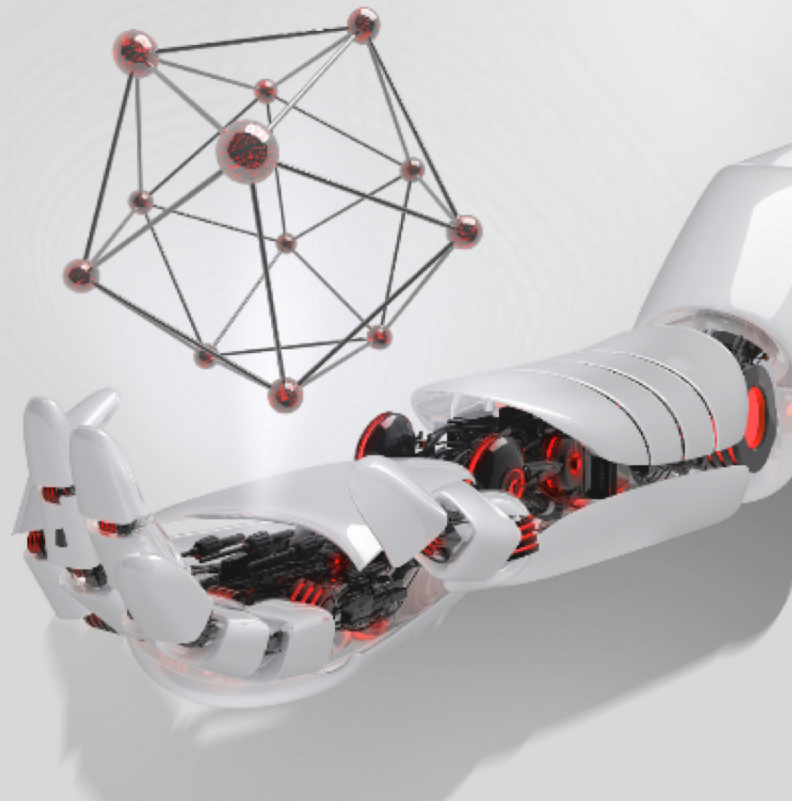


Experience on Fast Microservice Programming

Ju Zhen, Yang Bo

Open Source Software Capability Center

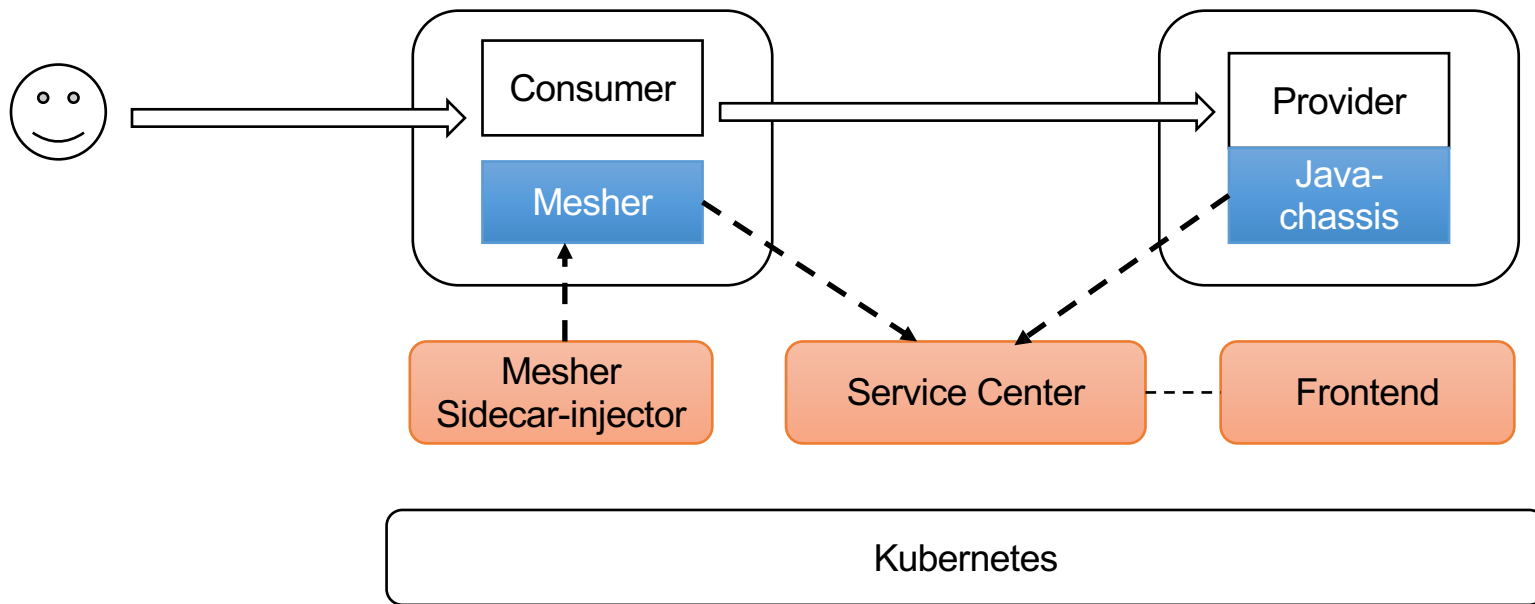
2018-10-12



Contents

- Case Introduction
- Provider Development
- Consumer Development
- Kubernetes Deployment
- Test and Verification

Case of Quick Microservice Development



Provider Development

- Access `start.servicecomb.io`, enter the **Group** and **Artifact**, and download the template project.
- Decompress the template project, and import it to the IDE using Eclipse or IDEA.
- Modify the registration center address in the **microservice.yaml** file.
- Develop an application in SpringMVC mode.
- After the development is complete, package the application as a Docker image.

