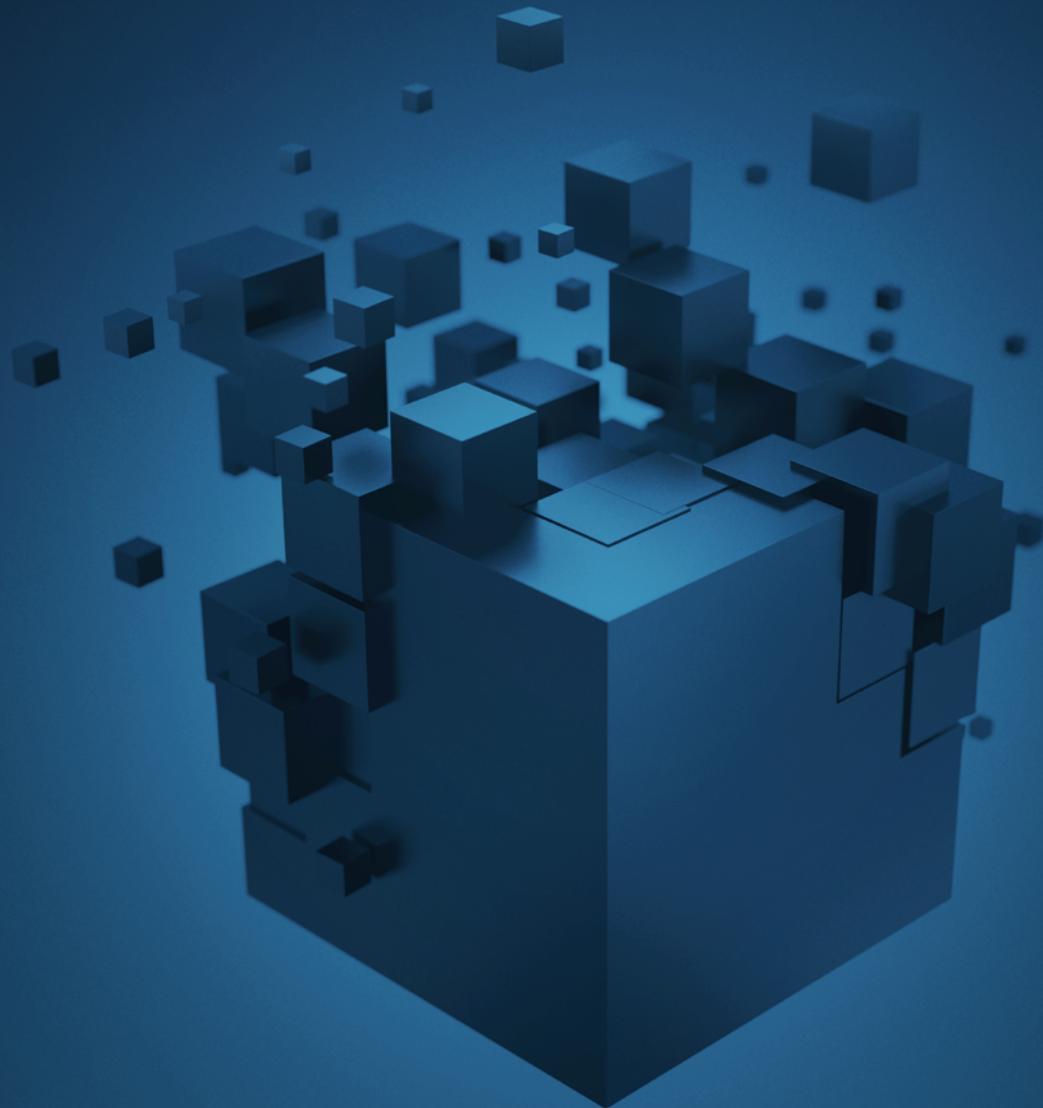


FROM IT MONOLITH TO MICROSERVICES:

Modulate Your Business and Guarantee Competitiveness in the Future



The experts' advice on how to break up your company's IT monolith to ensure competitiveness in the future

You are part of a large company with many years of experience and lots of favorable business – at home and perhaps even abroad.

