How to migrate a legacy application to

MICROSERVICES with ease

while creating evolving features & functionalities

Flexbase

https://www.flexbase.in

THE PROBLEM STATEMENT

You want to extend or migrate a legacy application to a new distributed architecture that utilizes cloud native infrastructure.

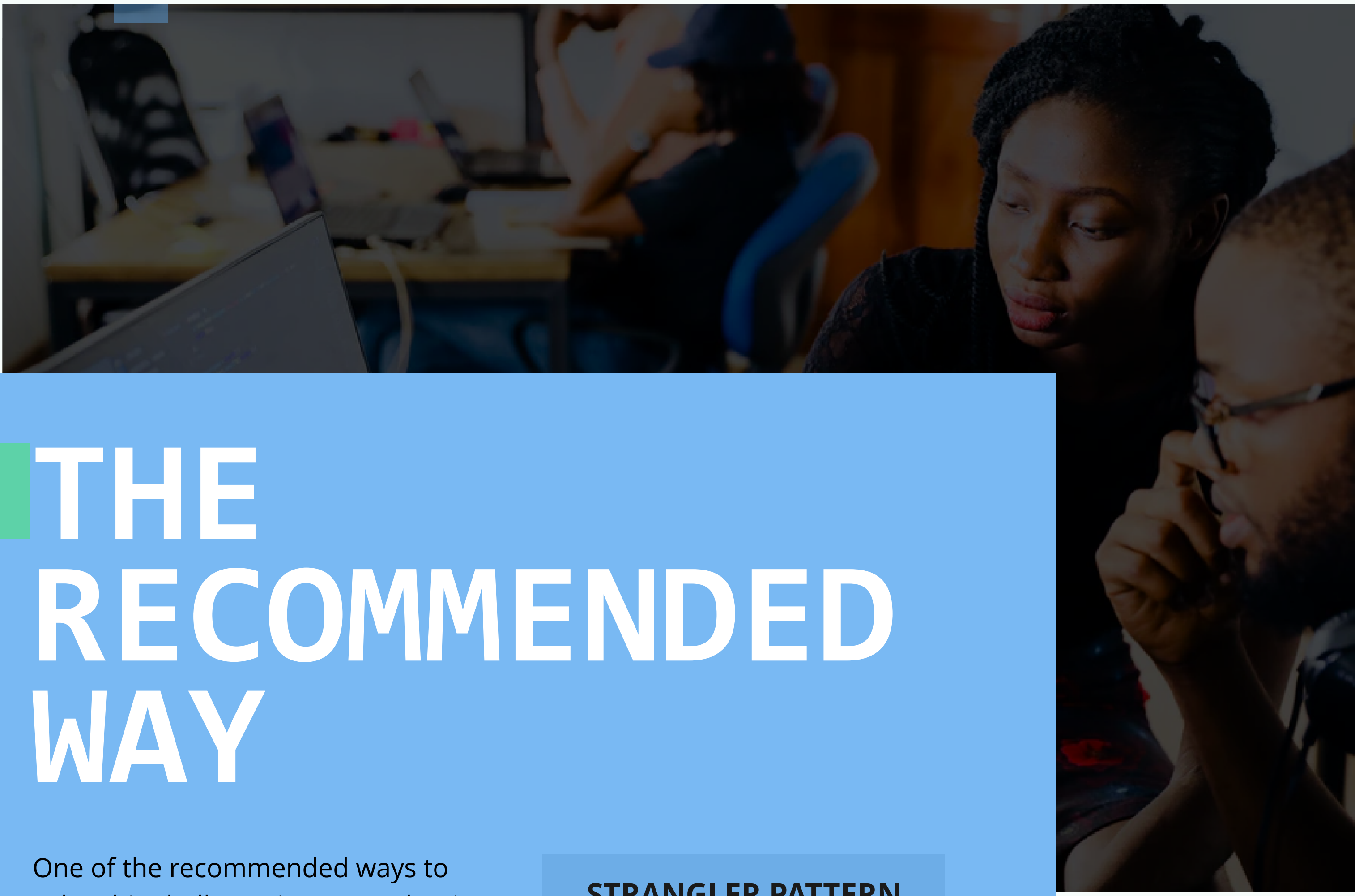
For example, say, you have a monolith ERP system and you want to turn it into microservices based system that uses power of cloud as well.

