



# Your Organization is Becoming Multi-Technological with Javonet

Discover how you can swiftly develop applications, share components across diverse programming languages, and achieve seamless integration—faster than traditional WebServices with unparalleled performance (20 000x faster). All this without:

- ✓ Building web service (REST, XML, GraphQL or any other)
- ✓ Building any native wrappers
- ✓ Re-writing same code in multiple languages

# What is Javonet?

Javonet is technology that allows you to interact from **any programming language** with **any module or dependency** regardless of technology in which it was created.

Our solution removes the need to write integration layer as calls to in-memory and remote-hosted logic is based on pure method calls like you would be referencing local module. You just create objects invoke methods, get/set fields, subscribe events etc..

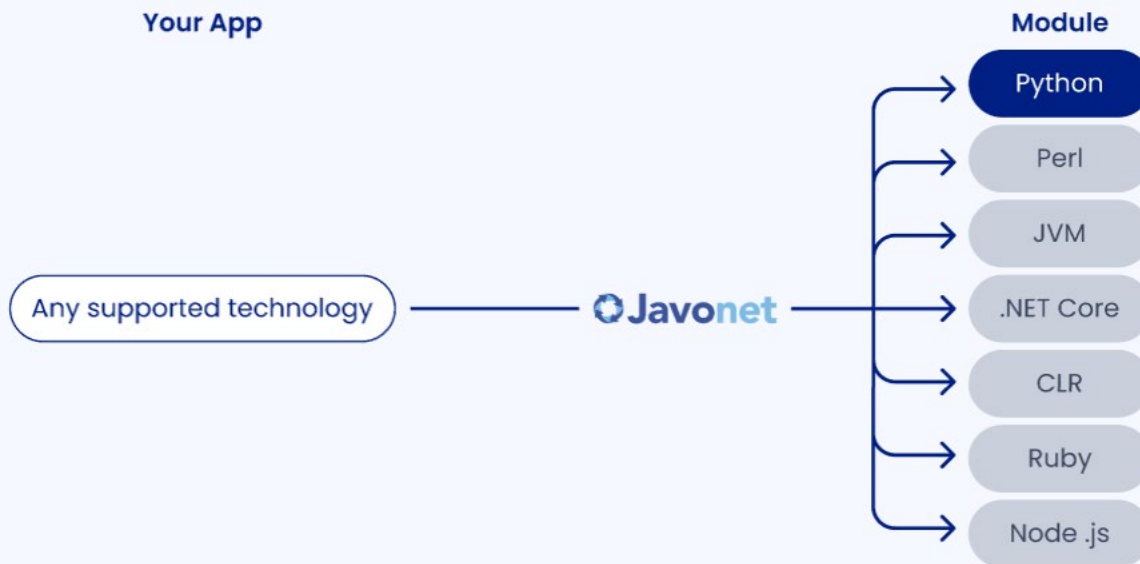
Javonet takes care to execute your call on **valid runtime, in memory** or on **remote machine**, without the need to change your code.

It is delivered in form of advanced library that you can reference in any of your projects.

# When

Use cases and scenarios for applying Javonet

# Use modules written in other programming languages



With Javonet you will be able to create application for **Windows, Linux and MacOS** in **JVM, CLR, Netcore, Python, Ruby, Perl, NodeJS, C++ and Go** that can access and use any existing (Nuget, Maven/Gradle, CPAN, PyPi, Ruby Gems, NPM) or any custom module written in:

- Python
- Perl
- JVM (Java, Clojure, Groovy, Kotlin)
- .NET/Netcore (C#, VB.NET, J#, F#)
- Ruby
- Node.js (JavaScript, TypeScript)

# Sample Use Case



**US Leading Logistic Software company leverages Javonet to access existing .NET route planning algorithms directly from Java microservices in the cloud based Logistic SaaS Application.**