Naturally, the company has an IT structure, and it probably has a few years behind it. The older it gets, the more complex and tangled-up it becomes. It is not uncommon to see IT systems that are 10, 15, or maybe 20 years old. Such a system is, in technical terms, called a monolith.

This IT structure can offhand seem favorable for the daily work life at the company. Everything is stored in the same huge database, which is easy for employees with high seniority to dive into.

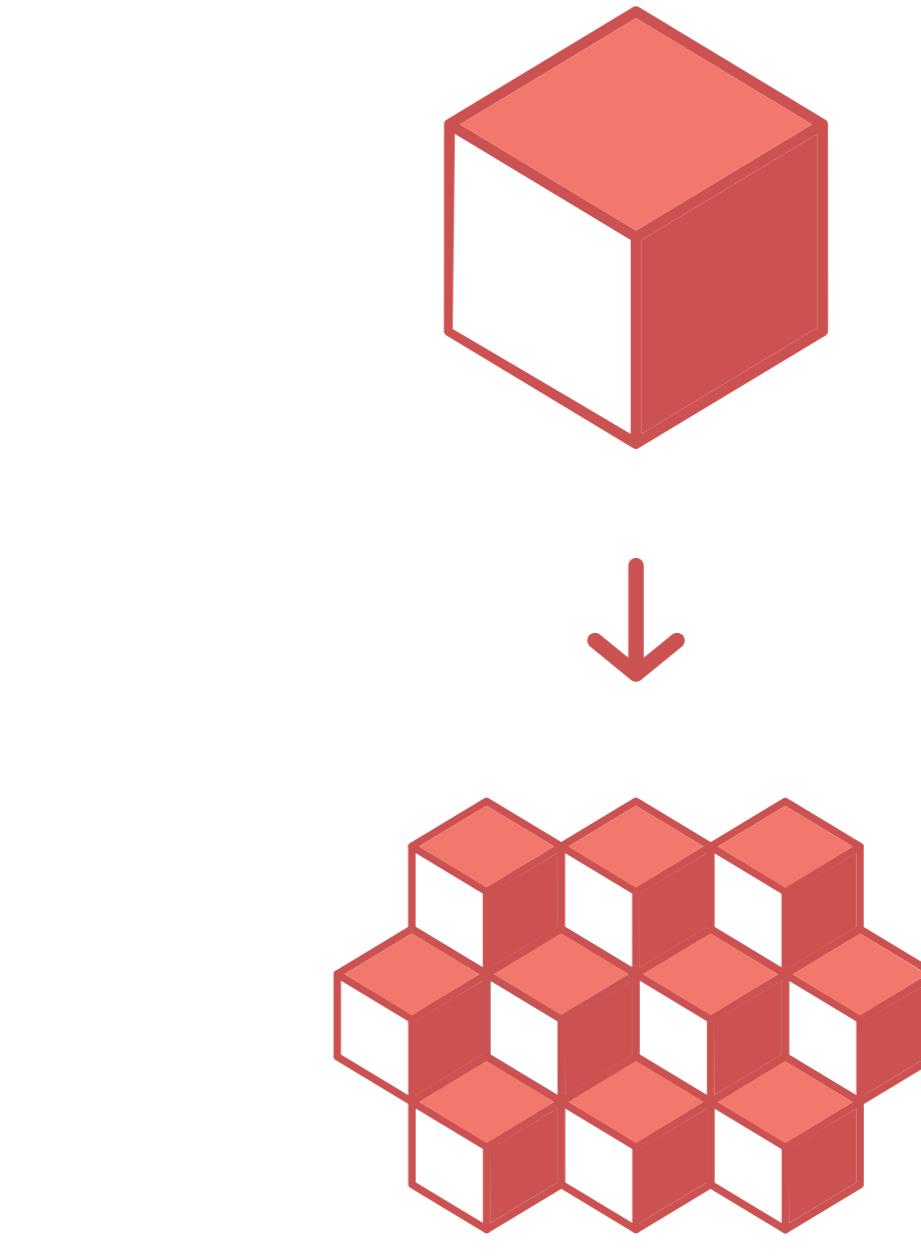
But the monolith is heavy, and it is built on increasingly old-fashioned technology. IT professionals are no longer being trained in the monolith's technology, and it is not attractive for the new IT generation to work with it. Every time an update is needed in one area of the monolith, your IT department is forced to update the entire system. This results in unnecessary

downtime in all system corners - not just in the required domain. This also means that adding new features and scaling is time consuming. Far more than it needs to be. Oftentimes, the result is so-called 'Shadow IT' (read: Excel sheets), which becomes important for the business. This is not desirable.

