

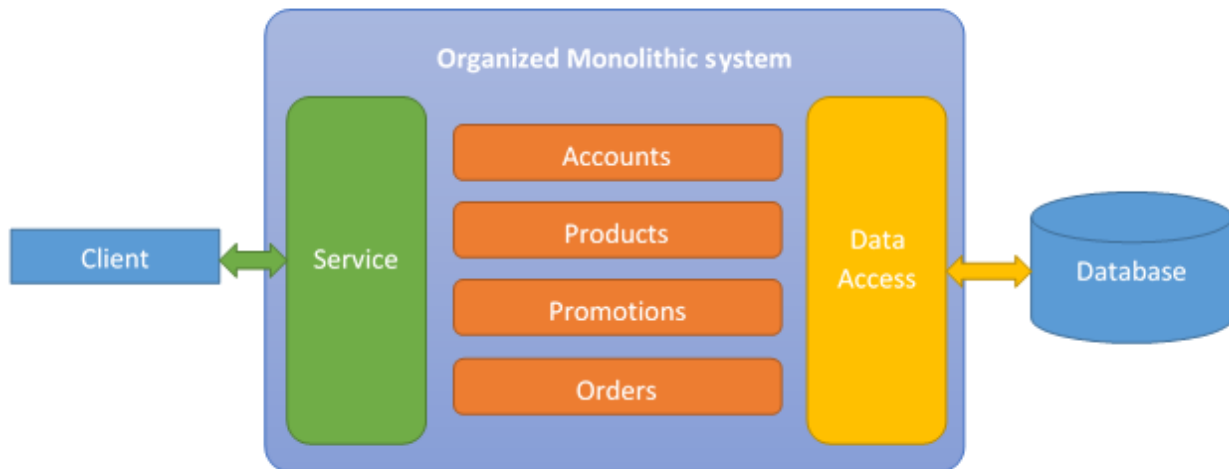
Microservices: Brownfield: Migration

 acarlstein.com/

Alejandro G. Carlstein Ramos Mejia

icon-arrow-left [Microservices: Brownfield: Introduction](#) | [Microservices: Brownfield: Migration: Database](#) *icon-arrow-right*

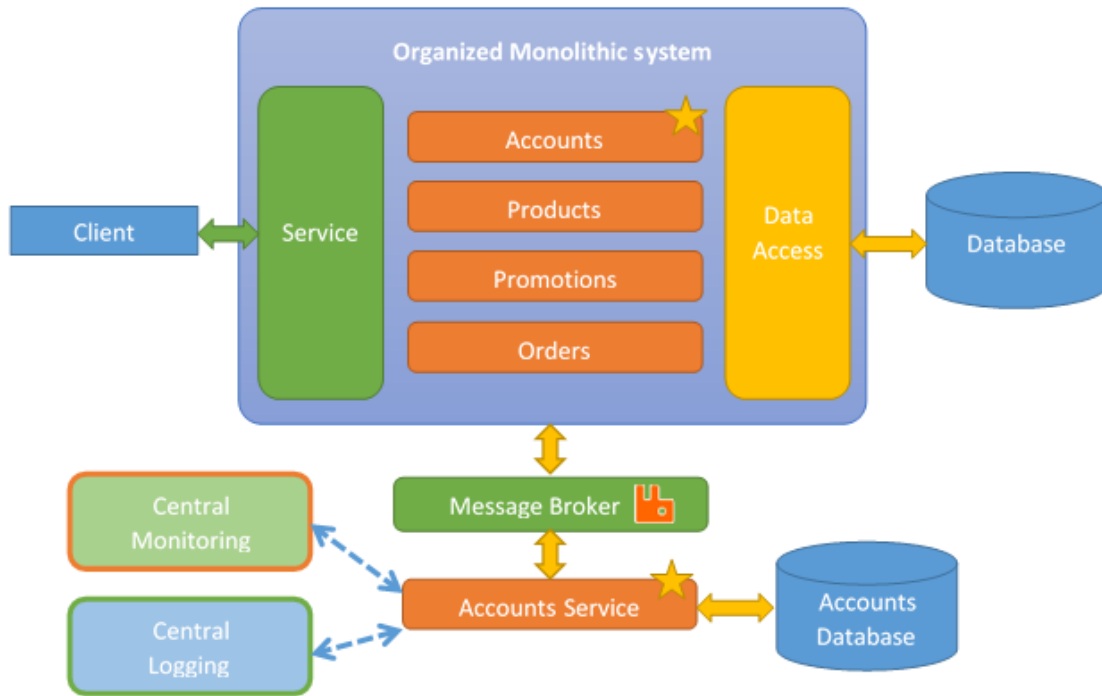
Assuming you have done the previous step of reorganizing your code, now can get ready for the migration to actual micro-services. Your monolithic architecture should be looking more or less as follow:



As you can see we have:

- Code grouped based on business domain or function.
- Clear boundaries with clear interfaces.

