

# webMethods Microservices Runtime

Fact sheet

## Embrace next-generation integration architecture with webMethods

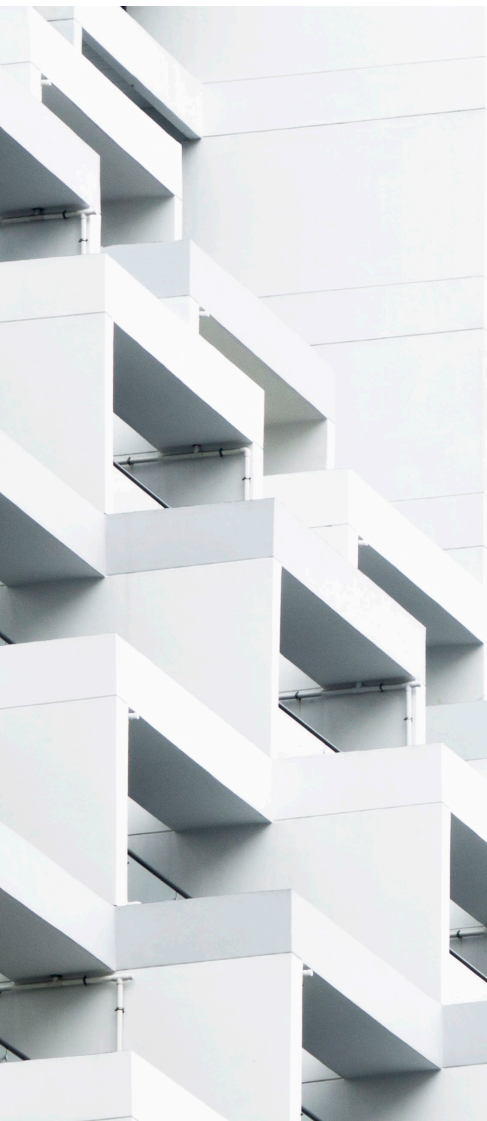
Businesses are increasingly adopting architecture styles that take advantage of the agility and scale of the cloud. Microservices are one of the key building blocks of the next generation of application and integration architecture, enabling developers to construct applications as collections of small services, each running in their own processes and communicating using lightweight protocols. These microservices are typically built to provide granular business functions that are independently deployable and very robust.

With webMethods Microservices Runtime you can deliver your integrations as microservices that are fault-tolerant, are highly scalable, and interact securely using lightweight messaging or APIs. You can:

- Create resilient, platform-agnostic microservices, in the cloud or on-premises
- Scale microservices with industry-standard tools like Docker®, Kubernetes®, Red Hat® OpenShift and others
- Communicate via lightweight messaging with Software AG's Universal Messaging
- Expose microservices as public or private APIs and secure them using webMethods Microgateway
- Manage infrastructure with a health check based on
- Prometheus™ and recovery capability

### Key benefits

- Create and deploy modular integrations that adapt independently to market requirements
- Scale APIs in the cloud with a distributed microservices-backed architecture
- Support CI/CD with built-in DevOps automation tooling
- Ensure reliability with a reactive architecture that includes builtin fault tolerance and recovery
- Encourage a rapidly evolving ecosystem in which business capabilities can spin up quickly



## Features

### Self-registering services:

Endpoints in the cloud can change unexpectedly, but user-facing applications need to provide uninterrupted service. webMethods Microservices Runtime provides pluggable service registry support, enabling applications to look up endpoint information at runtime so that users always get a response. Microservices Runtime comes preconfigured for the Consul registry and supports other service registry providers.

### Self-healing services:

When services rely on each other, those dependencies can cause cascading failures in the cloud. webMethods Microservices Runtime supports an architecture pattern known as “circuit breaker” that can be used to control cascading failures. The circuit breaker can be opened based on a predefined threshold of exceptions. While the circuit is open, the calling service can be configured to handle a failure gracefully. When a predefined reset period elapses, the circuit breaker will check on service health to determine whether to close the circuit, returning the service to normal behavior, or continue with alternate behavior.