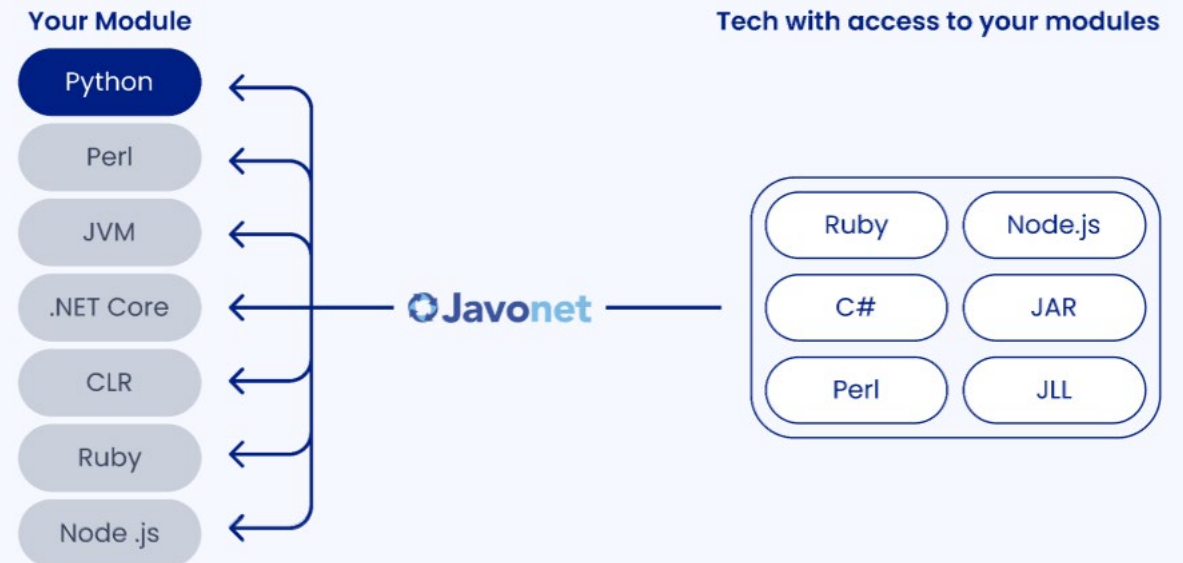
Javonet allows to reference .NET library directly from Java application and interact with its methods and properties and retrieve results as Java value types.

Such ability to share .NET components with other languages allowed to instantly leverage the capabilities of existing algorithms without any compromise and no need to building any integration layer. All of that with performance allowing for rapid exchange of complex data between Java microservices and logic of algorithm written in C# unachievable if it was exposed via remote calls like WebServices, Service Bus etc..

# Create module in one programming language and make it available for other technologies

You can also create module (i.e. SDK, Client Library, Algorithm etc..) in **JVM, CLR, Netcore, Python, Ruby, Perl, NodeJS** and make it accessible for any other application created in:

- Python
- Perl
- JVM (Java, Clojure, Groovy, Kotlin)
- .NET/Netcore (C#, VB.NET, J#, F#)
- Ruby
- Node.js (JavaScript, TypeScript)
- C++
- GoLang



# Sample Use Case



**Breas Medical vendor of ventilation supporting medical devices decided to use Javonet within their SDK to open single language implementation to many other technologies.**

Javonet allowed to implement driver and client library in .NET and provided to consumers building solutions in .NET, Java and other technologies without the need for any additional development.

Development Team leverage Javonet to pass the calls from strongly typed interfaces in different technologies to single implementation in .NET, ensuring that device use single implementation based on .NET USB libraries which provided expected reliability and were utilized in CSV validation.

**Final customized solution will be distributed to end-user customers. Number of targeted machines should reach 500 new machines per year.**



# For which Technologies

Javonet allows you to reference and use modules or packages written in (Java/Kotlin/Groovy/Clojure, C#/VB.NET, Ruby, Perl, Python, JavaScript/TypeScript) like they were created in your technology.

It works for applications created in JVM, CLR/Netcore, Perl, Python, Ruby, NodeJS, C++ or GoLang and gives you unparalleled freedom and flexibility with native performance in building your mixed-technologies products.

## Language you code in

Click to choose.

Click to choose.