The screenshot shows the start.servicecomb.io web interface. At the top, there's a navigation bar with the text "SERVCOMB SPRING INITIALIZR" and a subtext "bootstrap your application now". Below this, there's a form to generate a project. The form has a header "Generate a" followed by a dropdown menu set to "Maven Project", the word "with", another dropdown menu set to "Java", and "and Spring Boot" followed by a dropdown menu set to "1.5.12".

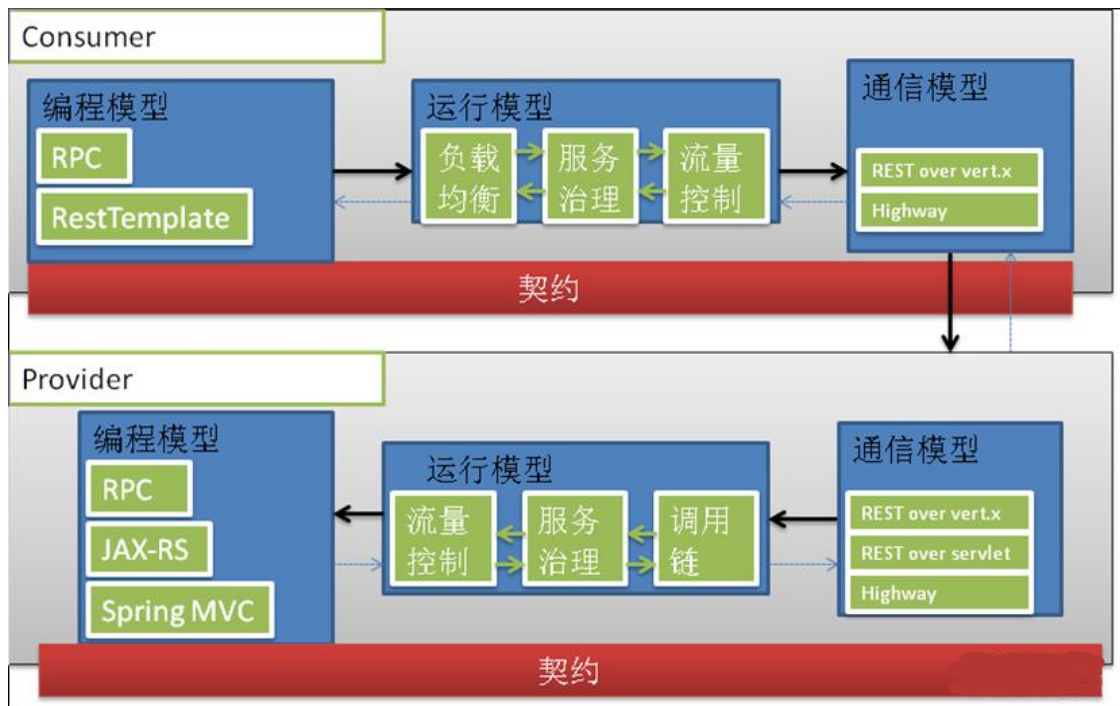
The form is divided into three main sections:

- Project Metadata:** Includes fields for "Artifact coordinates" (Group: demo.foo.bar, Artifact: demo).
- Dependencies:** Includes a search bar for dependencies (Web, Security, JPA, Actuator, Devtools...) and a "Selected Dependencies" list containing "ServiceComb".
- ServiceComb Parameters:** Includes fields for "Service Name" (HelloServiceComb), "Service Version" (0.0.1), "ServiceCenter Address" (http://127.0.0.1:30100), "REST listen address" (0.0.0.0:9000), "Highway listen address" (0.0.0.0:9090), and "Governance" options (Loadblance, Flowcontrol, CircuitBreaker, Tracing).

At the bottom right, there is a green button labeled "Generate Project" with a keyboard shortcut "alt + d".