So it is expensive to maintain your monolith. It is difficult to upgrade. Heavy and costly to scale. Furthermore, it is difficult to recruit employees to operate it. It is a growing challenge to keep up with competitors (who may already be in the process of breaking down their monolith). The monolith is, in other words, not future-proof.

Many of our customers have been in the same situation as you. They are now on a journey towards a lighter and more service-based and modulated IT structure. They are on the journey towards microservices, and maybe even Cloud and far easier scalability throughout their business.

In this eBook, some of our senior software architects give insight into why it



is important for your business to start breaking down your monolith and, of course, how to get started.

The journey is not for everyone. One of the most important considerations before starting the microservice train is whether your organization is ready for the change, and how you prepare yourself for it. It is intrusive for the entire organization, and with a reorganization and modernization of the IT struc-

ture, the entire company becomes an IT organization. It requires a readiness for change. The payoff is an IT structure that can be easily scaled to the needs of the business and the market. It is exciting, competitive, and valuable to the organization as a whole. That is what we believe, and our customers agree.

Let's dive into it...

1

From Monolith to Microservices – *Why?*

There are many considerations that need to be made before a company decides to break up with its monolith. "It's not you, it's me" cannot be used as a patch on the wound. Because it is the monolith that is the problem. At least if it is more like spaghetti than ravioli. Intricate and tangled-up rather than packaged and well defined.

The Monolith is Challenged on Performance

After many years of faithful service, the monolith has gradually grown huge, and its underlying database is opaque. If you change the code in one corner of the monolith, you hold your breath and wait to see where accidental outages occur. Changes are difficult to command, and the test effort is significant.

In a microservice structure, the company can scale and add new features much faster.

The Programming Language is Approaching Retirement Age

The monolith is probably built on increasingly old-fashioned technology that is no longer supported. Slowly, but surely, it is becoming harder and harder for you to find employees who master the monolith's older programming languages.

Your monolith cannot be future-proofed if you cannot recruit employ-

ees with the right programming language skills to maintain it.

Future Security is the Alpha and Omega for Continued Competitiveness

But the thought of reorganizing your IT structure turns you pale. It is a big effort. But what are the alternatives? If the company is to continue to develop at a competitive pace, a modern and well-responsive IT infrastructure is the alpha and omega.

Mjølner's experts agree that the solution is not to go hardcore from the beginning. Think big, but start small. We often say that, but this certainly also applies when it comes to going from monolith to microservices.

It is important to remember that the investment ensures the opportunity to scale the business now - or later. Just having the opportunity is key. And you achieve this flexibility with microservices.

We believe the first step should be to create a modular monolith, as a replacement for the spaghetti-monolith. Much more about that in Chapter 2.

2

From Monolith to Microservices – How?

At Mjølner, we have many years of experience with helping companies from monolith to microservices. In fact, this transition already began about eight years ago, when some of the first companies began their microservices journey.

A typical scenario goes something like this:

A company's monolith is built on an old IT-platform that needs to be migrated. The technology is obsolete and no longer supported. The company reaches out to Mjølner for advice and assistance with the process.

First Stop: the Modular Monolith

As the first step, we help the company create a modular monolith.

A modular monolith still has the limitation of a monolithic database that is ultimately a bottleneck for the system. Therefore, we subsequently disconnect the systems further and eventually split them into services - microservices.

It is a significant job to detangle the monolith and then assemble it into meaningful chunks. But this process is the right way forward.

The Balance between Change and Recognizability

Why not go straight from monolith to microservices? Why take a stopover in a modular monolith?

Here, we run into a hopped phenomenon. It is all about change management. Sudden changes in a large company on measures that are fundamental to the organization's structure can create frictions and unintentional bumps in the road.

Taking a monolith and dividing it into modules allows for faster new features, yet it is still recognizable. This way, the developer staff can keep up relatively easily, and they learn new things in the process.