## Your runtime

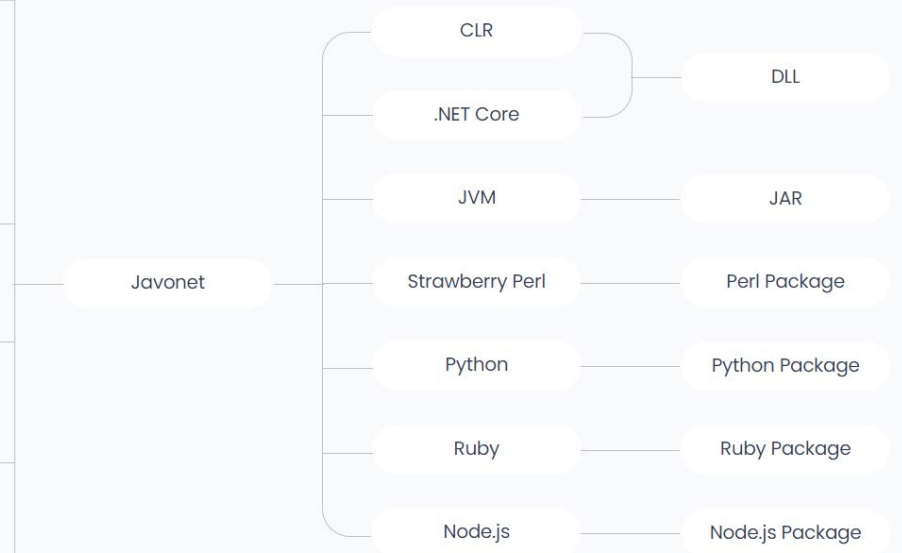
Runtime running your application

## Foreign runtime

Runtimes we can run in your application

## Modules you can use

You will be able to load and interact with them



On Windows, Linux and MacOS



# Daily Use Case Examples

Frontend <> Backend

Microservice <> Microservice  
(any technologies on both sides)

Business Logic > Foreign Language  
Dependency

Mobile App <> Cloud Backend

SDK writing in One Language >  
Exposed instantly to 9 languages

Business Logic <> Data Access  
Layer  
(any technologies on both sides)

Javonet is useful in many day to day use cases and multiple scenarios not even possible before.

Regardless if it's in memory or over the network in microservices, cloud or containers. You can use it to:

- Call any custom logic/library between any programming languages
- Use back-end Components and Algorithms (AI, voice recognition, ocr, route planning, financial scoring etc..)
- Access Physical Devices SDKs and Drivers
- Share UI components from .NET with Java
- Import, export and process high volumes of data
- And any other scenario

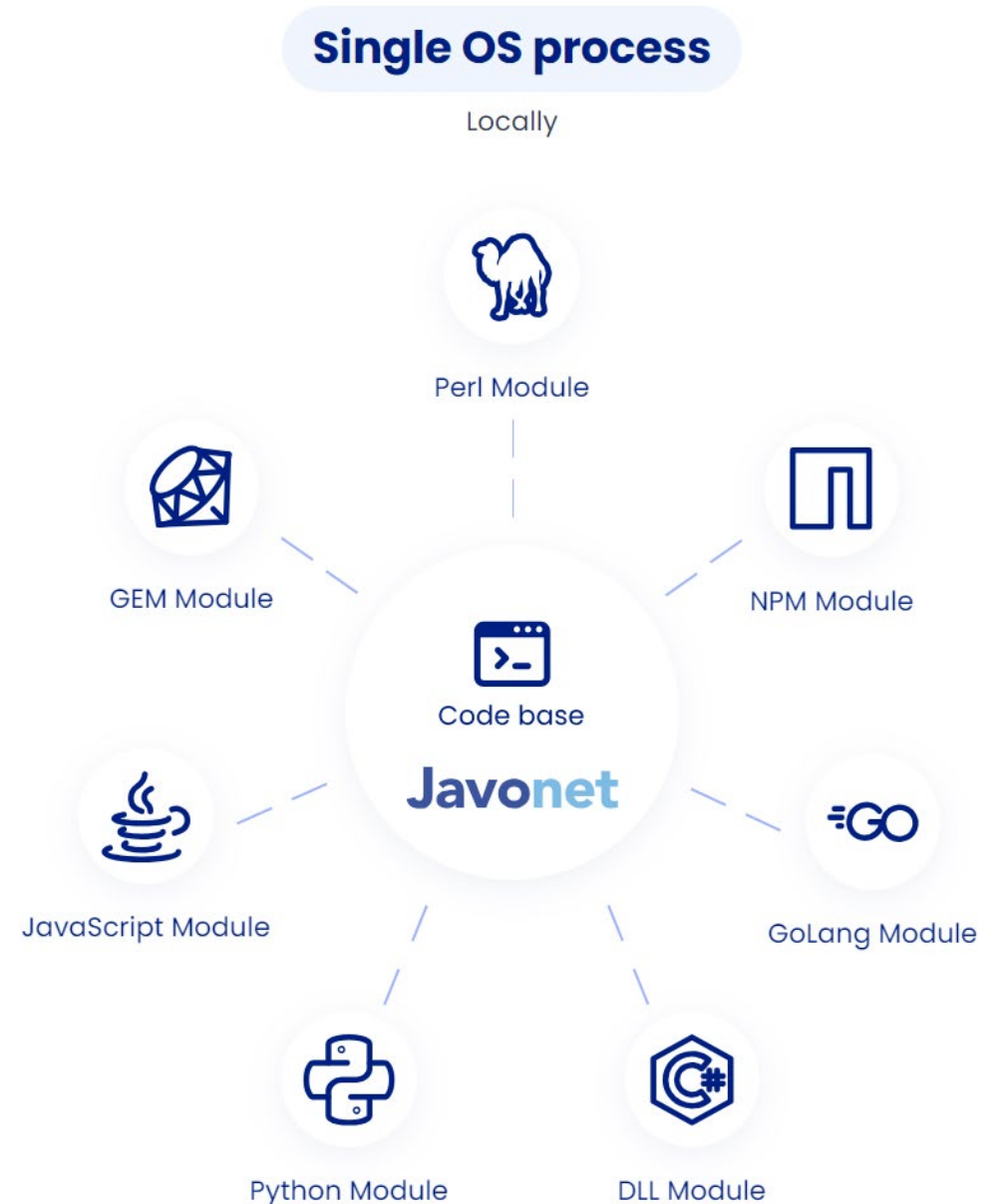
How?