Java-chassis SDK Principles

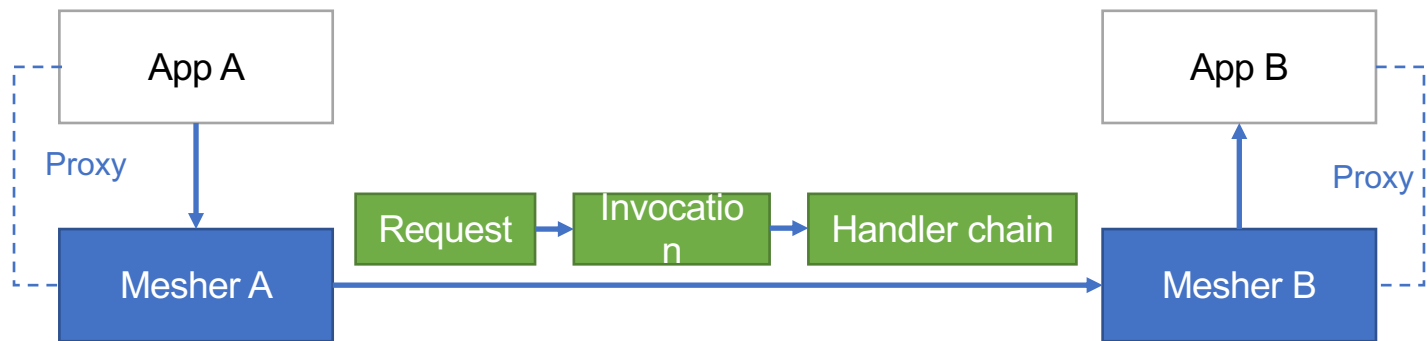
- Development experience on Provider is the same as that on Consumer.
- The communications model is isolated from the programming model to meet the requirements of different service scenarios.
- The running model can be flexibly expanded to support interconnection with external systems.



Consumer Development

1. Develop service logic (SDKs and dependencies are not involved).
2. Build a Docker image.

Code Zero-Intrusion: Basic Working Principles of Mesher



Quick Deployment: Sidecar-injector Principles

Kubernetes AdmissionWebhook

Prerequisites:

- Kubernetes 1.9 or later is used.
- MutatingAdmissionWebhook and ValidatingAdmissionWebhook are enabled.
- namespace, configmap, and mutatingwebhookconfigurations access permissions are available.

Service Configurations:

Pod Spec comment: **sidecar.mesher.io/inject: yes**

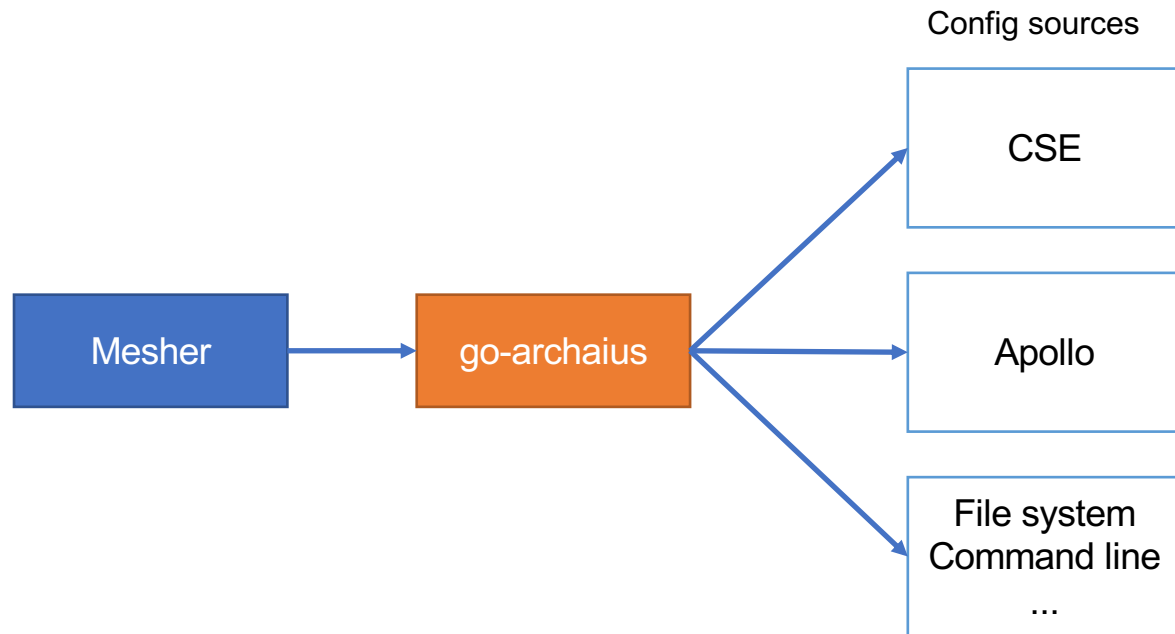
Namespace label: **kubectl label myns sidecar-injector=enabled**

Pod A

Mesh A

Sidecar-injector

Flexible Configuration: go-archaius



Kubernetes Deployment

Provider

- Deployment
- Service

Consumer

- Deployment
- Service

Apollo

- Deployment
- Service

Sidecar-injector

- Deployment
- Service
- ConfigMap
- ...

ServiceCenter

- Deployment
- Service
- ConfigMap

Frontend

- Deployment
- Service
- ConfigMap

Test and Verification

- Verify that the service is properly accessed.
- Verify the microservice governance functions such as fallbreak and flow control.

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home and
organization for a fully connected,
intelligent world.

**Copyright©2018 Huawei Technologies Co., Ltd.
All Rights Reserved.**

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.

Huawei Confidential

