



Benefits of ROCA

In this lesson, we'll discuss the benefits of ROCA.

We'll cover the following ^

- Clean architecture
- Web features can be harnessed
- Low bandwidth consumption
- High speed
- Resilience
- JavaScript not necessary
- ROCA versus SPAs
- Integration options

ROCA has a number of **advantages**:

Clean architecture

The applications have a *clean architecture*. The logic is on the server. Changes to the logic can easily be rolled out by a new server version.

Web features can be harnessed

All features of the web can also be used.

- URLs can be sent to other people (for example, by email) because they uniquely identify resources.
- HTTP caches can be used if, say HTTP GETs are not allowed to change data.
- Optimizations in the browsers are exploited. Browsers implement many tricks to show users the first parts of a website and to allow interactions as quickly as possible.

Low bandwidth consumption

#

Applications get by with *little bandwidth* as a result of usually having to transfer no more than HTML – and only for the web pages that are actually visited.

In an SPA, on the other hand, often the entire application must be transferred before any interaction is possible at all.

Modern SPA technologies optimize this by loading separate modules rather than all the code. However, initializing the application and making it react to user interaction still takes some time. This is easier with a simple web application.

Modern browsers are optimized to make user interaction with simple web applications as fast and responsive as possible.

High speed

The solution is *fast*, especially for mobile devices; the speed of JavaScript implementations often leaves much to be desired. ROCA applications require a minimum of JavaScript.

Resilience



Finally, an *error in JavaScript* or in the transfer of the JavaScript code due to a problem on the network only leads to hard-to-use, though still available applications. If logic were implemented in JavaScript, this would not be the case and the application would have less resilience.

JavaScript not necessary

Users with *JavaScript switched off* can still use the application. Nowadays, these users are practically non-existent so this advantage is irrelevant.

ROCA versus SPAs

ROCA is an alternative to SPAs (single page apps) where the browser is used only to execute JavaScript code.

- **ROCA takes advantage of the browser's optimizations** for HTML and HTTP, and also of the other advantages of a real web application.
- Of course, a ROCA system can also use JavaScript as discussed previously. The goal of ROCA is not to avoid JavaScript completely, but to **limit it** to areas where it makes sense or where there are no alternatives.
- **SPAs might be a better fit for complex UI** – for example, games or map applications on the web.
- However, for other types of applications, such as e-commerce applications, SPAs not only provide unneeded flexibility but make it harder to achieve some features easily, like search engine optimizations (SEO). For a lot of applications, SPAs are actually not that great a fit.



- SPAs seduce developers to implement **more logic on the client**. Although this might make the system more responsive, it can also lead to redundant implementation on the client and server that is harder to maintain.
- Also, it is hard to export the logic in the SPA as a REST service so that other clients can use it.

Chapter 4

(<https://www.educative.io/collection/page/10370001/5441945024331776/4860469534785536>) shows an example of a ROCA application that is quite comfortable to use.

Integration options

ROCA generally applies to good UI layer design. It makes sense as a guideline for implementing browser applications, monolithic or modularized.

However, ROCA does facilitate **simplified integration**. HTML is the focus in the frontend while JavaScript serves only to improve usability. Thus, the system can be modularized by providing HTML pages with links to other HTML pages that may originate from other microservices. Or, the HTML pages can even be composed of several parts. Each component can come from a different microservice, making it easy to modularize a ROCA UI.

Because ROCA supports all options for frontend integration, ROCA is a good basis for frontend integration and for a microservices architecture.



QUIZ

Z

1 How is less bandwidth consumed in a ROCA application?

- ☐ A) The entire app has to be transferred to the client's side beforehand.
- ☐ B) Only a bit of HTML needs to be retrieved from the server.
- ☐ C) The logic resides on the client, hence once the app is retrieved, no further requests need to be made.

Submit Answer



Question 1 of 3
0 attempted



Reset Quiz



In the next lesson, we'll discuss some challenges that frontend integration presents.

[← Back](#)[Resource-oriented Client Architecture ...](#)[Next →](#)[Challenges](#)[Mark as Completed](#)[Report an Issue](#)