public through http-based protocol. So, everyone can access web services and use it. This flaw can be avoided using authentication mechanisms.
- Guaranteed Execution is a major problem of web services because HTTP which is hypertext transport protocol is not a reliable protocol that is it does not provide any guarantee of delivery of response.

Advantages and Disadvantages of SOAP Web Services

Advantages

- SOAP web services can be written in any programming language and executed in any platform.
- Works well in distributed enterprise environments.
- Provides significant pre-build extensibility in the form of the Web Service Definition Language (WS*) standards.
- SOAP uses HTTP protocol for transport due to which it becomes scalable.

Disadvantage

- SOAP uses XML format that must be parsed to be read. It defines many standards that must be followed while developing the SOAP applications.
- It is slow and consumes more bandwidth and resource.
- SOAP uses WSDL and doesn't have any other mechanism to discover the service.
- SOAP restricted to the XML.

Difference between REST and RESTful:

REST web service is a Representational State Transfer and an architectural pattern for creating web services whereas the RESTful service is one that implements that pattern. So, there is no special difference in between. However, how good your architecture complies with an absolute standard but how well it tallies your needs & grows with your business.

When to use REST?

- **Limited bandwidth and resources:** SOAP messages are heavy in content and consume greater bandwidth, REST should be used where network bandwidth is the constraint.
- **Ease of coding:** coding REST service and implementing is easier than SOAP. So, if a fast solution is required for web services, then REST web service is the best way.
- **Cache:** if there is a need for cache and a lot of requests then REST is the perfect one. This increases the no of requests which are sent to the server. Implementing a cache, the most frequent queries results stored in an intermediate solution. Whenever the clients request for a resource, it checks the first cache. If resources exist, then it will not proceed to the server. Caching can help to minimize the number of trips.

Advantages and Disadvantages of REST Web Services

Advantages of REST API:

- REST API is easy to understand and learn, due to its simplicity, known API.
- With REST API, being able to organize complicated applications & makes it easy to use resources.
- The high load can be managed with help out of HTTP proxy server & cache.
- REST API is easy to explore and discover.
- It makes it simple for new clients to work on other applications, whether it is designed specifically for purpose or not.
- Use standard HTTP procedure call- outs to retrieve data and requests.
- REST API depends on codes, can use it to synchronize data with a website without any complications.
- Users can avail access to the same standard objects and data model when compared to SOAP-based web services.
- Brings flexibility formats by serializing data in XML or JSON format.
- Allows Standard-based protection with the use of OAuth protocols to verify your REST requests.

Disadvantages or Challenges in REST:

- **Lack of state:** most web applications require stateful mechanisms. Suppose you purchase a website which has a mechanism to have a shopping cart. It is required to know the number of items in the shopping cart before the actual purchase is made. This burden of maintaining the state lies on the client, which makes the client application heavy and difficult to maintain.
- RESTful web services are stateless.
- **Last of security:** REST does not impose security such as SOAP. That is the reason REST is appropriate for public URLs, but it is not good for confidential data passage between client and server
- REST have no standard definition language.

Advantages and disadvantages of Spring Boot

Advantages

- Spring boot works well with several servlet containers
- Bootstrapping saves memory space
- Decreased boilerplate code
- No XML configuration required
- A large community of helpful users
- Easy start
- Fast and easy development of Spring-based applications
- No need for the deployment of war files
- Helping to directly embed Tomcat, Jetty, or Undertow into an application
- Out of the box functionality

Disadvantages

- Large deployment files that result from unused dependencies
- The complex and time-consuming process of converting a legacy or an existing Spring project to a Spring Boot application.
- Its inability to build large, monolithic applications (although it works extremely well for developing microservices)
- Needing to rely on the Spring ecosystem instead of using other tools that many developers find advantageous
- Not suitable for large-scale projects
- Lack of control

Difference Between @Controller and @RestController

@Controller

@Controller is used to mark classes as Spring MVC Controller.

It is a specialized version of @Component annotation.

In @Controller, we can return a view in Spring Web MVC.

@RestController

@RestController annotation is a special controller used in RESTful Web services, and it's the combination of @Controller and @ResponseBody annotation.

It is a specialized version of @Controller annotation.

In @RestController, we cannot return a view.

@Controller

@Controller annotation indicates that the class is a “controller” like a web controller.

In @Controller, we need to use @ResponseBody on every handler method.

It was added to Spring 2.5 version.

@RestController

@RestController annotation indicates that class is a controller where @RequestMapping methods assume @ResponseBody semantics by default.

In @RestController, we don't need to use @ResponseBody on every handler method.

It was added to Spring 4.0 version.

