**Министерство образования и науки Российской Федерации**

федеральное государственное автономное образовательное учреждение высшего образования

**“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**УНИВЕРСИТЕТ ИТМО**”

**Факультет программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА № 7**

**ПО ДИСЦИПЛИНЕ «ТЕХНОЛОГИИ ВЕБ-СЕРВИСОВ»**

Студент: Носков Антон Андреевич

Группа: P41142

Преподаватель: Дергачев Андрей Михайлович

Санкт-Петербург

2020

1.Техническое задание

Требуется разработать приложение, осуществляющее регистрацию сервиса в реестре jUDDI, а также поиск сервиса в реестре и обращение к нему. Рекомендуется реализовать консольное приложение, которое обрабатывает 2 команды. Итог работы первой команды – регистрация сервиса в реестре; вторая команда должна осуществлять поиск сервиса, а также обращение к нему.

2.Описание работы

Было реализовано консольное приложение реализующие 6 команд: Find all service, Find service by key, Find service by name, Call metod, Add service, Delete service.

Консольное приложение расположено на GitHub по адресу: https://github.com/anton-mix/WebLabs

3.Ответы на вопросы

1. В каком случае вы предпочтете сервис-ориентированную архитектуру вместо монолитной? Приведите ваши рассуждения по данному вопросу.

Сервис-ориентированную архитектуру стоит использовать в больших проектах для придания им гибкости. Со временем любая программа становится громоздкой и сложной. В неё становится тяжело добавлять новые технологии. Так же всё больше времени тратит поиск ошибок и их исправление.

Если монолитное приложение разбить на несколько сервисов, то размер программ уменьшится, станет возможна параллельная разработка и использование разного стека технологий для решения разных проблем.

2. В чем заключается смысл использования UDDI-реестров в промышленных SOA-системах? Можно ли обойтись без реестров?

В промышленные SOA-системы очень сложны. В них может существовать сотни сервисов. А так как такие системы быстро развиваются, каждый день появляются новые или меняются требования к старым. UDDI-реестр для таки систем необходим. Он не только предоставляет удобный механизм поиска этих сервисов по необходимым критериям, но и хранит техническую информацию о этих сервисах, а также информацию как к этим сервисам обратиться.

Возможно можно обойдись без UDDI-реестра если в кампании правильно и своевременно ведётся документация и записываются изменения, происходящие с сервисами. Но искать необходимый сервис таким способом будет сложнее.

3. Для чего используются такие компоненты SOA, как ESB? Можно ли обойтись без них?

ESB – это инструмент который занимается передачей сообщений между сервисами. ESB занимается маршрутизацией сообщений, отвечает за гарантированную доставку сообщений сервисам, предоставляет общий интерфейс общения между сервисами, может обрабатывать и агрегировать сообщения.

Без него можно обойтись если описать и разработать модель взаимодействия сервисов заранее. Но это редко бывает возможно так как современные системы быстро растут, а новые требования к ним появляются чуть ли не каждый день.

4. Предположим Вам нужно спроектировать public API для вашей системы. Что вы выберете, REST или SOAP? Какие будут основные критерии для выбора? Приведите ваши рассуждения по данному вопросу.

Всё зависит от того сколько будет пользователей у этого api и с какой целью они будут его использовать.

Главным преимуществом REST в отличие от soap является более меньшие издержки при общении между сервисами. REST может использовать использует множество разных форматов общения и размер сообщения REST из-за отсутствия стандартизации меньше чем у SOAP.

С другой стороны, если этот public API будет иметь множеством сервисов и интерфейсов для общения, стандартизация SOAP и возможность получить описание его сервисов будет гораздо ценнее чем большие издержки в общении и выбор падёт на SOAP.