KubeCon

North America 2017

nirmata

Distributed Workflow for Microservice-style Applications

Yun Qin, Software Engineer, Nirmata

yun@nirmata.com

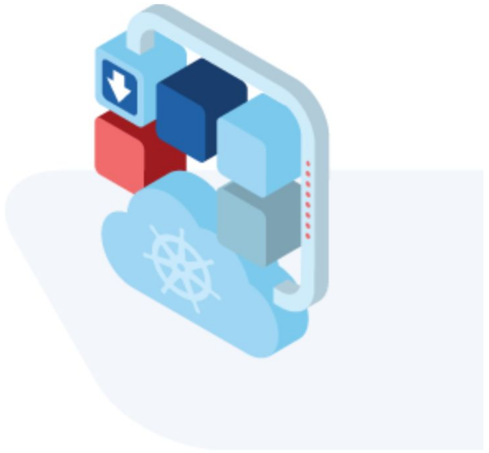
Microservices add complexity

- State management
- Workflow Execution
- Error handling

Image Credit: https://unsplash.com

nirmata

Overcome Pain Points for Microservices



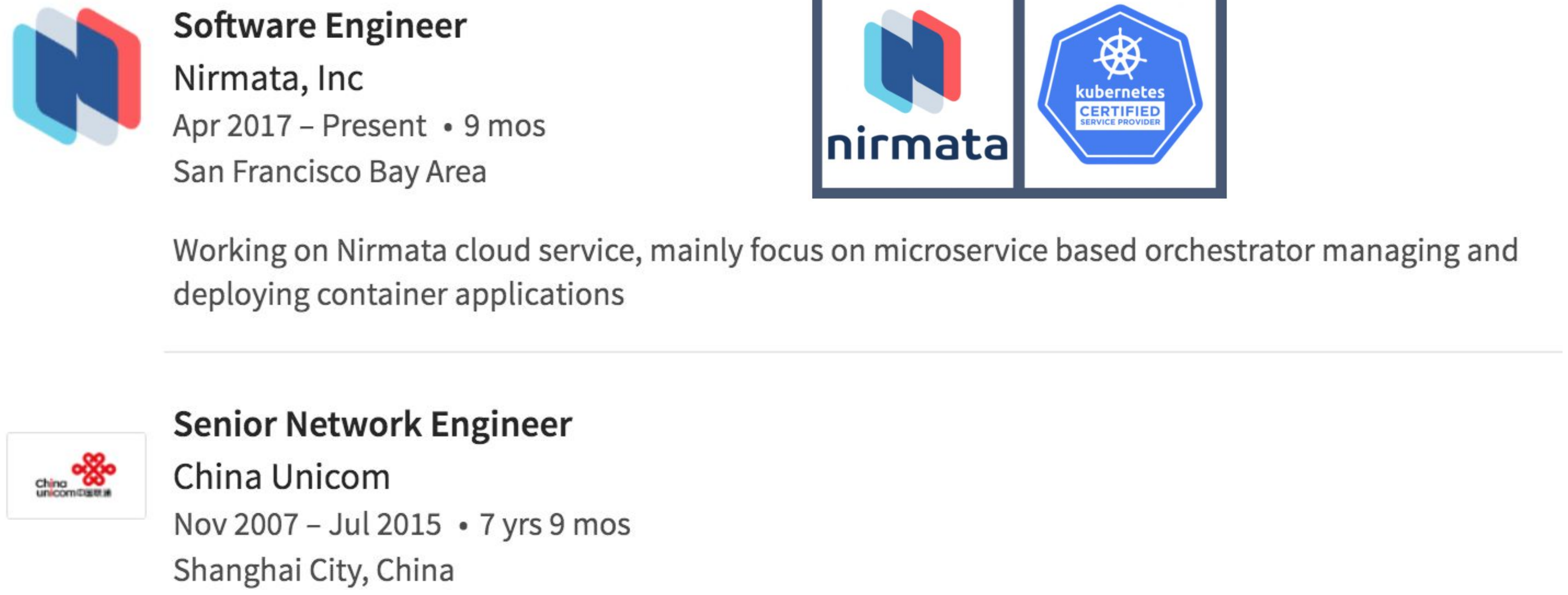
The Workflow Pattern makes life easier

Image Credit: https://unsplash.com

nirmata

About me

Yun Qin



Software Engineer
Nirmata, Inc
Apr 2017 – Present • 9 mos
San Francisco Bay Area

Working on Nirmata cloud service, mainly focus on microservice based orchestrator managing and deploying container applications