Difference between Spring and Spring Boot:

S. Spring	Spring Boot
Spring is an open-source lightweight framework widely used to develop enterprise applications.	Spring Boot is built on top of the conventional spring framework, widely used to develop REST APIs.
The most important feature of the Spring Framework is dependency injection.	The most important feature of the Spring Boot Autoconfiguration.
It helps to create a loosely coupled application.	It helps to create a stand-alone application.
To run the Spring application, we need to set the server explicitly.	Spring Boot provides embedded servers such as Tomcat and Jetty etc.
To run the Spring application, a deployment descriptor is required.	There is no requirement for a deployment descriptor.
To create a Spring application, the developers write lots of code.	It reduces the lines of code.
It doesn't provide support for the in-memory database.	It provides support for the in-memory database such as H2.

Internationalization

Internationalization or I18N is a process that makes your application adaptable to different languages and regions without engineering changes on the source code. You can display messages, currencies, date, time etc. according to the specific region or language, likewise you can say internationalization is a readiness of localization.

Advantages and disadvantages of i18n

Advantages

- Increase your consumer base
- Increase internet users' contentment (user-experience)
- Enter new markets quickly and strategically
- Reduce your time, money, and effort
- All languages' source code is centralized, making future modifications simple
- The internet market's reach is rapidly expanding
- Use global marketing to your advantage

Disadvantages

Content negotiation

In [HTTP](#), *content negotiation* is the mechanism that is used for serving different [representations](#) of a resource to the same URI to help the user agent specify which representation is best suited for the user (for example, which document language, which image format, or which content encoding).

Advantages and disadvantage of DevTools

Advantages

- **Property Defaults:** spring-boot-devtools module automatically applies sensible development-time configuration by disabling properties such as caching to false during development; Sets DEBUG mode for web logging group which helps get more information about incoming request.
- **Automatic Restart:** Applications that use spring-boot-devtools restart automatically whenever files on the class path change. This can be a useful feature when working in an IDE, as it restarts automatically with the code changes.
- **Live Reload:** The spring-boot-devtools module includes an embedded LiveReload server that can be used to trigger a browser refresh when a resource is changed.

- **Global Settings:** You can configure global devtools settings by adding a file named spring-boot-devtools.properties to your \$HOME folder. Any properties added to this file apply to all Spring Boot applications on your machine that use devtools.
- **Remote Applications:** spring-boot-devtools provides out of the box remote debugging capabilities via HTTP, to have this feature it is required that spring-boot-devtools are packaged as part of the application. This can be achieved by disabling exclude Devtools configuration in the plugin in maven.

Disadvantage:

Layered Architecture:

Advantage:

- It provides modularity and helps to achieve separation of concern
- It is flexible, scalable and maintainable.
- Increases development as developer can work independently in each layer

Disadvantage:

- The more Layers you have the more risks you have for things to breakdown.
- It is not always possible to divide functionalities.
- No communication between non-adjacent layer.

Tier Architecture

Advantage:

- **Better Security:** Each tier operates on its own infrastructure, it's possible to enforce different levels of protection
- **Scalability:** As your organization grows You can scale up your DB-Tier with DB-Clustering without touching other tiers.
- **Maintainability:** Web designer can change the View-code, without touching the other layers on the other tiers.

Disadvantage:

- A larger number of tiers also means more networks and devices associated with the system, and this can result in slow performance of the application
- Every tier involved in a software application requires more hardware, deployment and maintenance needs, each of which increases cost.

MVC:

Advantage:

- It provides modularity and helps to achieve separation of concern
- It is flexible, scalable and maintainable.
- Increases development as developer can work independently in each layer

Disadvantage:

- We need extra overhead due to layers which will do negative impact on the performance.
- It adds complexity to simple application.

Web Services:

Advantage:

- It provides interoperability between machine-to-machine communication over a network.
- It is language independent way of communication. Java application can interact with other language .net, PHP.

Disadvantage:

- Availability. Web services cannot guarantee, it is present all the time.
- Security: Web services are available to the general audience through HTTP-based protocol.

SOAP

Advantage:

- No restriction on transport. It can be over http or queue.
- SOAP language is independent and can be written on a wide range of platforms.
- It provides robust security, SOAP's support for WS-Security can come in handy.

Disadvantages of SOAP.

- SOAP is known to be slow and consumes a lot of time. This is because it uses the XML format which has to be constantly gone through.
- It is limited to Web Service Description Language (WSDL) to operate.
- A restriction on the format of XML exchanged.

