

Оглавление

Предисловие	15
Для кого написана эта книга	15
Зачем я написал эту книгу	15
Мир микросервисов сегодня.	16
Структура книги.	16
Соглашения, принятые в этой книге.	18
Благодарности.	18
Об авторе	20
От издательства	21
Глава 1. Микросервисы.	22
Что же такое микросервисы.	23
Небольшие и нацеленные на то, чтобы хорошо справляться только с одной работой	23
Автономные	24
Основные преимущества	25
Технологическая разнородность	25
Устойчивость	26
Масштабирование.	27
Простота развертывания.	27
Решение организационных вопросов.	29
Компонуемость.	29
Оптимизация с целью последующей замены	30
А как насчет сервис-ориентированной архитектуры?	30
Другие технологии декомпозиции	31
Совместно используемые библиотеки	31
Модули.	32
Никаких универсальных решений.	33
Резюме	34

Глава 2. Архитектор развития.	35
Неверные сравнения	35
Эволюционное видение для архитектора	37
Зонирование	38
Принципиальный подход	40
Стратегические цели	40
Принципы.	40
Инструкции.	41
Объединение принципов и инструкций	41
Практический пример	42
Необходимый стандарт	42
Мониторинг	43
Интерфейсы	44
Архитектурная безопасность.	44
Управление посредством кода	45
Экземпляры	45
Подгоняемый шаблон сервиса.	45
Технические обязательства	47
Работа с исключениями	47
Руководство и ведущая роль центра	48
Формирование команды.	50
Резюме	50
Глава 3. Как моделировать сервисы	52
Представление MusicCorp.	52
Как создать хороший сервис	53
Слабая связанность	53
Сильное зацепление	53
Ограниченный контекст	54
Общие и скрытые модели	54
Модули и сервисы.	56
Преждевременная декомпозиция	56
Бизнес-возможности	57
Внизу сплошные черепахи	57
Обмен данными с точки зрения бизнес-концепций	59
Техническая граница	59
Резюме	61
Глава 4. Интеграция	62
Поиск идеальной интеграционной технологии	62
Уклонение от разрушающих изменений.	62
Сохранение технологической независимости применяемых API	62

Сохранение простоты использования сервиса потребителями	63
Скрытие внутренних деталей реализации	63
Взаимодействие с потребителями	63
Совместно используемая база данных	64
Сравнение синхронного и асинхронного стилей	66
Сравнение оркестрового и хореографического принципов	67
Удаленные вызовы процедуры	70
Технологическая связанность	71
Локальные вызовы не похожи на удаленные	71
Хрупкость	71
Неужели RPC настолько страшен?	73
REST	74
REST и HTTP	74
Гиперсреда как механизм определения состояния приложения . . .	75
JSON, XML или что-то другое?	77
Опасайтесь слишком больших удобств	78
Недостатки REST с использованием HTTP	79
Реализация асинхронной совместной работы на основе событий	80
Выбор технологии	80
Сложности асинхронных архитектур	82
Сервисы как машины состояний	83
Реактивные расширения	84
DRY и риски повторного использования кода в мире микросервисов	84
Доступ по ссылке	86
Управление версиями	88
Откладывание изменений на максимально возможный срок	88
Выявление критических изменений на самой ранней стадии	89
Использование семантического управления версиями	90
Существование различных конечных точек	90
Использование нескольких параллельных версий сервиса	92
Пользовательские интерфейсы	93
Движение в направлении к единым цифровым устройствам	94
Ограничения	95
API-композиция	95
Составление фрагментов пользовательского интерфейса	97
Внутренние интерфейсы, предназначенные для внешних интерфейсов	99
Гибридный подход	102
Интеграция с программами сторонних разработчиков	102
Отсутствие должного контроля	103
Адаптация	103

Тонкости интеграции	104
На наших собственных условиях	104
Шаблон Strangler (Дроссель)	107
Резюме	109
Глава 5. Разбиение монолита на части.	110
Все дело в стыках	110
Разбиение MusicCorp на части	111
Мотивы для разбиения монолита на части	112
Темпы изменений	112
Структура команды	112
Безопасность	112
Технология	113
Запутанные зависимости	113
База данных	113
Решение проблем	113
Пример 1: разрыв взаимоотношений, использующих внешние ключи	115
Пример 2: совместно используемые статичные данные	117
Пример 3: совместное использование данных	118
Пример 4: совместно используемые таблицы	119
Перестройка баз данных	120
Транзакционные границы	122
Повторная попытка	123
Отмена всей операции	124
Распределенные транзакции	124
Так что же делать?	125
Создание отчетов	126
База данных для создания отчетов	126
Извлечение данных посредством служебных вызовов	128
Программы перекачки данных	130
Альтернативные направления	132
Перекачка данных на основе событий	132
Перекачка данных на основе систем резервного копирования	133
Переход к реальности	134
Цена внесения изменений	134
Умение разбираться в основных причинах	135
Резюме	136
Глава 6. Развертывание	137
Краткое введение в непрерывную интеграцию	137
Отображение непрерывной интеграции на микросервисы	139