Senior Network Engineer
China Unicom
Nov 2007 – Jul 2015 • 7 yrs 9 mos
Shanghai City, China

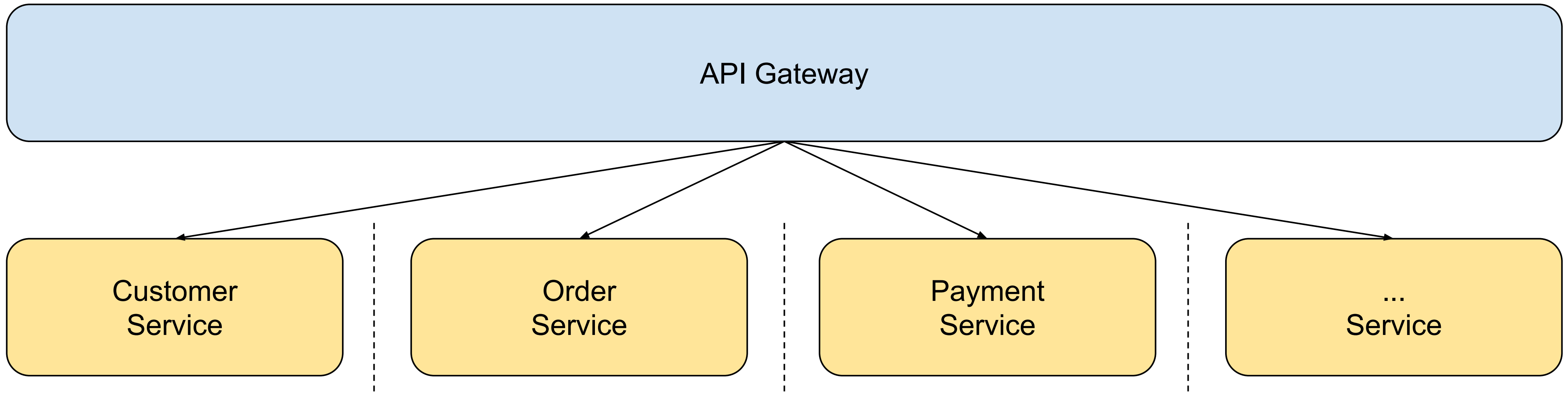
Presentation Goal

To introduce the Distributed Workflow pattern and its usage in Microservices-style applications using NirmataOSS workflow library

Agenda

- Microservice Architecture
- Workflow Overview
- Workflow Management on Microservices
- NirmataOSS Workflow
- Demo
- Other solutions

