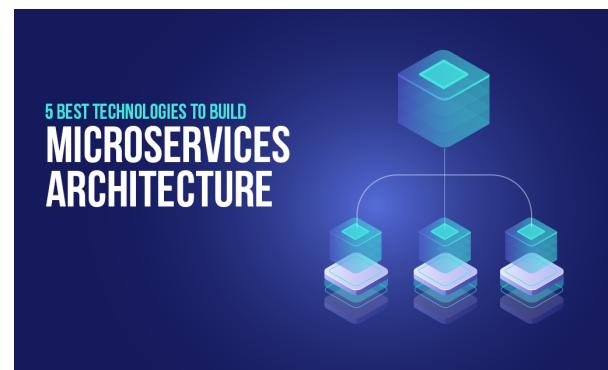




A blog about software development best practices, how-tos, and tips from practitioners.



5 Best Technologies To Build Microservices Architecture

By [Vinugayathri](#)

"To be effective in fulfilling your goals, you must break them down into little tasks and then spend most of your time working on the task in front of you, instead of being overwhelmed by the enormity of the big goal." — Sunday Adelaja,

This may not be the ideology on which Microservices work, but it surely is the principle we follow while developing an App on the Microservices architecture.

Microservices are gaining popularity in the market as businesses work towards building complex and larger applications that can be segregated and handled as a compilation of smaller services. There have been an increasing number of people looking to



A Complete Guide to IT Outsourcing 2020

How can you expect a successful outsourcing implementation if you don't have a clear idea? Read this guide to understand 5 important factors to capture the full potential of outsourcing. In light of this, you can identify the [Right Approach](#), [Optimized Costs](#) and [Quality Delivery](#)...

[DOWNLOAD NOW](#)

Agile

Remote Developers

Drupal Developer

MOST POPULAR POST

20 Scenarios for Testing login Pages & Search Functionalities

5 Best Technologies To Build Microservices Architecture

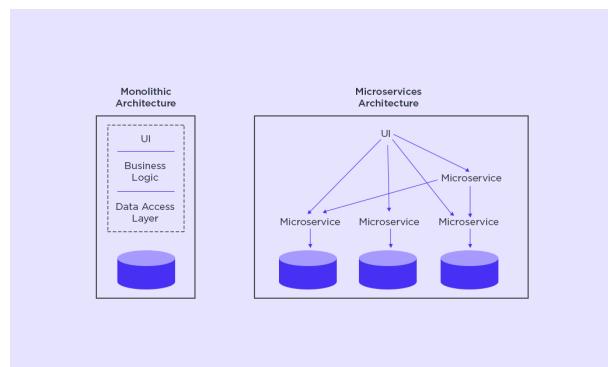
The natural question that comes to your mind is, Which is the most favorable language to be used while developing microservices for your next project?

You can use various technologies based on what best serves the purpose of your microservices. Let's go through some of the programming languages that have the features and libraries, which will help you in getting your job done. But first, let's begin by understanding Microservices.

What is Microservices Architecture?

The microservices architecture is a development methodology wherein you can fragment a single application into a series of smaller services, each executing in its own process and interacting with lightweight mechanisms. The microservices are developed around business capabilities, which are independently deployable with automated deployment mechanism. The microservices architecture needs a bare minimum of management of these services, built in different programming languages and employs different data storage technologies.

Reasons to Migrate to Microservices Architecture



- In a monolithic architecture, for a large application, it is hard to comprehend the complexity and sometimes it is tough to handle the code.
- Applications require extensive manual testing in order to comprehend the impact of alterations.
- Even for a small change, the entire application requires to be

MVC vs Flux vs Redux -
The Real Differences



A Complete Guide to IT Outsourcing 2020

How can you expect a successful outsourcing implementation if you don't have a clear idea? Read this guide to understand 5 important factors to capture the full potential of outsourcing. In light of this, you can identify the [Right Approach](#), [Optimized Costs](#) and [Quality Delivery](#)...

