Offloading Asynchronous Activities with Lightweight, Short-lived Tasks

Richard Seroter
SENIOR DIRECTOR OF PRODUCT, PIVOTAL

@rseroter

Overview

The role of asynchronous processing in microservices

Problems with the status quo

Defining “serverless” computing

Describing Spring Cloud Task

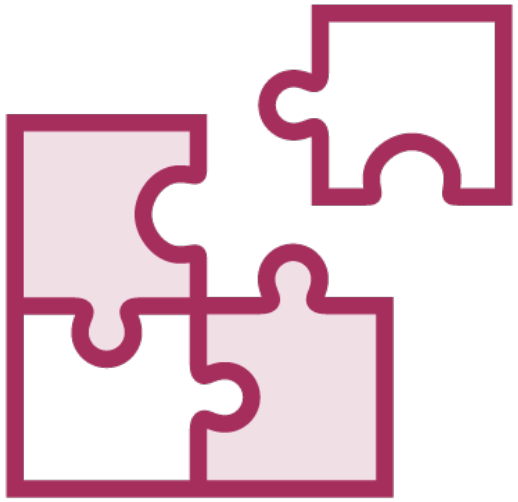
Creating a Task

Reviewing storage options for results

Options for invoking tasks

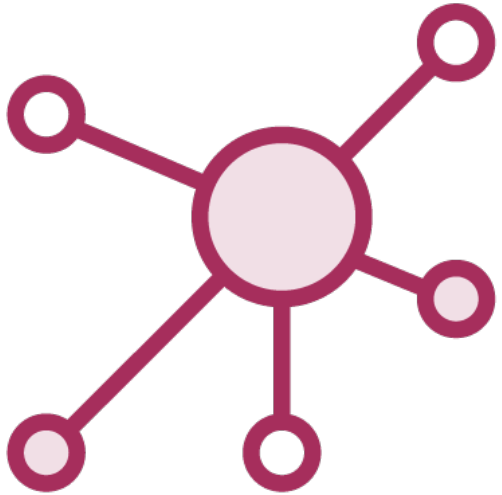
Summary

The Role of Asynchronous Processing in Microservices



Reduce dependencies between services

Support low latency, high throughput



Facilitate event-driven computing

Problems with the Status Quo

- Consuming resources even when services aren't in use
- Services baked into monolithic deployments
- Challenges scaling services on demand
- Difficulty tracing service calls

What Exactly Is “Serverless” Computing?

Deploy “function” instead of “application”