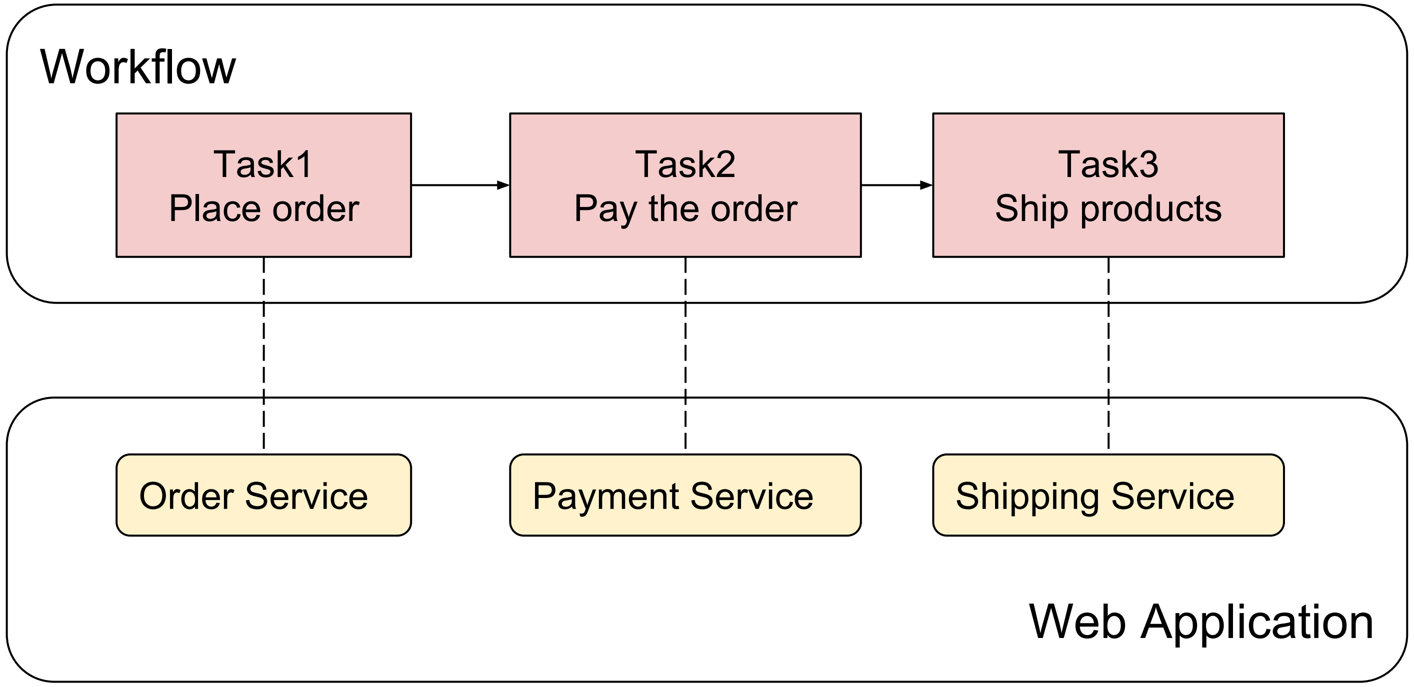
Microservice Architecture



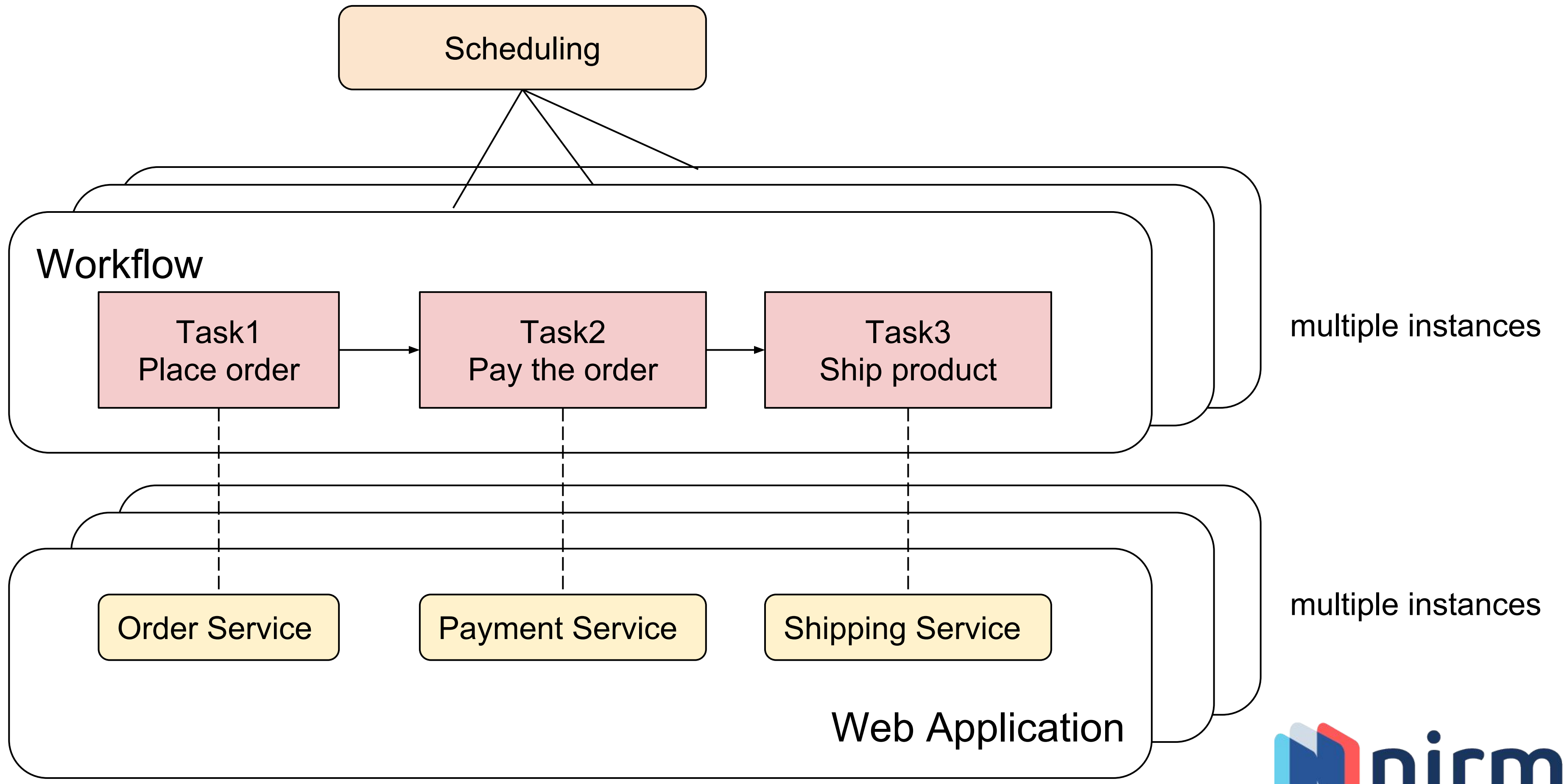
- Independent modular services
- Communicate through well-defined mechanism (e.g. REST api)