REST

Advantage:

- No strict data exchange format. However, JSON is popular.
- No Standard language required like WADL/Swagger.
- REST makes work easier for the developer as it allows him to give commands and wait for the work to be executed with so much ease.

Disadvantages of REST.

- It transports over Http only.
- It is hard to use as compared to other web services.
- The client may sometimes not be aware of the state of the web service which makes it difficult for him to manage.
- It doesn't have a built-in messaging system. If a communication fails, the client has to deal with it by retrying.

Spring Boot

Advantage:

- Auto configure which are ready for development and saves our effort.
- Starter projects with all dependencies along with proper version between related dependencies.
- Spring boot offers an effortless way to create spring-based applications.
- It reduced development time. Spring Boot minimizes the time spent in developing and increasing productivity.
- Also, it helps in reducing all the manual work of writing annotations, boilerplate codes and XML configurations.
- Developers have easy access to Embedded HTTP servers such as Jetty, Tomcat and also easily test the web applications effortlessly.

Disadvantage:

- Lack of control. Spring Boot creates a lot of unused dependencies, resulting in a large deployment file.

- The complex and time-consuming process of converting a legacy or an existing Spring project to a Spring Boot application.
- Spring boot is opinionated framework, so we mightn't have latest updated of some spring framework until spring boot is updated.

Internationalization I18N

Advantage:

- Higher quality software that meets the technical and cultural needs of multiple locales
- Reduced time, cost, and effort for localization (L10n)
- Single source code for all languages of the product
- Simpler, easier maintenance for future iterations of the product.

Disadvantage:

- If internationalization testing is not performed, it may have an adverse impact

Spring Data JPA:

Advantage:

- The Spring data JPA remove the boilerplate code and make implementation of persistence layer easier and fast.
- It provides set of repository interface which we only need to extend to define a specific repository one of our entities.
- It provides generation of database queries based on method names. As long as your query isn't too complex, you just need to define a method on your repository interface with a name that starts with find...By. Spring then parses the method name and creates a query for it.

Disadvantage:

- Application code will be couple to the library and to its specific abstractions.
- Spring Data JPA does not currently support dynamic sorting for native queries, because it would have to manipulate the actual query declared, which it cannot do reliably for native SQL
- Spring Boot employs many Template classes such as JDBC Template, JMS Template, etc., which provide high-level simplified APIs that perform complex boilerplate tasks in the background

Spring JMS:

Advantage:

- Due to asynchronous communication, the message queue of JMS provides high performance, high throughput.
- It provides loosely couple. The messaging application need not know each other for messaging.
- It is reliable as the Producer and Consumer do not have to be available at the same time.
- Scalability, flexible and agility

Disadvantage:

- JMS is Java-based.
- In multi-tiered applications using microservices, where multiple languages and frameworks are used, this can become a hindrance.

- In JMS, although APIs are specified, the message format is not. This is a limitation of JMS. They just have to use the same API.

P2P Model JMS:

Advantage:

- P2P guarantees that only one consumer will process a given message.
- P2P messaging can be used to provide the interoperability between these heterogeneous platforms
- This provides a Queue Browser that allows the JMS client to take a snapshot of the queue to see messages waiting to be consumed.

Disadvantage:

- There can be more than one consumer listening on a queue but only one of them will be get the message.

Pub-Sub Model JMS:

Advantage:

- At a time, message can be published to a multiple subscriber. It keep the copy of message and subscribe to all subscriber.
- reduces complexity by removing all the point-to-point connections with a single connection to a message topic, which will manage subscriptions to decide what messages should be delivered to which endpoints.
- Publishers and subscribers are decoupled and work independently from each other which allows to develop and scale them independently.

Disadvantage:

- Decreased the flexibility to modify publisher and structure of published data.
- It does not support synchronous end-to-end communications
- Implementing Pub/Sub will be a waste of resources and lead to unnecessary complexity for smaller systems where scalability and decoupled are not vital.

Layered architecture advantage/disadvantage?

Advantage:

- Layers are isolated from each other this provides flexible and maintainable architecture.
- Each layer separation also helps to test layer independently.
- Promotes faster development, as it allows for independent concurrent development of each layer.

Disadvantage:

- Increase complexity and risk, requires integration between layers and there's risk of having breakdown between them.
- Requires functionality distribution between layers and layers are dependent on the layers above it.
- Scaling may be difficult because the structure of the layer is inter-dependence on layers above and may not allow for growth.

Tiers advantage/disadvantage?

Advantage:

- Scalability, tiers work as independent infrastructure and can be scaled as required. we can horizontally or vertically scale it to increase performance and stability.
- Security and Performance, additional security can be added.