Сборочные конвейеры и непрерывная доставка	141
Артефакты для конкретных платформ	143
Артефакты операционных систем.	144
Настраиваемые образы	145
Образы как артефакты	148
Неизменяемые серверы	148
Среды.	149
Конфигурация сервиса.	150
Отображение сервиса на хост	151
Несколько сервисов на каждый хост	152
Приложения-контейнеры	154
Размещение по одному сервису на каждом хосте	156
Платформа в качестве услуги	157
Автоматизация	158
От физического к виртуальному.	159
Традиционная виртуализация.	159
Vagrant.	162
Контейнеры Linux	162
Docker	164
Интерфейс развертывания.	165
Резюме	168
Глава 7. Тестирование	169
Разновидности тестов	169
Области применения тестов.	171
Блочные тесты	172
Тесты сервиса	173
Сквозные тесты	174
Компромиссы	174
Что и в каком объеме проводить.	175
Реализация тестов сервисов	176
Использование имитации или применение заглушки	176
Более интеллектуальный сервис-заглушка	177
Сложности, связанные со сквозными тестами.	178
Недостатки сквозного тестирования	179
Тесты со странностями, не дающие четкого представления об источнике сбоя	179
Кто создает все эти тесты.	180
Насколько продолжительными бывают тесты	181
Сплошное нагромождение	182
Метаверсия	183

Тестируйте маршруты, а не истории.	183
Тесты на основе запросов потребителей, спасающие ситуацию	184
Раст	186
О переговорах	187
А нужно ли вообще пользоваться сквозными тестами?	188
Тестирование после перевода в производственный режим работы	188
Отделение развертывания от выпуска.	189
Канареечный выпуск	191
Что важнее: среднее время восстановления работоспособности или среднее время безотказной работы?.	192
Межфункциональное тестирование	193
Резюме	195
Глава 8. Мониторинг	197
Один сервис на одном сервере.	198
Один сервис на нескольких серверах	199
Несколько сервисов на нескольких серверах	200
Журналы, журналы и еще журналы....	201
Отслеживание показателей сразу нескольких сервисов	201
Рабочие показатели сервисов	203
Искусственный мониторинг	204
Реализация семантического мониторинга.	205
Идентификаторы взаимосвязанности	205
Каскадные сбои	208
Стандартизация.	208
Расчет на аудиторию	209
Перспективы	210
Резюме	211
Глава 9. Безопасность.	213
Аутентификация и авторизация	213
Общепринятые реализации технологии единого входа	214
Шлюз технологии единого входа.	215
Авторизация с высокой степенью детализации	217
Взаимная аутентификация и авторизация сервисов	217
Разрешение всего в пределах периметра	218
Стандарт HTTP(S) Basic Authentication	218
Использование SAML или OpenID Connect	219
Клиентские сертификаты	220
HMAC через HTTP	220
API-ключи.	222
Проблема помощника	222

Безопасность данных, находящихся в покое	225
Пользуйтесь хорошо известными средствами	225
Все зависит от ключей	226
Выберите защищаемые объекты	226
Расшифровка по требованию	227
Шифровка резервных копий	227
Глубоко эшелонированная оборона	227
Брандмауэры	227
Регистрация	228
Система обнаружения (и предотвращения) вторжений	228
Обособление сетей	228
Операционная система	229
Рабочий пример	230
Проявляйте сдержанность	232
Человеческий фактор	233
Золотое правило	233
Создание системы безопасности	234
Внешняя проверка	234
Резюме	235
Глава 10. Закон Конвея и проектирование систем.	236
Доказательства	236
Организации со слабыми и сильными связями	237
Windows Vista	237
Netflix и Amazon	237
Так что же со всем этим делать?	238
Адаптация к направлениям обмена данными	238
Владение сервисом	239
Побудительные мотивы для создания общих сервисов	240
Слишком большие трудности разбиения на части	240
Команды разработки функций	240
Узкие места, касающиеся вопросов поставки	241
Семейственный открытый код	242
Роль кураторов	243
Зрелость	243
Инструментарий	243
Ограниченные контексты и структуры команд	244
Осиротевшая служба?	244
Конкретный пример: RealEstate.com.au	245
Закон Конвея наоборот	247
Люди	248
Резюме	249

Глава 11. Масштабирование микросервисов	250
Сбои могут происходить везде	250
Слишком много — это сколько?	251
Снижение уровня функциональных возможностей	252
Архитектурные меры безопасности	253
Антихрупкая организация	256
Настройки времени ожидания	257
Предохранители	257
Переборки	259
Изолированность	261
Идемпотентность	261
Масштабирование	262
Наращивание мощностей	262
Разделение рабочих нагрузок	263
Распределение риска	264
Балансировка нагрузки	265
Системы на основе исполнителей	267
Начинаем все заново	268
Масштабирование баз данных	268
Доступность сервиса против долговечности данных	269
Масштабирование для считываний	269
Масштабирование для производства записей	270
Совместно используемые инфраструктуры баз данных	271
CQRS	272
Кэширование данных	273
Кэширование на стороне клиента, прокси-сервере и стороне сервера	273
Кэширование при использовании технологии HTTP	274
Кэширование, проводимое для операций записи	276
Кэширование в целях повышения отказоустойчивости	276
Скрытие источника	276
Не нужно ничего усложнять	278
Отравление кэша: предостережение	278
Автоматическое масштабирование	279
Теорема CAP	280
Принесение в жертву согласованности	282
Принесение в жертву доступности	282
А как насчет принесения в жертву терпимости к разделению?	283
AP или CP?	283
Это не все или ничего	284
И реальный мир	285

Обнаружение сервисов	285
DNS	286
Динамическая регистрация сервисов	287
Zookeeper	288
Consul	289
Eureka	290
Прокатка собственной системы	290
Не забывайте про людей!	291
Документирующие сервисы	291
Swagger	291
HAL и HAL-браузер	292
Самостоятельно описываемая система	293
Резюме	294
Глава 12. Коротко обо всем	295
Принципы микросервисов	295
Модель, построенная вокруг бизнес-концепций	296
Внедрение культуры автоматизации	296
Скрытие подробности внутренней реализации	297
Всесторонняя децентрализация	297
Независимое развертывание	298
Изолирование сбоев	298
Всестороннее наблюдение	299
А когда не следует применять микросервисы?	299
Напутствие	300