Workflow Overview

- Sequence of tasks
- Coordinated execution
- Different processing entities

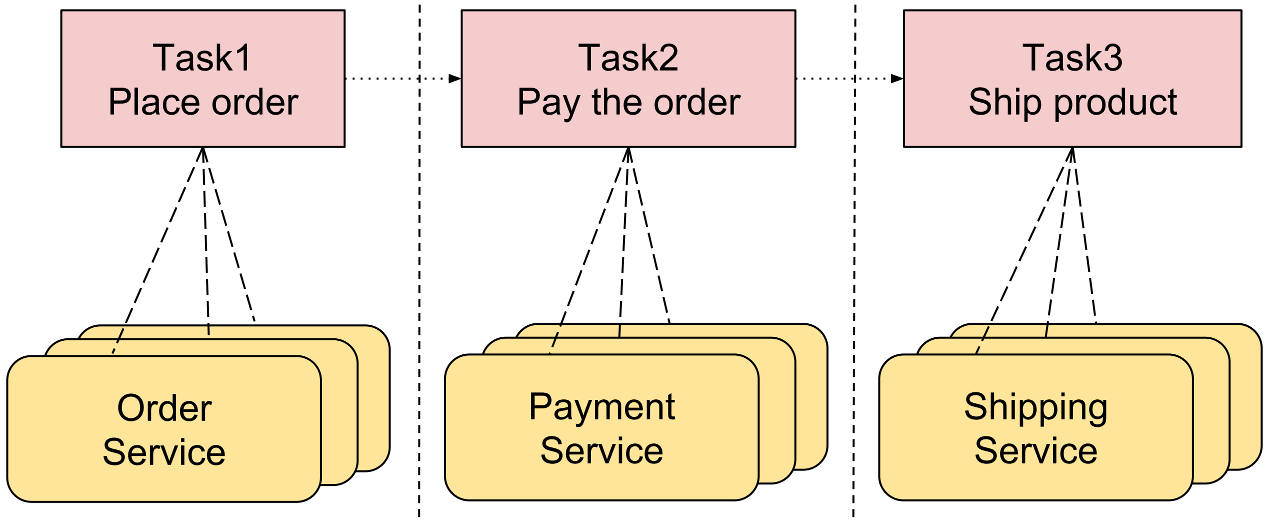


Workflow Overview (Distributed)



Workflow Management with Microservices

- Challenge
  - Tasks execution across multi-services
  - Distributed asynchronized environment
  - Dependencies between tasks
  - Complex logic handling



