



vFunction Frequently Asked Questions (FAQ)

Q: What is unique about vFunction?

A: vFunction is the only purpose-built cloud native modernization platform for Java applications. By combining dynamic analysis, static analysis, AI and automation the platform extracts any service from your Java monolith into a compilable, working microservice, containerizes it and deploys it onto modern infrastructure.

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 our [Support Matrix](#) for more details.

Q: How does vFunction work?

A: The vFunction modernization cycle consists of 5 phases: learning, analysis, and extraction, followed by the containerization and deployment of the auto-created microservices. For more info read our one pager or go to our web site www.vFunction.com or see our [User Manual](#).

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, since vFunction relies on analyzing traffic to/from the databases via JDBC, it can't analyze stored procedures or anything that happens within the database.

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

A: Service Exclusivity within the vFunction platform refers to 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 and thus is 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 AI to identify the proposed services for extraction. It builds the full call tree and amongst other things, optimizes the best entry point to create the most exclusive service.

Its runtime analysis requires the user to perform as many user flows as possible in order to cover the functionality of the application.

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



Because vFunction takes a “user’s perspective” to identify services, the automatically identified services represent a similar set of services that one would come up with through a Domain-Driven-Design exercise (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 or hint 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: There are several options to extract a service. A common approach is to extract a service starting from the end-point of the service, at which point no API has to be added. Such an endpoint can be a REST endpoint, a Message-Driven-Bean, or multiple other options. In case a service needs to be extracted but no endpoint currently exists, then once the service is extracted, the user has to manually add an endpoint to the service, using the entry-point to the service as the service-layer. The user then needs to add a call to the service from the monolith, and rerun the analysis. In a coming release vFunction will offer automatically generated Restful APIs for certain frameworks (starting with Struts).

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 vFunction from extracting those services.

In some cases, a manual intervention may be required, for example if there are static objects that need to be initialized, or 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: Does vFunction support re-platforming of the Java container?

A: At this moment, vFunction’s approach is to keep the exact same Java container for the microservice (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, many customers have found that it enables an easier migration to a lighter container, sometimes even to Tomcat (Spring Boot).

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: 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 (ideally) perform automatic testing to cover as many flows of the system as possible. If no

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



automatic testing is available, manual testing can do. This can take anywhere between one hour to a day.

Analysis - once data is collected, a 4-day workshop with a vFunction architect is scheduled. During the 4-day 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 how they behave, 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: The 'Fast Learning' feature of vFunction supports analyzing applications running in production environments. Fast Learning has limitations and not all application servers and frameworks may be supported. Please contact the vFunction team for more information on this new capability.

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 applicable solution for any Java application.

Q: Where do I find vFunction installation guides?

A: [Server Installation Guide](#)
[Server Installation Guide \(Sudo less\)](#)
[Controller/Agent Installation Guide \(Linux\)](#)
[Controller/Agent Installation Guide \(Win\)](#)
[Controller/Agent Installation Guide \(Sudo less\)](#)
[Controller/Agent Installation Guide \(Manual\)](#)

Q: Do you support .NET application modernization?

A: Only Java application modernization is supported today with additional support for frameworks such as .NET in consideration as potential roadmap features for future releases.

Q: How is vFunction priced?

A: Pricing is based on a per application (generally one or more JVMs) basis with volume and site licensing options available. Please contact vFunction info@vfunction.com for further pricing details. Generally, anyone can install vFunction, analyze an application, identify services for extraction, understand their interdependencies and design the target microservices architecture for free. Only the automatic extraction and creation of services requires a paid license.

Q: What cloud native platforms do you integrate with (e.g., Openshift, Kubernetes, Knative, Anthos)?

A: Microservices created from the vFunction platform can be run on any container or cloud native environment. A specific operator has been created for Red Hat Openshift <https://marketplace.redhat.com/en-us/products/vfunction> integration (with additional integrations planned for future releases), and customers have successfully integrated with Kubernetes, Knative, and related environments.

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



Q: Do you support Oracle ATG commerce Web Commerce application migrations?

A: vFunction has released Beta support for Oracle ATG commerce Web Commerce application migrations in release 1.5.

Q: Does vFunction work with any specific integrators or delivery partners?

A: vFunction partners with a wide array of delivery partners with an application modernization practice. If you are interested in exploring a partnership with vFunction please contact us at info@vfunction.com.

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