[DOWNLOAD NOW](#)

- The heavy application with monolithic architecture can slow down the start-up time.

Benefits of Microservices

- **Easier Process Adoption** - With a microservices architecture, new technology, and process adoption become simple.
- **Small Modules** - As the application is broken into smaller chunks, it is easy for developers to develop and maintain.
- **Independent Scaling** - Each module in microservices can scale independently through
 - X-axis scaling - by cloning with more memory Or CPU
 - Z-axis scaling - by size using sharding
- **DURS** - Each service in microservices architecture can be independently DURS (Deployed, Updated, Replaced & Scaled)
- **Unaffected** - Even the failure of a single module won't affect the remaining part of applications.

Microservices guarantee increased the autonomy of development teams (speed to market), better fault isolation (reliability), re-usability and scalability.

Criteria for Choosing a Technology for Microservices

With microservices, you can build a reliable platform to extend the business while taking advantage of diversity in languages. Of course, you can use different technologies or languages for different services, but it does not mean it is effective. The microservices architecture comes with lots of operational overhead; hence adding a diverse programming language on top of that can exponentially raise that performance overhead. To reduce that, you should standardize your microservices technology stack by choosing the programming language based on your business needs. Here are the criteria to evaluate the programming language for microservices development:

- Highly observable
- Support for automation
- Consumer-first approach
- Independent deployment
- Modelled around business domain
- Decentralization of components
- Support for continuous integration

Best Languages for Microservices



A Complete Guide to IT Outsourcing 2020

How can you expect a successful outsourcing implementation if you don't have a clear idea? Read this guide to understand 5 important factors to capture the full potential of outsourcing. In light of this, you can identify the **Right Approach, Optimized Costs and Quality Delivery...**

[DOWNLOAD NOW](#)



of them. Let us explore the languages that support microservices development in detail:

1. Java

Annotation syntax, which is easy to read, is the key factor that makes Java a great programming language for developing microservices. This feature makes Java Microservices much easier to develop when powered by Microservices frameworks. It offers more value in readability, particularly while working with complex systems. Java includes many opinions to support developing & deploying Java Microservices. It offers a user Interface, model components as well as connectivity to backend resources, everything within the boundaries of a single, isolated and independently deployed apps.

In addition, many of Java EE standards are well suited for microservices applications like:

- JAX-RS for APIs
- JPA for data handling
- CDI for dependency injection & lifecycle management

In addition, service discovery solutions like Consul, Netflix Eureka or Amalgam8 are effortless in connecting with Java Microservices.

There are several Frameworks for developing Microservices architecture. Some of the Java Microservices Frameworks are as follows:

- **Spring Boot** – This framework works on top of various languages for Aspect-Oriented programming, Inversion of Control and others
- **Dropwizard** – This Java microservices framework assembles stable and mature libraries of Java into a simple and lightweight package
- **Restlet** – It supports developers to build better web APIs, which trail the REST architecture model
- **Spark** – One of the best Java Microservices frameworks, supports creating web apps in Java 8 and Kotlin with less effort

2. Golang

If you want to enhance your existing project, the Golang can be a good choice for microservices development. Golang, also known as Go is popular for its concurrency and API support in terms of microservices architecture. With the Golang's concurrency possibility, you can expect increased productivity of various machines and cores. It includes a powerful standard for developing web services. It is exclusively designed for



A Complete Guide to IT Outsourcing 2020

How can you expect a successful outsourcing implementation if you don't have a clear idea? Read this guide to understand 5 important factors to capture the full potential of outsourcing. In light of this, you can identify the **Right Approach, Optimized Costs** and **Quality Delivery...**

[DOWNLOAD NOW](#)

- **GoMicro** - It is an RPC framework, which comes with the advantages like Load balancing, server packages, PRC client, and message encoding.
- **Go Kit** - The key difference of Go Kit from GoMirco is it needs to be imported into a binary package. Moreover, it is advanced for explicit dependencies, Domain-driven design, and declarative aspect compositions.