NirmataOSS Workflow

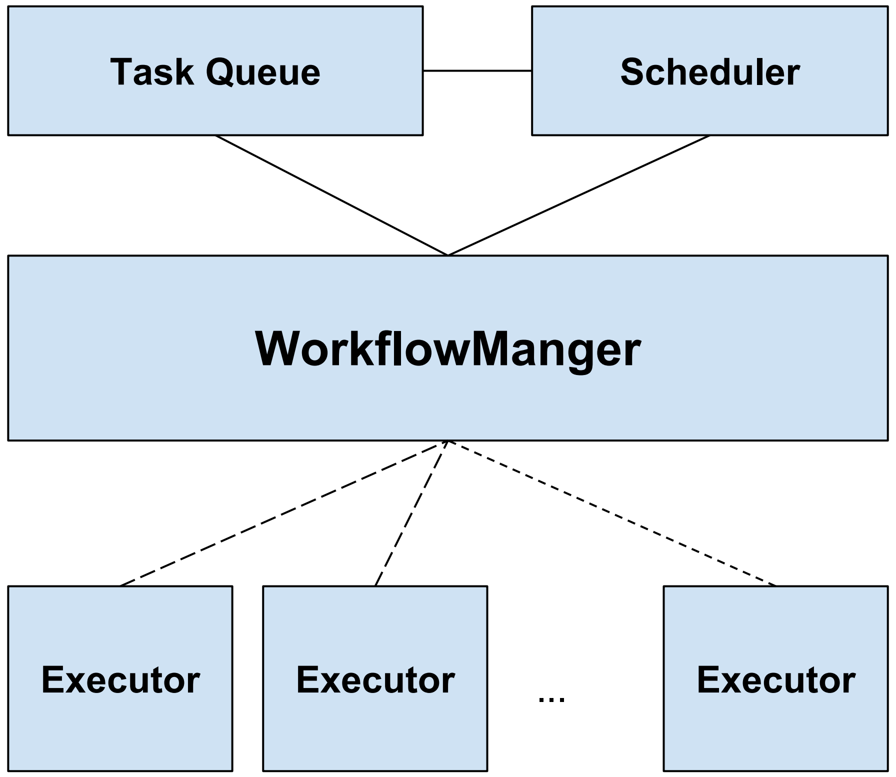
- Open source lib http://nirmataoss.github.io/workflow/
- Java based
- Apache ZooKeeper and Apache Curator based
- Lightweight and easy to use

NirmataOSS Workflow

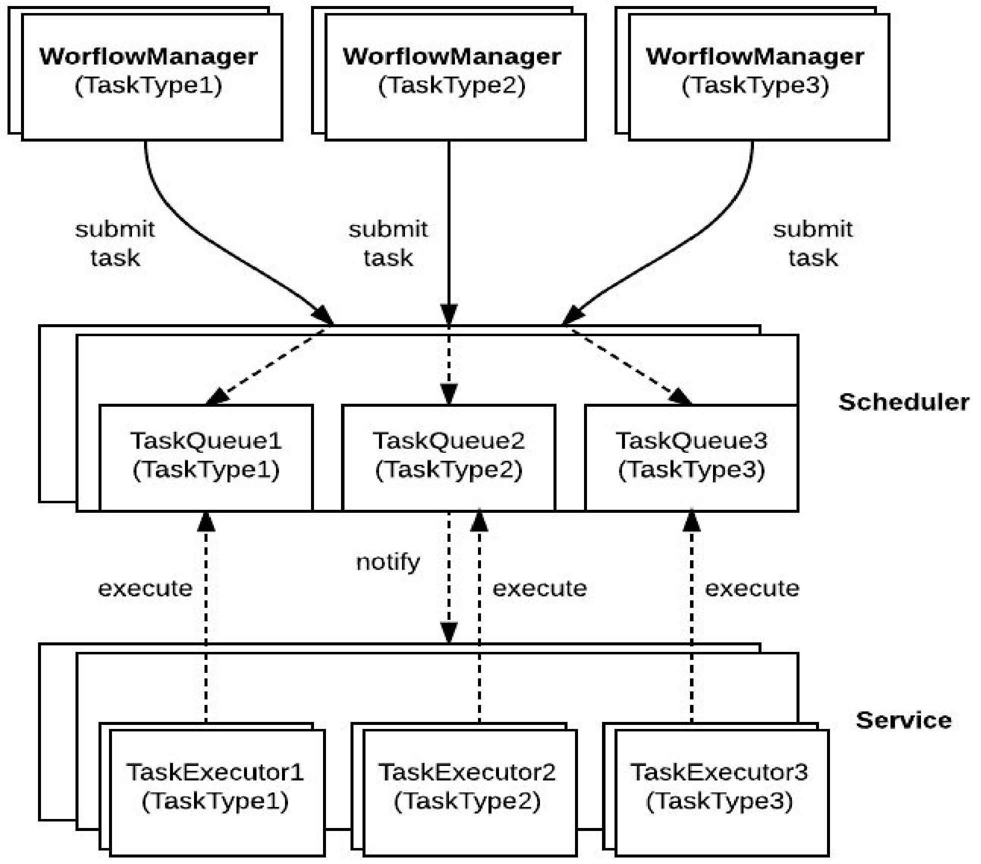
- Main Features
  - Task relationships management
  - Distributed scheduling
  - Task-types customization
  - Runtime cluster changes support
  - No Single point of failure