In fact, elements from this new process go back to principles on how to structure code from many developers' university era. It is not something completely new and groundbreaking, but in the daily work life in an old monolith, it can seem more appealing and convenient to continue 'the old way'.

Employee Ambassadors are Key

If you are in the process of breaking down your monolith, you must get the organization and employees involved in your enthusiasm. The benefits must be clear, and you need internal ambassadors who are committed and lead the way. This is central to the entire process.

As an organization, you must be aware of how to communicate and handle the change among experienced employees. You must be clear on how the organization is approaching the

change. You will figure out the technical requirements and challenges, but the cultural change and especially the reconciliation of expectations requires a great deal of effort. It is the employees who work with the system on a daily basis, and they are the ones you need to excitedly hop onboard the microservices voyage.

Conway's Law and Arbitrary Communication Structures

To succeed in dividing the monolith into modules, you start by analyzing and defining the areas of the organization. It is precisely these areas that must be reflected in the modular monolith you are working towards. This is a very important point, because you typically get the application that the company has structured around.

Renowned programmer and hacker Melvin E. Conway is known for the following quote:

"Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure."

This is central when your company begins the journey away from the spaghetti-monolith. It is challenging to create a microservices architecture if you do not have arbitrary communication structures. In other words: if the individual development teams do not

have self-determination and ownership of their functionality, well, then it will be tangled-up into spaghetti in the end anyway.

That is why it is necessary to divide the entire business into teams. These teams should not only consist of developers, but also of business and operations managers, testers, UX experts, etc., and the teams should each have their well-defined area of responsibility. This makes development with microservices much easier because you do not constantly step into each other's code and domains.

Well-Defined Strategy and Overview of the Domains

One of the most important things to start with is to define a clear strategy on how to untangle the spaghetti. For this purpose, you need the key people who know the existing business and monolith well, and who can take it apart. While the process is in progress, you want to be able to sleep well at night. How? Prepare a roadmap for how existing applications should be sliced up.

When we say that things are tangled-up, it means that the same code does different things and points to the same database. It is the same inputs that are being used, so they need to be separated. It requires skill from the developers on the task, but also from

MELVIN E. CONWAY

"Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure."

the business itself. You must look at it from above and ask "*What exactly are we doing?*" You have to get a razor-sharp overview of your business areas. The result will be a list of a number of modules, e.g. customer management, order management, employee management, inventory management, etc. This way, you gain an understanding of what your new modular system should be able to contain and absorb.

Each module or service must be responsible for its area and data. Often, data in a monolith is mixed in a large database. The method here should be: first we split data apart, and then we assign them to the new well-defined modules. Be aware that the same data (e.g. name and address of a customer) may very well live in several places (e.g. customer support, Business Intelligence, CRM, etc.), and, therefore, some types of data probably appear in several modules (in different forms).

The identification and listing of the domains are some of the most central acts in the entire process. If the overview of domains is wrong or incomplete from the beginning, it will eventually result in a system that is difficult to maintain. Defining which domains exist and which interfaces they have is crucial. Here, your product owners are particularly essential to obtain the correct overview and insight.

So, Simply Done... a Strangler

Pattern and a PoC

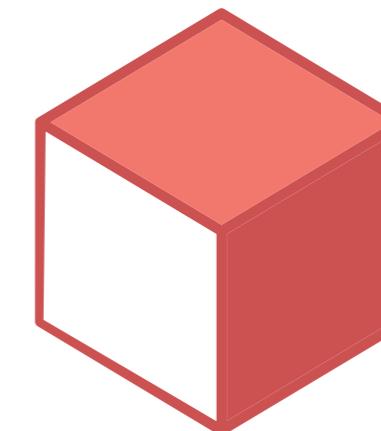
And now, on to the fun part (depending on your preferences). Let's get technical. Here, it is important that you do not begin by testing on any of the existing functionality.

No, you must create something new and use a [strangler pattern](#) to incorporate the new functionality that is in a microservice. Let it be something that provides a bit of business value, but is not of critical technical value. That way, you can also keep the business running without users, either internal or external, noticing anything.