Or, in another scenario you want to add a microservices based e-commerce application on top of an existing ERP monolith.

In both the cases you want to utilize cloud native infrastructure. You want to have a high performance scalable distributed architecture. At the end of the day, you want applications whose functionality and architecture can evolve easily over time.

You are aware that just containerizing the old application code and transferring it to cloud is not the best way out. It might give you a short term performance boost. But you wouldn't have solved anything in the long run.

At the same time you also know that code rewrite will set you on a path of enormous effort and cost. There doesn't seem to be an easy way out.

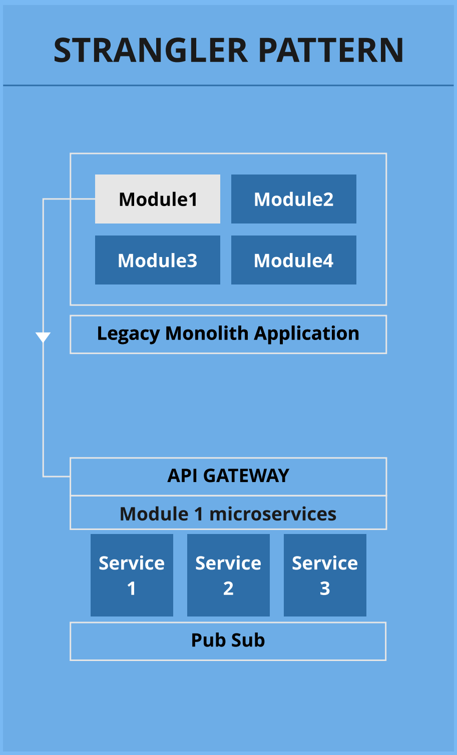


THE RECOMMENDED WAY

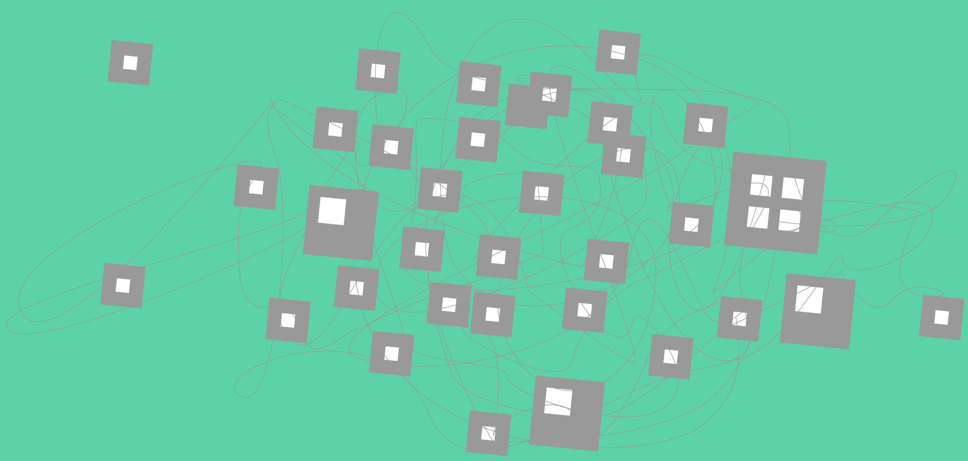
One of the recommended ways to solve this challenge is to use what is called "The Strangler Pattern".

Say, for example, you want to rewrite the existing ERP application. First step will be to do a domain analysis & identify first batch of modules with least dependency that can be moved out. Then rewrite features of these modules as microservices.

Once the new microservices are stabilized, strangulate the first batch of modules. Continue this process till all modules are migrated to microservices.



BUT THERE ARE COMPLEXITIES



Many complexities arise when taking the strangler pattern approach

01 Feature transfer is not straight forward as you will have to rewrite old features in new pattern to make use of distributed architecture.