Disadvantage:

- Extra Resources and additional cost, physical separation requires extra additional resources and cost to maintain and upgrade.
- Integration time and complexity, large number of tiers means more system to integrate with and more time to work on.

MVC advantage/disadvantage?

Advantage:

- Allows faster development, as layers aren't inter-twined, and it achieves separation of concern.
- Promotes loose coupling, high scalability and single responsibility.
- More supportive of testable and flexible code.

Disadvantage:

- MVC pattern add abstraction between layers which increases complexity and can be difficult to debug and can impact performance.
- Increase time and complexity as more adaptable functionality needs to be introduced to middle layer to bridge a gap between layers.
-

Web Services advantage/disadvantage?

Advantage:

- Machine-to-machine interoperability and Language independent, allows systems to communicate and transfer data with each other despite difference in programming language.
- Uses Standard-based communication and is available over the web, this allows the system to expose the features over the web, which can be reusable to all.

Disadvantage:

- Security and reliability, web services are available to all and uses http protocols which passes through the general public route and data can be intercepted or blocked. And services may not be available all the time.
- Verbose and need to match the specific requirement. e.g., need to communicate via specific data representation (JSON or XML) and via specific protocols.
- They don't support bidirectional communication and complex distributed component; web services only know each other for short period of connection time and can't call client as needed.

SOAP advantage/disadvantage?

Advantage:

- SOAP is language independent, it's written program can be executed in any platform.
- Includes its own security as WS security.
- WSDL document provides contract and technical details, without exposing underlying implementation.
- Works well in distributed enterprise environments

Disadvantage:

- SOAP uses only xml format which is verbose and consumes lot of bandwidth and resource.
- Its WSDL dependent, requires communicating via its WSDL.

REST advantage/disadvantage?

Advantages:

- It supports multiple technologies, including popular JSON format, and is lightweight and faster.
- No contract is defined between server and client and so it promotes loosely coupled implementation.

Disadvantage:

- Supports only Http/Https, there can't be asynchronous calls
- Server and Client communication is stateless (aren't aware of each other state) and makes it difficult to communicate.
- No contract defined between service and client, and it has to be defined with documentation or other means WADL.

Spring boot (Spring boot starter project) advantage/disadvantage?

Advantage:

- Provides autoconfigure for the common dependencies, which saves time and configuration effort.
- Starter project came with all proper versioning and required dependencies.
- Starter project provides out-of-the-box functionality, comes with ready tools for starting and working with spring application. e.g., Data-JPA provides query templating out-of-box, starter-web provides embedded tomcat and configured web context
- Open source has large community and documents well.

Disadvantage:

- Uses opinionated dependency configuration, need to customize and configure, if we require other dependency.

- Spring boot takes away control. Spring Boot creates a lot of unused dependencies, resulting in a large deployment file.

Internationalization advantage-disadvantage

Advantage:

- Software can provide technical and cultural needs of multiple locales.
- Reduce time, cost and effort for adding and supporting new locales.
- Increase software consumer base and reach, this helps in expansion and growth.

Disadvantage:

- Increases the complexity and the need to add resource for storing, maintaining and translating locales.
- Requires combinatorial testing and verification for all added language.

Spring Data JPA advantage-disadvantage?

Advantage:

- Provides the templating and reduce boilerplate code.
- helps in speeding up the development process with the help of dynamic query creating.

Disadvantage:

- Takes up memory and increase time with method parsing and query conversion.
- Makes our application code coupled with library and to its specific abstraction

JMS advantage-disadvantage?

Advantage:

- Provides high performance and throughput with asynchronous mode of communication
- Promotes loose coupling architecture and is flexible and scalable.

- Decrease the application load overhead and acts as a middleware for applications to communicate reliably.

Disadvantage:

- Single point of failure, if a middleware is down, all services are interrupted.
- Application needs to come with a additional approach to ensure the message are transmitted securely and reliably to the targeted party.

P2P advantage-disadvantage?

Advantage:

- Suitable for the model of single Consumer per queue.
- Less overhead on single queue, needs to push message to only one consumer and the process completes faster.

Disadvantage:

- The consumer who comes first will get a message and one queue serves a purpose to only one consumer.

Pub-Sub model advantage/disadvantage?

Advantage:

- All subscribers can get a message and ensures multi message distribution.
- Topics can allow additional functionality to ensure what message to deliver to which endpoints
- Consumer and produces are loosely coupled, as producer-consumer has the ability to subscribe and bind to the type of message on topic

Disadvantage:

- Adds extra complexity, to keep track of subscribers, create message copy and send them to the subscriber endpoints.
- Requires extra memory usage and communication overhead slowing down the middleware.