# In Memory

With Javonet **writing your application or wrapper in any supported technology** (including: C#, Java, Groovy, Kotlin, Clojure, Python, C++, GoLang, Perl, JavaScript or TypeScript), **you can reference and interact with: Ruby GEM, Perl, NPM, .NET DLL, Python and JAR.**

Javonet will load in the same process (Windows or Linux/MacOS) additional required runtimes and establish native binding allowing you unrestricted seamless ability to build truly multi-technology apps.

You can use modules created by other team members, any packages from public repositories or third-party SDKs, client libraries and components.



## In the cloud

Microservices



# In the Cloud

Either starting from the scratch or having your existing implementation with Javonet *In Memory* approach, with single line of code you can switch to distributed architecture.

Once you indicate that specific module or runtime should run on another node all your invocations of methods and interactions with interface of that module will get sent and fulfilled on specified node.

**All of that without a single line for writing integration layer. No REST, No SOAP, No gRPC pure methods invocations and Javonet handles the rest!**

Choose from **IaaS, PaaS, or Containers/K8s**. This flexibility allows you to use technology dependency modules as microservices, easily determining which runtimes operate on which node, and deciding where the code should be hosted.

Why?

# Waive the need for integration Layer (development and maintenance)



**Easy integration:** Javonet lets different computer languages work together instantly and easily, no matter what device or system you are using (MacOS, Windows, Linux)



**Fast:** Javonet works instantly, making your integrations run smoother and faster. Up to 20k times faster than WebServices.



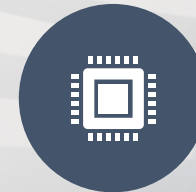
**Safe:** Protected from online threats as it is processed within local OS system or through secure TCP/IP channels in the cloud.



**User-friendly:** Very simple to use, no matter what computer language you are familiar with. Unified SDK and unrestricted interactions between 52 programming languages pairs.



**No extra work:** With Javonet, you don't need to make or look after any extra layer for joining systems together. It saves you time and effort. You just call method you need Javonet decides how and where it runs, regardless of technology.



**No redundancy:** Javonet cuts down on the need to do the same task multiple times in different languages. We literally remove the programming languages barriers.



**Embracing Microservices and containers:** As the tech industry leans towards microservices and containerization tools like Docker and Kubernetes, Javonet ensures diverse services, regardless of language or platform it works on IaaS, PaaS and Containers.



**Connect anything:** No matter how different or tricky software setup is, Javonet can join them together instantly and easily.

# Open new possibilities

**Access New Libraries:** With Javonet, you can tap into libraries you previously couldn't use, expanding your toolkit. Regardless language you use you can access NPMs, JARs, DLLs, Ruby Gems, Python Packages and Perl Modules.

**Create Hybrid Apps:** Build apps that use multiple technologies together, making your software more versatile.

**Share Code Easily:** Spread your code across different teams and technologies without rewriting it.

**Break Boundaries:** Do things that seemed impossible before, thanks to Javonet's powerful integration.

**Save Time and Effort:** Skip the hassle of creating extra layers or repeating tasks, making development smoother.

**Stay Protected:** Enjoy top-notch security while integrating different systems, keeping your data and software safe.

# Deliver Faster

**Direct Module Access:** Quickly tap into various modules without the need to construct time-consuming web services. Just call the method you need and Javonet will route it to right runtime in memory or in remote node.

**Simplified Transition:** Shift from monolithic structures to microservices using just a single line of code.

**Boosted Productivity:** Deliver more in less time, accelerating your development and rollout process.



Learn More

# Got Interested? Learn More!

You can start with Javonet for FREE. Check the materials below:

- [Quick Start Guide](#)
- [Javonet Website](#)
  - [Intro Video](#)



**Thank You! Remember to share  
with your Team Members!**

[Click here to Watch the Intro session with our CTO featured  
on Visual Studio Code Channel!](#)