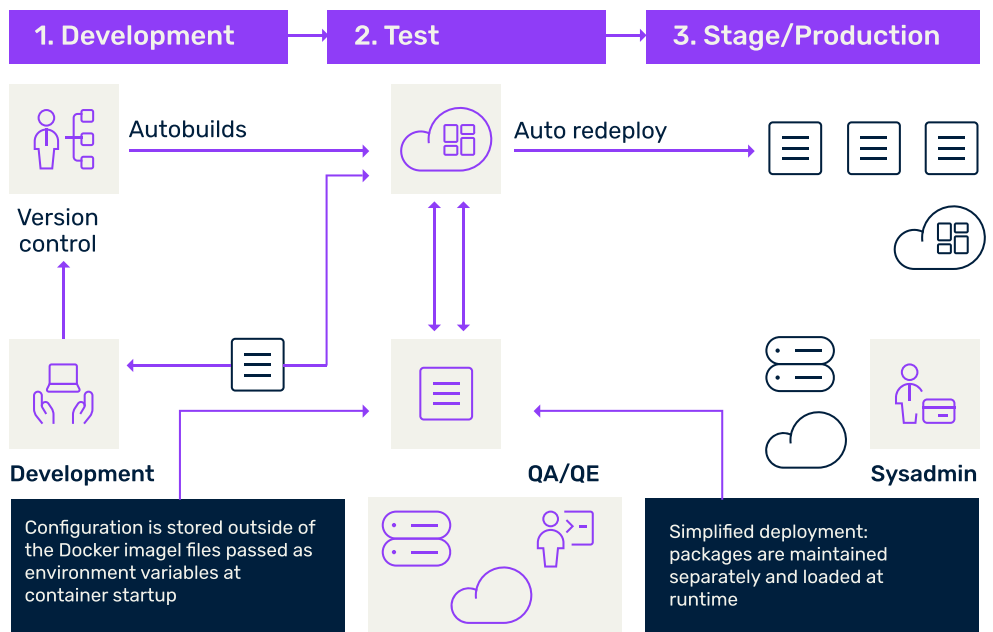
You can also set up custom health checks with out-of-the-box indicators like disk space, JDBC® pools, memory, Java® Message Service (JMS) connections and adapter connections. Administrators can define a combination of these key indicators and use them for proactive monitoring or to automatically restart a misbehaving service.

### Continuous delivery:

To achieve reliability and quality in a CI/CD environment, you can follow a test-driven development methodology. The webMethods Unit Test Framework enables test cases to be designed before starting development, and those tests can be automated and included in the DevOps lifecycle. The webMethods development environment integrates with most common third-party DevOps tools.

### Self-scaling services:

To achieve the rapid scalability and flexibility of a microservices architecture, it’s critical to be able to automate deployment. webMethods Microservices Runtime can be deployed with Docker and Kubernetes for packing, distributing and managing your applications automatically. Out-of-the-box Docker scripts simplify deployment, and built-in readiness and liveness checks enable Kubernetes to manage your containers.



**On-the-fly Docker deployment:**

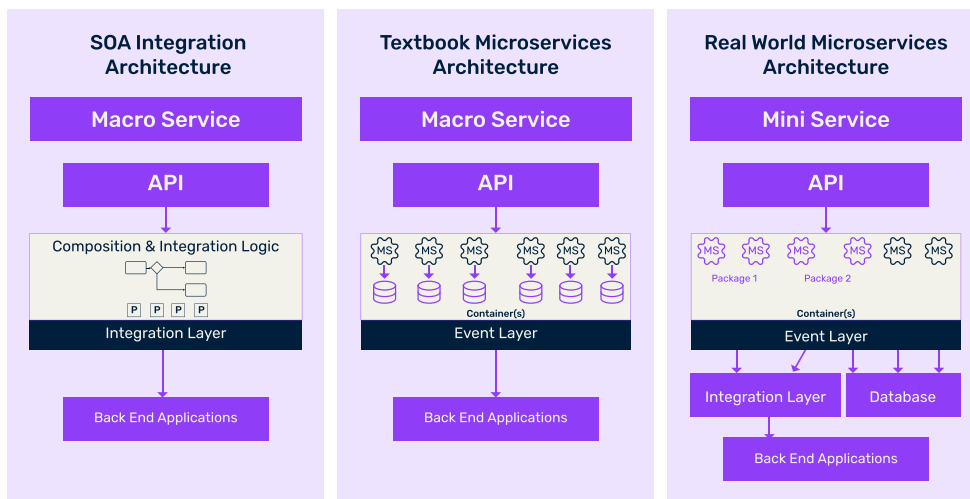
When a microservice is deployed using Docker containers, there are many configurations that might need to be updated: connector configurations, access control lists, extended settings and schedulers, cache manager configurations, and more. With webMethods Microservices Runtime, you can dynamically inject the appropriate configuration at runtime using Docker environment variables without rebuilding the container. In addition, the desired versions of services or packages that constitute the microservice can be selected at deployment time. By isolating the elements of your microservices runtime and building it on the fly instead of prebuilding static Docker images, you deploy what you need, when you need it.