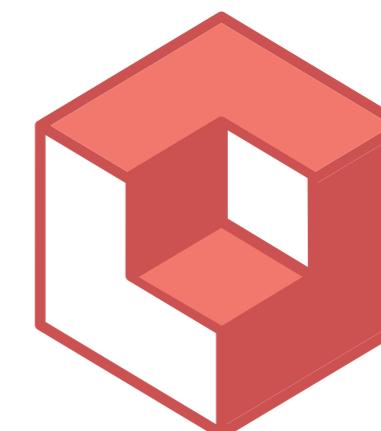
From here, you achieve learnings about what it means to work in and with microservices, as well as a sense of what procedures you should apply in-house. Many topics and questions will arise: how important is deployment, how do you relate to testing and monitoring, do you have the right skills to be in an operating organization with loosely connected systems, etc.

In other words, you gain experience from a Proof of Concept - your first attempt to develop a domain separately and then incorporate the functionality into the system.

But remember: start small. And get your hands dirty. That is the only way to learn.

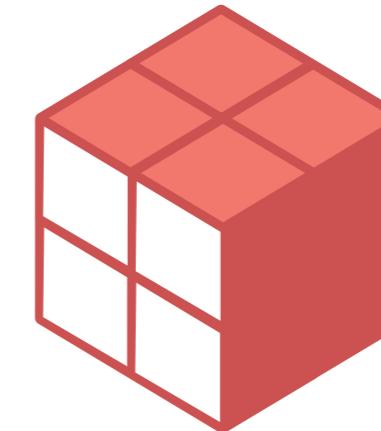


MONOLITH



INDEPENDENT MODULE

Replaces equivalent functionality in the monolith



MODULATED MONOLITH

3

Get Ready for Cloud and Instant Scalability with Microservices

Your company has succeeded in untangling the spaghetti-monolith and is now working with ravioli-it instead; well-defined microservices, or at least modules, each working within its own domain and with clear communication channels between each other. The company is now, to a much greater extent than before, ready for Cloud and to reap the benefits hereof.

Cloud and Scalability - a Match Made in Heaven

There are many reasons to go the Cloud route. You will not be responsible for hosting, and you will achieve fast scaling opportunities (new business opportunities, performance capacity, development resources, etc.) when the need arises in the individual domains. This is a specific and very valuable advantage of microservices, namely that you can just expand individual services when they are loaded, so you avoid unnecessary and expensive optimization of an entire monolith. Combined with Cloud, you can very dynamically turn resources up and down, and you only pay for the real-time need you have. In addition, autoscaling becomes an option for you.

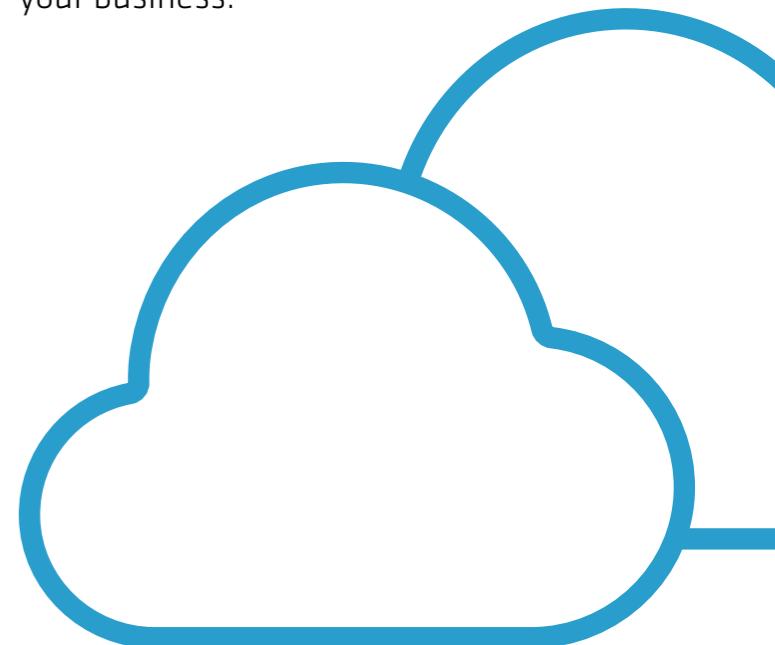
With Cloud, you get access to a sea of existing services with just the touch of a button. Moreover, when you do not even have to operate it yourself, it is obvious that there are great savings

available from expensive operations as well as heavy maintenance.

