
Оглавление

Предисловие	20
Кому стоит прочитать эту книгу.....	20
Почему я написал эту книгу.....	20
Что изменилось с момента выхода первого издания.....	21
Навигация по книге	22
Часть I. Основы	22
Часть II. Реализация	23
Часть III. Люди	24
Условные обозначения.....	24
Благодарности	25
От издательства.....	26

ЧАСТЬ I. ОСНОВЫ

Глава 1. Что такое микросервисы	28
Первый взгляд на микросервисы	28
Ключевые понятия микросервисов.....	31
Независимое развертывание	31
Моделирование вокруг предметной области бизнеса.....	32
Контроль над ситуацией	33
Размер.....	34
Гибкость	35
Согласование архитектуры и структуры организации.....	36
Монолит	40
Однопроцессный монолит.....	40
Модульный монолит	41
Распределенный монолит	42
Монолиты и конфликт доставки.....	43
Преимущества монолитов	43
Технологии, обеспечивающие развитие	44
Агрегирование логов и распределенная трассировка	44
Контейнеры и Kubernetes	45

Потоковая передача данных	46
Публичное облако и бессерверный подход	47
Преимущества микросервисов	47
Технологическая неоднородность	48
Надежность	49
Масштабирование	49
Простота развертывания	51
Согласованность рабочих процессов в организации	51
Компонуемость	51
Слабые места микросервисов	52
Опыт разработчика	52
Технологическая перегрузка	53
Стоимость	53
Отчетность	54
Мониторинг и устранение неполадок	55
Безопасность	55
Тестирование	55
Время ожидания	56
Согласованность данных	56
Стоит ли вам использовать микросервисы?	57
Кому микросервисы не подойдут	57
Где микросервисы хорошо работают	59
Резюме	60
Глава 2. Как моделировать микросервисы	61
Представляем MusicCorp	61
Что делает границу микросервиса качественной	62
Скрытие информации	62
Связность	64
Связанность	64
Взаимодействие связанности и связности	65
Типы связанности	65
Предметная связанность	67
Сквозная связанность	69
Общая связанность	72
Связанность по содержимому	76
Немного предметно-ориентированного проектирования	77
Единый язык	78
Агрегат	79

Ограниченный контекст	82
Сопоставление агрегатов и ограниченных контекстов с микросервисами.....	85
Метод Event Storming	86
Аргументы в пользу предметно-ориентированного проектирования микросервисов.....	88
Альтернативы границам предметной области бизнеса.....	89
Волатильность	89
Данные.....	90
Технологии.....	92
Организационный подход.....	93
Смешивание моделей и исключений.....	95
Резюме.....	96
Глава 3. Разделение монолита на части.....	97
Осознайте цель.....	97
Постепенный переход.....	98
Монолит не всегда плохой вариант	99
Опасность преждевременной декомпозиции.....	99
Что отделить в первую очередь	100
Декомпозиция по слоям	102
Сначала код	102
Сначала данные.....	103
Полезные шаблоны декомпозиции	104
Шаблон «Душитель»	104
Параллельное выполнение	105
Шаблон переключаемых функций.....	105
Проблемы декомпозиции данных.....	106
Производительность	106
Целостность данных	109
Транзакции	109
Инструментарий	110
База данных отчетов	110
Резюме.....	112
Глава 4. Стили взаимодействия микросервисов	113
От внутрипроцессного к межпроцессному	113
Производительность	114
Изменение интерфейсов	115
Обработка ошибок.....	115

Технология межпроцессного взаимодействия: так много вариантов выбора.....	117
Стили взаимодействия микросервисов.....	118
Смешивание и сочетание	119
Шаблон: синхронная блокировка.....	119
Преимущества.....	120
Недостатки.....	120
Где использовать	121
Шаблон: асинхронная неблокирующая связь	123
Преимущества.....	123
Недостатки.....	124
Где использовать	125
Шаблон: связь через общие данные	126
Реализация	126
Преимущества.....	128
Недостатки.....	128
Где использовать	128
Шаблон: связь «запрос — ответ».....	129
Реализация: синхронная или асинхронная.....	130
Где использовать	133
Шаблон: событийное взаимодействие.....	133
Реализация	135
Что входит в событие	136
Где использовать	139
Действуйте с осторожностью	140
Резюме.....	142

ЧАСТЬ II. РЕАЛИЗАЦИЯ

Глава 5. Реализация коммуникации микросервисов	144
В поисках идеальной технологии	144
Упростите обратную совместимость.....	144
Сделайте свой интерфейс выразительным.....	145
Следите за тем, чтобы ваши API не зависели от технологий	145
Сделайте свой сервис простым для потребителей	145
Скройте детали внутренней реализации	146
Выбор технологий	146
Удаленные вызовы процедур	146
REST	151