NirmataOSS Workflow

- Key Components
  - WorkflowManager
  - Scheduler
  - Task Queue
  - Task Executor

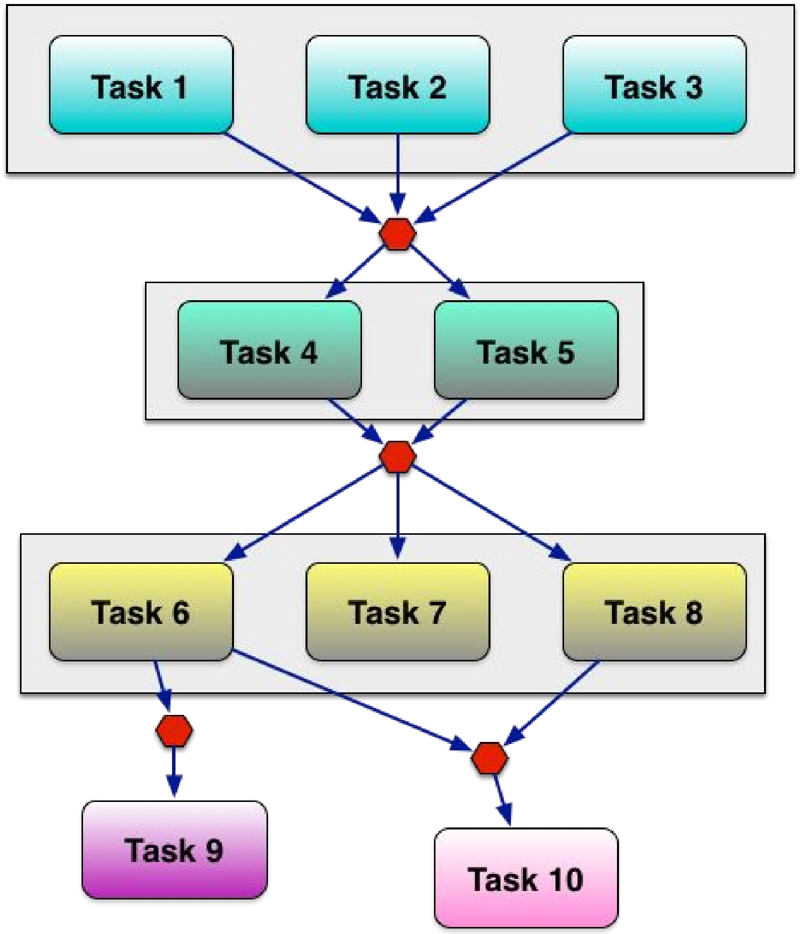


NirmataOSS Workflow


WorflowManager (TaskType1)
WorflowManager (TaskType2)
WorflowManager (TaskType3)
submit task
submit task
submit task
Scheduler
TaskQueue1 (TaskType1)
TaskQueue2 (TaskType2)
TaskQueue3 (TaskType3)
execute
notify
execute
execute
Service
TaskExecutor1 (TaskType1)
TaskExecutor2 (TaskType2)
TaskExecutor3 (TaskType3)

- Task Execution Model
  - Producer-Consumer based
  - Decentralized
  - Distributed
  - Asynchronous