02 Faulty service boundary designs can lead to messy microservices. So, a lot of time and effort needs to be dedicated to domain design.

03 Complexities arising from wrong choices become difficult to fix later.

04 You will repeatedly be writing lots and lots of plumbing code for the new microservices architecture. Regression issues will blow your costs off the roof.

05 You will be swarmed with plumbing code for the new and old infrastructure making it challenging to move out from old infrastructure bindings.

06 In the process of handling these complexities you may run out of time to rewrite features using scalable and extensible design patterns. You will be forced to copy old patterns & code from the legacy application. This will make you wonder if the entire process was worth it.

FLEXBASE HELPS TRANSCEND COMPLEXITIES

With Flexbase it becomes practical to take up strangler pattern approach for legacy migration. This is how Flexbase helps you overcome the complexities,

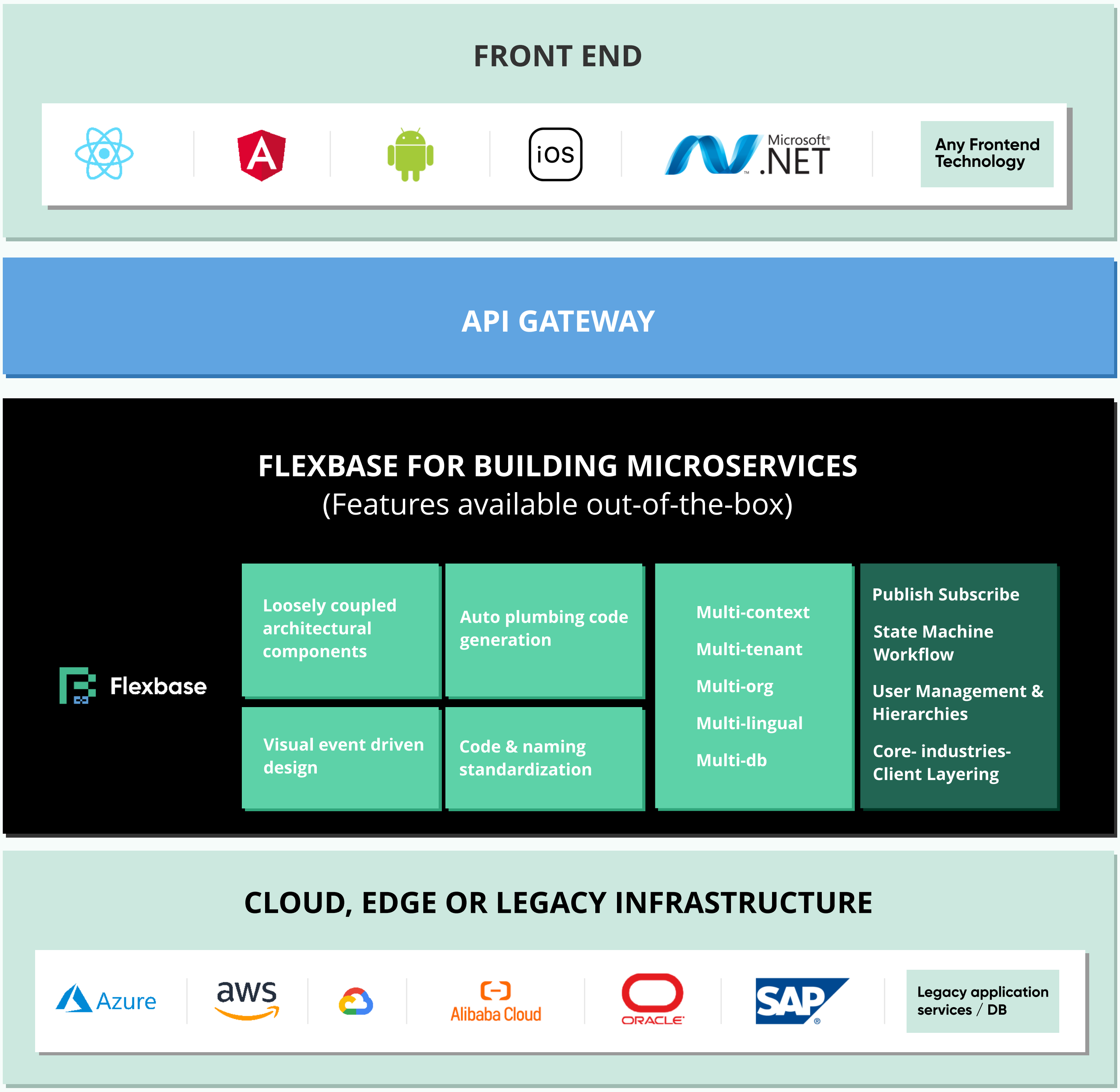
01 You don't need to weave a distributed, event driven, asynchronous architecture from scratch. It is available out of the box in Flexbase.