Run code without knowledge of infrastructure

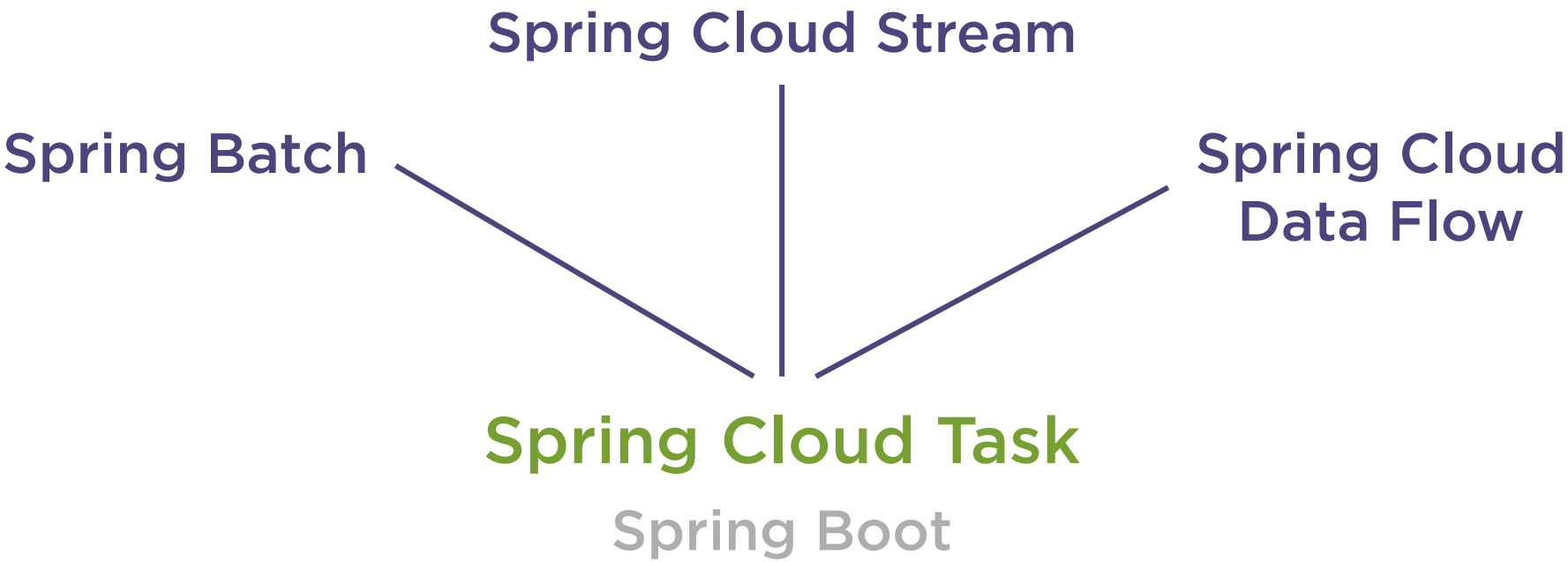
Elastic, automatic horizontal scaling

Start fast, run short

Spring Cloud Task

Short-lived, asynchronous microservices.

How This Fits into the Spring Ecosystem



Creating a Task

Add classpath dependencies to POM

Annotate the class with @EnableTask

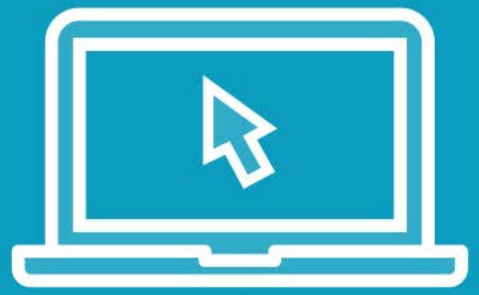
Add business logic to run the Task

Deploy to the Maven repository

How Does Task's Logic Work?

- Spring (Boot) app with access to beans
- Task is stateless
- Bootstrap logic with Runner
- Can subscribe to lifecycle events

