



## vFunction Frequently Asked Questions (FAQ)

### Q: What is unique about vFunction?

A: vFunction is the first and only platform for developers and architects that intelligently and automatically transforms complex monolithic Java applications into microservices, restoring engineering velocity and optimizing the benefits of the cloud. Designed to eliminate the time, risk and cost constraints of manually modernizing business applications, vFunction delivers a scalable, repeatable factory model purpose-built for cloud native modernization..

### Q: What languages does vFunction support?

A: vFunction currently supports only Java applications. In fact, anything that runs in a JVM is supported (e.g. Scala, Kotlin, etc.) - see the support matrix document on the resources page ([www.vfunction.com/resources](http://www.vfunction.com/resources)) on the vFunction web-site.

### Q: Does vFunction send data to the cloud?

A: No. vFunction is installed locally and doesn't send any information outside the firewall.

### Q: What systems does vFunction require integration to?

A: The vFunction analysis is conducted independently on the vFunction platform. Once a service needs to be extracted, the vFunction platform generates a "Service Spec" which is a list of all classes, configuration files & dependent jars that compose the service. It then copies the relevant sources and configuration files with their package structure & tests if available - hence access to the code repository and build systems is required.

vFunction then creates a pom.xml for the service.

Microservice code is exported, ready for compilation, vFunction can automatically then configure and deploy it on OpenShift or any other cloud platform.

### Q: What about modernizing the database? I have a lot of stored-procedures and really need to clean this up.

A: vFunction focuses on modernizing the application layer within the JVM. However since it analyzes transactions, connections and database table accesses, as constraints that impact service separation, it significantly helps to break the database into multiple databases if the customer so chooses. With that said, vFunction relies on analyzing traffic to/from the databases via JDBC, so for instance, it sees calls to stored procedures but it can't analyze anything that they trigger within the database.

### Q: What does Service Exclusivity mean within the vFunction platform?

A: Service Exclusivity within the vFunction platform means the degree to which classes, methods and variables are exclusive to that service. So 100% class exclusivity means that classes exist only within that service. That would mean it's relatively easier to separate that service.

### Q: How does vFunction identify services and does it relate to Domain-Driven-Design services?

A: vFunction combines static and dynamic code analysis, together with data science and graph theory algorithms to identify the proposed services for extraction. It builds the full call tree and among other things, optimizes the best entry point to create the most exclusive service.

**Accelerate your journey to Cloud-Native Architecture with vFunction, today.**



## vFunction Frequently Asked Questions (FAQ) - CONT.

Its runtime analysis requires the user to perform as many user flows as possible in order to cover the functionality of the application or alternatively collect data in production for a sufficient period of time to achieve such coverage.

Because vFunction takes a “user’s perspective” to identify services, the automatically identified services resemble to a very high degree the services that one would come up with through a Domain-Driven-Design exercise (or Event Storming for example).

That said, if for some reason, the system fails to identify a service one wants to extract, it allows the user to provide a cue to start building the service by entering a class or a method (through its search feature) that serves as a starting point for that service. It would then automatically calculate the interdependencies and exclusivity of that service and allow the user to extract it.

**Q: Does vFunction automatically create the API to a service?**

A: Yes. In cases where there are no existing end-points for the service, then vFunction will automatically create these end-points, as RESTful endpoints using Spring REST or JaxRs. The user then needs to add a call to the service from the monolith, and rerun the analysis.

**Q: Can a service be extracted even if it doesn’t get to 100% exclusivity?**

A: Absolutely. For example a Read-Only database table that is read from 2 different services may impact the exclusivity of those services, but still, a Read-Only table shouldn’t prevent us from extracting those services.

In some cases, a manual intervention may be required, for example if there is a database transaction that needs to be broken up. In these cases either a developer is required to handle breaking up the dependency, or the architect can make an informed decision to alter the architecture and circumvent the dependency.

**Q: What if I’m left with non exclusive classes?**

A: Non exclusive classes that are left at the end of the analysis are effectively your refactoring to-do list. They serve more than a single service, but are not a utility used by many. The recommendation in this case is to start by duplicating the class between the services (which happens automatically) and then refactoring them in a later time.

**Q: Does vFunction support tracking JTA and XA transactions?**

A: Yes. vFunction tracks access to database tables, JDBC transactions, and also JTA and XA transactions are those relate to relational databases.

**Q: Does vFunction support re-platforming of the Java container?**

A: It depends what target platform is required. vFunction’s approach is to keep the exact same Java container for the microservice as a first step (e.g. Websphere to Websphere), in order to minimize risk and unknowns. However, since vFunction extracts a very thin service in terms of the required context, it certainly enables the migration to a lighter container, sometimes even to a Tomcat (Spring Boot). If the starting platform is Spring, vFunction can automatically generate Spring Boot microservices.

**Accelerate your journey to Cloud-Native Architecture with vFunction, today.**



## vFunction Frequently Asked Questions (FAQ) - CONT.

**Q: How does a typical vFunction engagement look like?**

**A:** Installation of vFunction is quite simple and straightforward and can take up to half a day - depending on customer's network configuration and security constraints, since there has to be a connection between the JVM agent and the vFunction server (all installed locally). A vFunction engineer can remotely support the customer in the installation.

**Learning** - once the system is installed properly, the customer needs to perform (ideally) automatic testing to cover as many flows of the system as possible. If no automatic testing is available, manual testing can do. This can take anywhere between one hour to a day. Alternatively, the agent can be installed in production and collect data while the application is operating there.

**Analysis** - once data is collected, up to a 4-day workshop with a vFunction solution architect is scheduled. During the workshop, services are identified, a plan is put together and 1-4 services are extracted, containerized and deployed.

Further extraction of services can follow.

**Q: What if I already know how I want to break the system up into services? Do I still need vFunction to extract only the identified services?**

**A:** The vFunction platform identifies services based on how they are currently written, and tries to minimize constraints and increase the exclusivity of the services.

However, it is encouraged for the architect to use the information exposed by the platform, and modify the automatically identified services to match the target architecture, taking into account the constraints that may arise from these modifications.

**Q: Can vFunction analyze a production system?**

**A:** Yes. Many customers deploy vFunction in production and leave its agent to collect data over a few hours up to a day or two. The performance impact on the application is about 1%-2%. vFunction's factory dashboard allows the project team to monitor and manage the various applications in the modernization pipeline.

**Q: What if my application runs on multiple JVMs?**

**A:** You can install the vFunction JVM agent on multiple JVMs, run serial measurements on those JVMs and then merge the results to get a unified view of the system on the vFunction platform.

**Q: Does vFunction fit a particular set of Java applications?**

**A:** vFunction tracks information in a very fundamental way (native JVM agent, memory interactions etc.) and therefore is a very generic solution for any Java application.

**Q: Where do I find vFunction installation guides?**

**A:** Contact us at [info@vfunction.com](mailto:info@vfunction.com) and we will be happy to provide you access to our installation guides.

**Accelerate your journey to Cloud-Native Architecture with vFunction, today.**