02 Loosely coupled architectural components are available that let your write extensible application code.

03 Flexbase provides an architectural environment that has publish subscribe, state machine workflow, pluggable user management, multi-context handling and replaceable infrastructure as in-built features.

04 80% of plumbing code is autogenerated.

05 Flexbase studio tool helps you design, build & iterate event driven features fast.


FRONT END
iOS
Microsoft .NET
Any Frontend Technology
API GATEWAY
FLEXBASE FOR BUILDING MICROSERVICES
(Features available out-of-the-box)
Flexbase
Loosely coupled architectural components
Auto plumbing code generation
Multi-context
Multi-tenant
Multi-org
Multi-lingual
Multi-db
Publish Subscribe
State Machine Workflow
User Management & Hierarchies
Core- industries- Client Layering
Visual event driven design
Code & naming standardization
CLOUD, EDGE OR LEGACY INFRASTRUCTURE
Azure
aws
Alibaba Cloud
ORACLE
SAP
Legacy application services / DB

THE BENEFITS

01 You are able to migrate to microservices fast whether you take a piece-meal or big-bang approach.

02 You get an evolutionary architecture that lets you focus on domain, features & future extensibility.

03 You can focus on domain design and create features with new extensible design patterns.

04 You can iterate features fast and edit service boundaries without much overheads.

05 As a result, ample time is available to design service boundaries accurately.

06 You can fix wrong choices easily even when considerable development work has happened.

07 You can easily plug in cloud native infrastructure. Plumbing code gets autogenerated.

08 Standardization of code across the project is easily done.

09 Less number of regression issues.

10 You can create ample lines of high quality reusable code and design patterns.

Conclusion

Moving critical legacy applications to microservices is a necessity today. But doing it right and fast is not easy. You not only need to be able to rewrite features using new design patterns but also innovate.

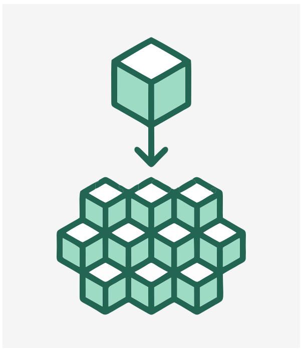
Flexbase helps you handle these complexities with ease. It is like giving ready made ingredients and great equipment in hands of a master chef. You can satiate every taste bud of your customers with ease & style.

You might also be interested in reading

Digital Transformation to Microservices: The Approach

How do you go about this transformation? Is it worth it or is it fine to continue with your current approach with some modifications? How do you decide?

https://dzone.com/articles/digital-transformation-to-microservices-the-approa



Connect with Satya on Linkedin

https://www.linkedin.com/in/satyajitbehera-architect/

Satyajit Behera is chief architect & creator of Flexbase. He has around 20+ years of experience working with .NET/C# stack. A strong proponent of DDD & Event Storming methods, Satya has created Flexbase so that developers and architects can focus on innovating and solving complexities of the domain.

Connect with Harish on Linkedin

https://www.linkedin.com/in/harish-ramachandran-0195431/

harish@sumerusolutions.com

Harish is CEO @ Sumeru Software Solutions Pvt. Ltd. He heads the Flexbase product business development as well. If you would like to consider Flexbase for your next project, connect with Harish on LinkedIn or drop an email to him