Demo

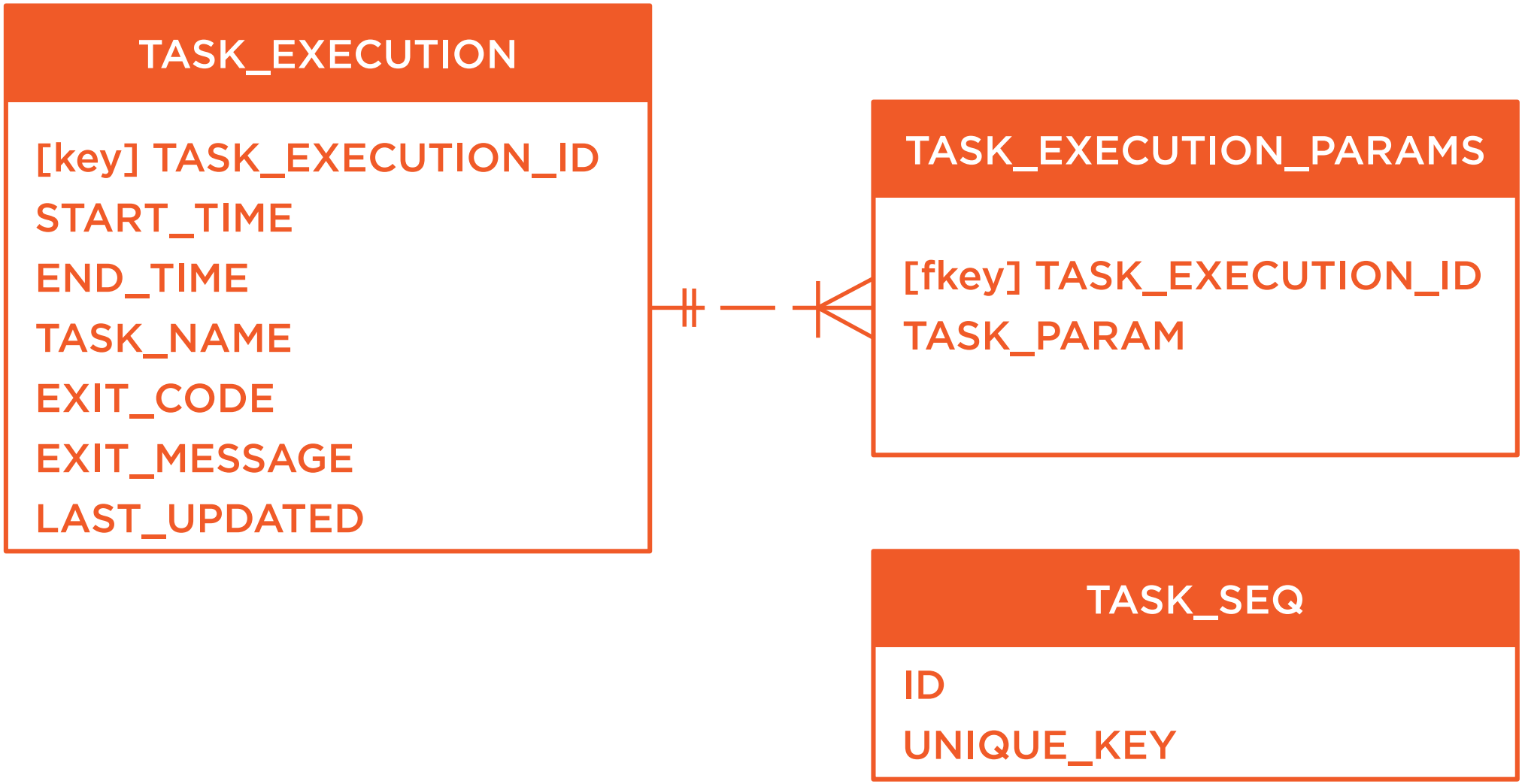


- Create a new Spring Boot project for Toll Processing task
- Annotate primary class
- Add task logic as CommandLineRunner
- Execute task and observe results