We begin by converting one of these groups into a micro-service. This will allow us to learn in progression of migration, the use of new technology, the use of distributed transactions, the implementation of separate data, and allow us to get comfortable with the whole process of migration.



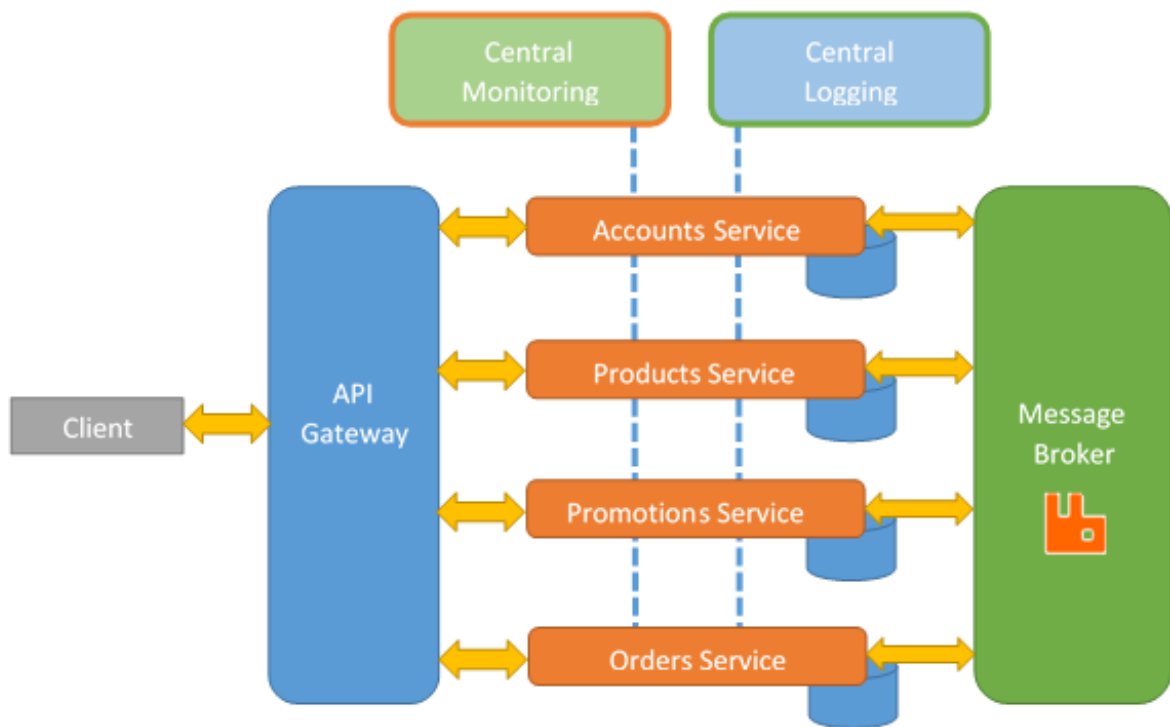
This provides an advantage since you can switch between the old functionality (in the monolithic architecture) and the new micro-service functionality. If you encounter any issue with the new micro-service functionality, you can switch back to the old functionality of the monolithic architecture. Note that you may have to keep maintaining the two variations of the same code until you only use the micro-service alone.

One thing that you may have to do is to prioritize what parts of the system you can migrate to a micro-service with the least impact. There are different ways to prioritize:

- By level of risk.
- By dependencies.
- By technology utilized.
- By complexity.

Since we are following an incremental approach when doing our migration, we can ensure that our micro-services are going to work with our existing monolithic system and that they will run alongside. Also, this helps with the learning curve since we are introducing a new way to work. Remember to keep monitoring both the monolithic system and the micro-services for any kind of impact. Always keep reviewing and improving your infrastructure in order to support the new distributed transaction done by the micro-services.

If we do everything well, we should be ending with our new micro-service architecture as shown below:



© 2017, Alejandro G. Carlstein Ramos Mejia. All rights reserved.

[+](#) Share / Save [f](#) [t](#) [↗](#)