NirmataOSS Workflow



- Multi-tasks Workflow Model
  - DAG task
  - Concurrent vs Sequential execution

nirmata

NirmataOSS Workflow

- How to build workflow

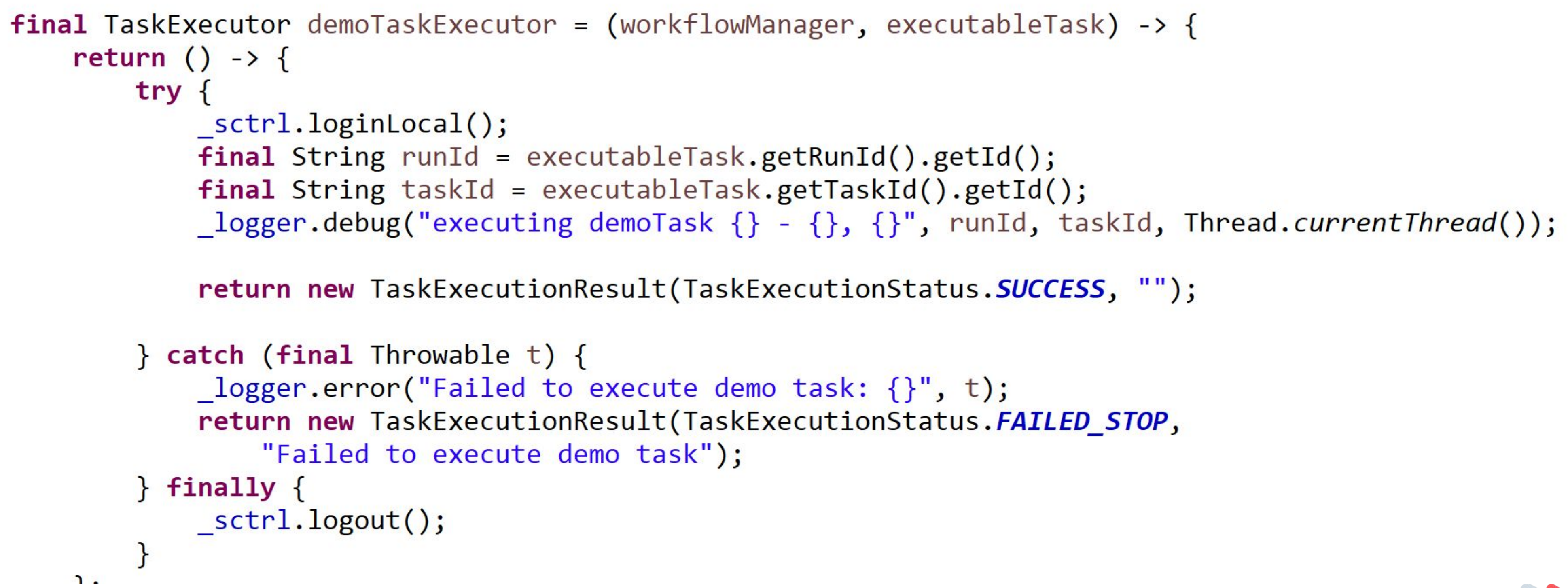
```
private WorkflowManager buildWorkflow() {
    Duration runPeriod = Duration.ofSeconds(60);
    AutoCleaner cleaner = new StandardAutoCleaner(Duration.ofMinutes(5));
    final WorkflowManagerBuilder workflowManagerBuilder = WorkflowManagerBuilder.builder().withCurator(
        _curator, _namespace, WORKFLOW_VERSION).withAutoCleaner(cleaner, runPeriod);
```

- Adding concurrent executor

```
workflowManagerBuilder.addingTaskExecutor(demoTaskExecutor, CONCURRENT_TASKS, DEMO_TASK_TYPE);
```

NirmataOSS Workflow

- Writing executor



```
final TaskExecutor demoTaskExecutor = (workflowManager, executableTask) -> {
    return () -> {
        try {
            _sctrl.loginLocal();
            final String runId = executableTask.getRunId().getId();
            final String taskId = executableTask.getTaskId().getId();
            _logger.debug("executing demoTask {} - {}, {}", runId, taskId, Thread.currentThread());

            return new TaskExecutionResult(TaskExecutionStatus.SUCCESS, "");

        } catch (final Throwable t) {
            _logger.error("Failed to execute demo task: {}", t);
            return new TaskExecutionResult(TaskExecutionStatus.FAILED_STOP,
                "Failed to execute demo task");
        } finally {
            _sctrl.logout();
        }
    };
```

Demo

Other solutions

- Netflix Conductor
  - a JSON DSL based blueprint that defines the execution flow.
- AWS Simple Workflow
  - a cloud workflow management application to coordinate applications across multiple machines.

Thank you !

Nirmata booth S61