In addition to simple syntax, Go microservices architecture includes excellent testing support as it makes it simple to write robust tests as well as embed them flawlessly into workflows.

3. Python

Python is a high-level programming language that offers active support for integration with various technologies. Prototyping in Python is faster and easier when compared to other frameworks and languages. It includes powerful substitutes for heavy implementations like Django. Microservices Python ensures compatibility with legacy languages like ASP and PHP, which allows you to create web service front-ends to host Microservices.

With all these benefits, Microservices Python is considered to have an edge over other languages. Developers who implement Microservices Python use a RESTful API approach - a comprehensive way of utilizing web protocols & software to remotely manipulate objects. With this technology, it becomes easier to monitor the application since it is now broken into components. There is a broad range of Python microservices frameworks to choose from for your web application development. Some of them are as follows:

- **Flask** - Most popular Python Micro framework based on Jinja2 and Werkzeug
- **Falcom** - Create smart proxies, cloud APIs and app back-ends
- **Bottle** - Simple, lightweight and fast WSGI micro framework
- **Nameko** - Best among the Python Microservices frameworks that allows developers to concentrate on application logic
- **CherryPy** - Mature, Python object-oriented web framework

4. Node JS

Node JS became the go-to platform in the past few years for enterprises and startups who want to embrace microservices. Node JS is built with the V8 runtime; hence, microservices Node JS is going to be super-fast for Input-Output (IO) - bound tasks. Normally, Microservices Node JS is developed either using CPU-bound or IO-bound code. CPU-bound program demands many intensive calculations. Every time you run an IO call, Node JS doesn't block the main-thread but submits the tasks to be



A Complete Guide to IT Outsourcing 2020

How can you expect a successful outsourcing implementation if you don't have a clear idea? Read this guide to understand 5 important factors to capture the full potential of outsourcing. In light of this, you can identify the **Right Approach, Optimized Costs** and **Quality Delivery...**

[DOWNLOAD NOW](#)



Microservices Node JS gains popularity in terms of IO-bound tasks.

Stephen Comisso, senior software engineer of GoDaddy states that proper utilization of microservices and Node JS can help to handle the same load with just 10 % of the hardware.

Highly innovative enterprises accept that Microservices Node JS is an excellent combination for attaining increased productivity, high performance, satisfied developers and reduced costs.

5. .Net

ASP.Net, the .Net framework for web development makes it simple to build the APIs that becomes the microservices. It includes built-in support for building and deploying microservices using Docker containers. .Net comes with APIs that can simply consume microservices from any application you developed including desktop, mobile, web, gaming and more. If you have an application, you can start adopting .Net microservices without entirely revamping that application. The initial setup for .Net Docker images has already been done and available on Docker Hub, helping you to concentrate only on building your microservices.

The .Net microservices architecture allows a compilation of technologies between each service; as such, you can use .Net for a certain part of your app without implementing it everywhere. You can mix the .Net microservices with applications written in Java, Node JS, or any other languages. This allows a gradual migration to .Net core technology for new microservices that function in combination with other microservices and with services built with other technologies. Similarly, the .Net microservices can run on all leading cloud platforms.

The future of microservices leads us closer to serverless architecture; particularly, the promise of cost savings by only paying the amount of compute utilized is even more appealing. Microservices is an extensive concept that applies to churn apps, products or solutions to more granular and modular level. Keep in mind, it is not recommended to start microservices architecture from scratch since it is difficult to define the boundaries of each service at the beginning. There is no better way to choose the perfect technology for your microservices. Every technology decision depends on the tools you will use to develop other parts of your application. It also depends on the current knowledge of your development team.



A Complete Guide to IT Outsourcing 2020

How can you expect a successful outsourcing implementation if you don't have a clear idea? Read this guide to understand 5 important factors to capture the full potential of outsourcing. In light of this, you can identify the **Right Approach, Optimized Costs and Quality Delivery...**

[DOWNLOAD NOW](#)