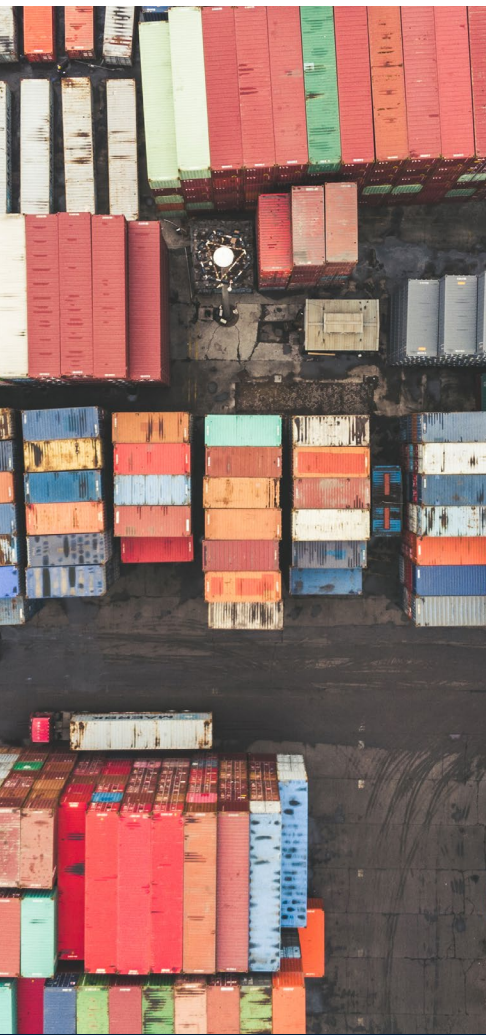
**Right-sized microservices:**

Textbook microservices require an event-driven architecture where all services are loosely coupled and data is stored local to the microservice. However, many organizations are taking a more pragmatic approach to achieve their business objectives, using a transitional style known as miniservices that can be managed and scaled independently. Alternatively, it may be desirable to deploy well-defined business capabilities as large-grained services known as macroservices. webMethods Microservices Runtime supports different levels of granularity and service patterns by enabling users to easily add modules for event routing, RDBMS support, flat file handling and more—up to and including all functionality in webMethods Integration Server.

**Choice of implementation style:**

One of the advantages of microservices is the ability to use the appropriate programming model and platform for each service. webMethods Microservices Runtime supports several programming languages for service development: Java, C, C++, and native FLOW. In addition, pre-built APIs and services, an event bus, and in-memory data make implementation faster and more flexible. Finally, webMethods Microservices Runtime runs on a wide variety of operating systems.





### **Connectivity to data and apps:**

Microservices need to connect to a variety of data sources and storage technologies as well as other applications and services. webMethods Microservices Runtime has a full range of connectors to support the most modern standards, such as OData, REST and the full complement of SOAPbased web service support. Also available are pre-built adapters for simple drag & drop interfaces to dozens of commercial applications, both inside and outside the firewall. webMethods Microservices Runtime comes preconfigured for highperformance, low-latency messaging within the enterprise and for web and mobile connections. For distributed microservice architectures, reliable, secure and scalable messaging that supports asynchronous communication is a key component.

### **Microgateway protection and policy enforcement:**

Creating and deploying microservices is only part of the challenge with consumer-facing applications. It's also essential to secure and monitor access to them. webMethods Microgateway can be deployed as a headless, self-contained Java app or as a Docker image. It can also be deployed standalone or in sidecar mode, which enables the microgateway to easily scale with the microservice and support distributed microservices architectures.

webMethods Microgateway supports both transport and message-layer security policies, including authentication, authorization, digital encryption and digital signatures. It enables you to route requests using a variety of routing methods based on payload content or context, straightthrough routing and load-balancing among multiple service providers. And finally, you can create policies to control and monitor API consumers.



### **Take the next step**

To learn more about webMethods Microservices Runtime, talk to your Software AG representative or visit [www.softwareag.com/integration](http://www.softwareag.com/integration).

## **ABOUT SOFTWARE AG**

Software AG began its journey in 1969, the year that technology helped put a man on the moon and the software industry was born. Today our infrastructure software makes a world of living connections possible. Every day, millions of lives around the world are connected by our technologies. A fluid flow of data fuels hybrid integration and the Industrial Internet of Things. By connecting applications on the ground and in cloud, businesses, governments and humanity can instantly see opportunities, make decisions and act immediately. Software AG connects the world to keep it living and thriving. For more information, visit [www.softwareag.com](http://www.softwareag.com).

© 2021 Software AG. All rights reserved. Software AG and all Software AG products are either trademarks or registered trademarks of Software AG. Other product and company names mentioned herein may be the trademarks of their respective owners.

fs\_webmethods\_microservices\_en