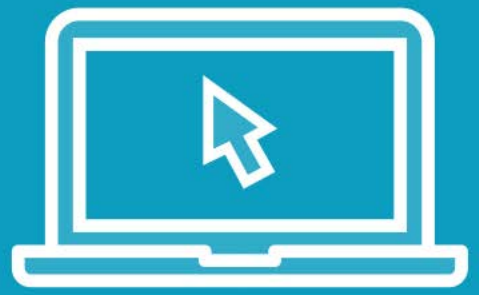
Multiple Task Result Storage Options

H2	HSQLDB	MySQL
Oracle	PostgreSQL	SQL Server

Repository Entity Relationship Diagram



Demo



Create MySQL database

Add MySQL dependencies in POM

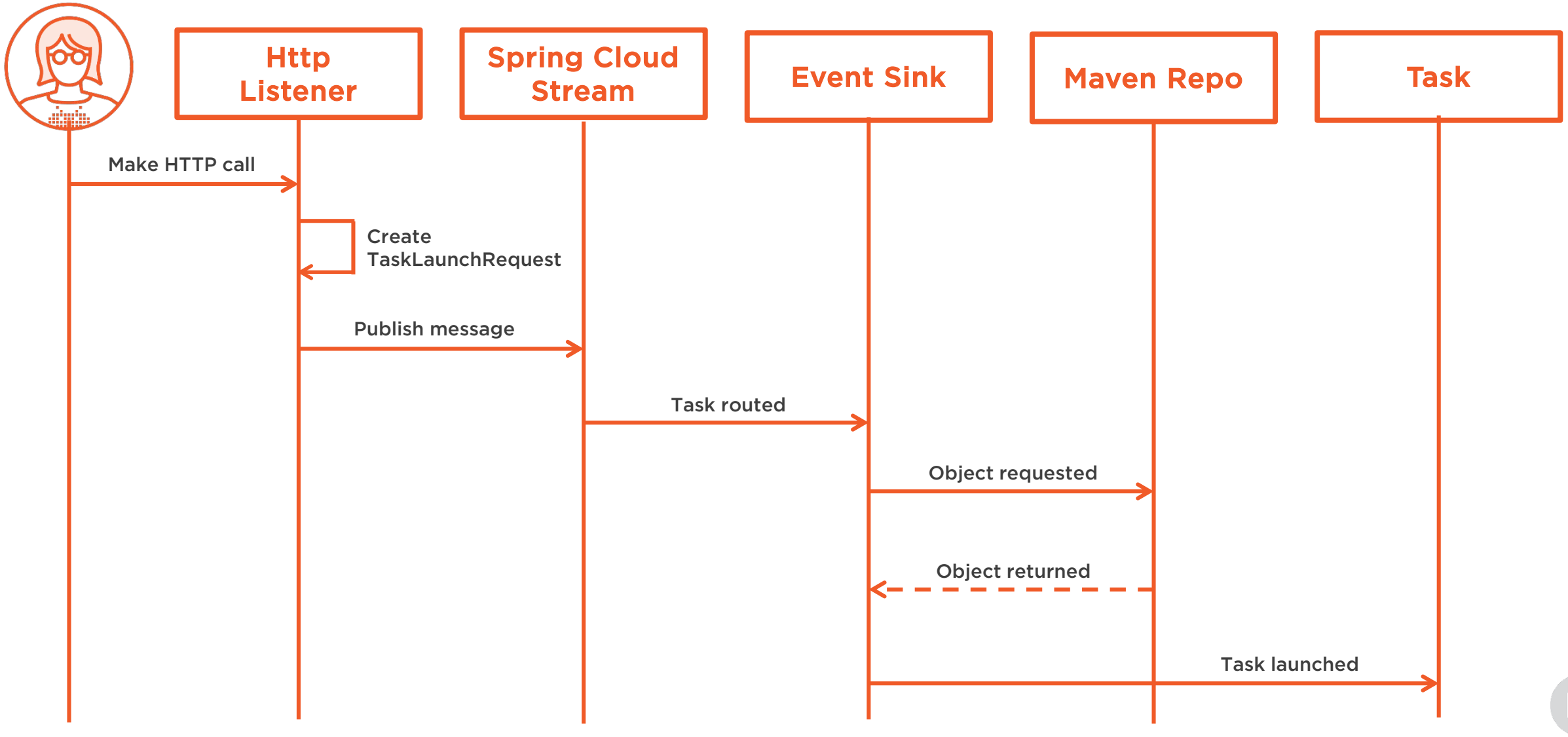
Update application properties

Call Task and observe stored results

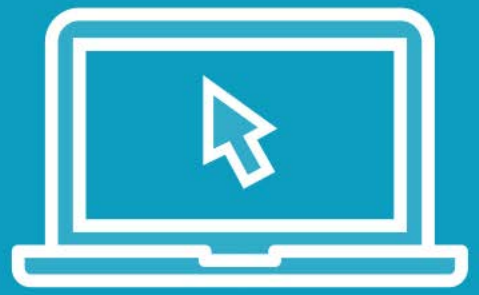
Options for Invoking Tasks

- ✓ **Run jar as Cron (scheduled) job**
- ✓ **Include as part of data pipeline**
- ✓ **Subscribe to event bus**
- ✓ **Directly invoked via custom Launcher**

Invoking Task via HTTP and Spring Cloud Stream



Demo



Create RabbitMQ environment

Build event sink

Build launcher

Submit request and see execution

Summary

- Overview
- The role of asynchronous processing in microservices
- Problems with the status quo
- Defining “serverless” computing
- Describing Spring Cloud Task
- Creating a Task
- Reviewing storage options for results
- Options for invoking tasks