Get Ready for Cloud - and Decide Later

It is important to say, that it would be false to claim that you cannot go the Cloud route with a monolith. You can develop your business, even if the spaghetti is intact. But it is far more difficult. Typically, there will be several subtleties in monoliths, and they make it difficult and opaque to go from on-prem to Cloud.

We recommend that you get your organization ready for Cloud - either by making a modular monolith or microservices. Take small steps towards something that is not tied to a specific hardware platform. Then, you are no longer forced to run your set-up in one place, and it is subsequently easy for you to look at a Cloud solution that allows for an exponential scalability of your business.





Cloudy with a Chance of Ravioli: a Summary

By now, we hope it is clearer to you why your business should embark on the journey away from the spaghetti-monolith, towards the intermediate station of a modular monolith, and hopefully eventually on towards ravioli and microservices. Then you are very much ready for Cloud and the competitive scaling options you aim for.

It should also be clear that starting small is the way to go. Obtain learning from small Proof of Concepts and focus on whether your organization is ready to take the plunge. Get an overview of your organization and the domains in it.

When you start the work, you must divide your monolith's code base into independent domains. Then you can take the next step, if you want, and turn it into actual microservices. This is a very pragmatic approach to microservices, and it is the one we recommend.

The mindset of the microservice world is that you can constantly develop and improve processes, technologies, and competencies. You must always be ready for the next service or functionality. How are you best ready and most future-proof? Microservices simply are more agile than the spaghetti-monster – so they are your answer.

In the eBook, we have tried to illustrate that before you can get to micro-

services, there is a lot of footwork that needs to be completed. Once done, you can go the microservices path if you want. But no matter what, you are in a much better place than you were when you started with the tangled-up monolith. You have modernized and created an overview.

If you are just getting started, you have already come a long way. If you are not up and running yet, do not despair, but make it clear that you need to get started now to remain competitive in 5-10 years.

The goal should be for the IT engine to be future-proofed, while the users continue their normal operation of your system, without noticing anything. You must have a strong focus on coexistence; suffocate the old system, while building new functionalities next to it, smoothly, and without the users of the system noticing.

Be mindful of the fact that this journey is most of all about your organization's ability to handle change. Going

from monolith to microservices is very much about an organization's mindset, its competencies, and the people who are in it.

Your communication and the narrative of the project's vision are of immense importance for the success of the effort in the end. You must work with your employees and get them on board an exciting journey towards greater opportunity for agility and quick response to market needs and queries. We wish you good luck! You are more than welcome to contact one of us if you need feedback or have questions about the trip you are about to embark on.

Hey, wait a minute! There is never just one truth...

Alongside the microservice wave, there is currently a backlash in the professional tech world that says some-

thing along the lines of: "Hey, what we achieve with microservices is that we can develop code with self-content and code that is good at doing one thing. There are principles for how to make modular code, and you can also do that in a monolith."

You can achieve a lot of the benefits you get with microservices, by reaching for the textbook on how to structure your code in a monolith. Sometimes two domains are helplessly connected, and it may not make sense to separate them at all.

Microservices can be used as an overall guiding star. We recommend you to walk in that direction, but things are rarely black and white. You should not do microservices just to do microservices. It has to make sense for your business. We have tried to shed light on these nuances in this eBook. So remember: think big, but start small!

Do you have any questions or want to know more about how we help companies from monolith to microservices?

Feel free to contact:

LASSE SKOV

BUSINESS DEVELOPMENT MANAGER

lsk@mjolner.dk

+45 29 44 43 77



CREDITS

This eBook is based on interviews with:



MORTEN JOKUMSEN
SOLUTION ARCHITECT



MADS KEBLOV KJELDSEN
SOFTWARE ARCHITECT



MARTIN HENNING JENSEN
SOFTWARE ARCHITECT

KEN BUHL SCHMIDT
BUSINESS DEVELOPMENT MANAGER

LASSE SKOV
BUSINESS DEVELOPMENT MANAGER

Written by:

ANNE KJÆRHOLM POULSEN
COMMUNICATION & MARKETING MANAGER

Layout by:

ANNA LINDBALLE LØNBORG
JUNIOR DIGITAL DESIGNER