GraphQL.....	157
Брокеры сообщений.....	159
Форматы сериализации.....	164
Текстовые форматы.....	165
Двоичные форматы.....	165
Схемы.....	166
Структурные и семантические разрывы контрактов.....	167
Стоит ли использовать схемы.....	167
Обработка изменений между микросервисами.....	168
Избегание критических изменений.....	169
Наращивание изменений.....	169
Устойчивое считывание.....	169
Правильные технологии.....	171
Явный интерфейс.....	171
Своевременно выявляйте случайные критические изменения.....	173
Управление критическими изменениями.....	174
Поэтапное развертывание.....	174
Сосуществование несовместимых версий микросервиса.....	174
Эмулируйте старый интерфейс.....	176
Какой подход предпочтителен.....	177
Общественный договор.....	178
Отслеживание использования.....	179
Крайние меры.....	179
DRY и опасности повторного использования кода в мире микросервисов.....	180
Совместное использование кода через библиотеки.....	180
Обнаружение сервиса.....	182
Система доменных имен (DNS).....	183
Динамические реестры сервисов.....	185
Не забывайте о людях!.....	188
Сервисные сети и API-шлюзы.....	188
API-шлюзы.....	189
Сервисные сети.....	192
А как насчет других протоколов?.....	196
Документирование сервисов.....	196
Явные схемы.....	197
Самоописывающаяся система.....	198
Резюме.....	200

Глава 6. Рабочий поток	202
Транзакции базы данных	202
Транзакции ACID	202
Все еще ACID, но с недостаточной атомарностью?	204
Распределенные транзакции — двухфазная фиксация	206
Просто скажите «нет» распределенным транзакциям.....	208
Саги	209
Режимы сбоя саги	211
Реализация саг	216
Саги в сравнении с распределенными транзакциями.....	222
Резюме.....	223
Глава 7. Сборка	224
Краткое введение в непрерывную интеграцию	224
Вы действительно выполняете CI?.....	225
Модели ветвления.....	226
Конвейеры сборки и непрерывная доставка.....	228
Инструментарий	230
Компромиссы и среды выполнения	231
Создание артефакта	231
Сопоставление исходного кода и сборок с микросервисами.....	233
Один гигантский репозиторий — одна гигантская сборка.....	233
Шаблон: один репозиторий на один микросервис (то есть мультирепозиторий)	235
Шаблон: монорепозиторий	239
Какой подход использовал бы я?	245
Резюме.....	246
Глава 8. Развертывание	247
От логического к физическому	247
Несколько экземпляров.....	248
База данных.....	250
Среды выполнения.....	253
Принципы развертывания микросервисов	256
Изолированное выполнение	257
Сосредоточьтесь на автоматизации	259
Инфраструктура как код	260
Управление желаемым состоянием.....	263

Варианты развертывания	266
Физические машины	267
Виртуальные машины.....	268
Контейнеры	270
Контейнеры приложений	276
Платформа как услуга (PaaS).....	277
Функция как услуга (FaaS).....	278
Какой вариант развертывания подходит именно вам.....	286
Kubernetes и оркестрация контейнеров	288
Пример для контейнерной оркестровки	288
Упрощенный взгляд на концепции Kubernetes	289
Мультиарендность и федерация.....	291
Федерация нативных облачных вычислений.....	294
Платформы и мобильность.....	295
Helm, Operator и CRD.....	296
Knative.....	297
Будущее.....	298
Стоит ли вам его использовать	298
Поэтапная доставка.....	299
Отделение развертывания от релиза	300
Переходим к поэтапной доставке.....	301
Переключатели функций	301
Канареечный релиз.....	302
Параллельное выполнение	303
Резюме.....	304
Глава 9. Тестирование	306
Типы тестов.....	306
Охват тестирования	309
Модульное тестирование.....	310
Сервисное тестирование	311
Сквозное тестирование.....	312
Компромиссы	313
Внедрение сервисных тестов	314
Макетирование или заглушки	315
Более самостоятельная сервис-заглушка.....	315
Внедрение (этих хитрых) сквозных тестов	316
Хрупкие и flaky-тесты.....	318

Кто пишет эти сквозные тесты	319
Как долго должны выполняться сквозные тесты.....	322
Великое нагромождение	323
Метаверсия	323
Отсутствие возможности независимого тестирования	324
Следует ли избегать сквозных тестов	325
Контрактное тестирование.....	325
Заключительное слово	329
Опыт разработчика	329
От предварительного тестирования к эксплуатационному.....	330
Виды эксплуатационных тестов.....	331
Среднее время восстановления превышает среднее время между отказами?.....	332
Кросс-функциональное тестирование	333
Тесты производительности.....	334
Тесты надежности.....	336
Резюме.....	337
Глава 10. От мониторинга к наблюдаемости.....	338
Сбой, паника и замешательство	338
Один микросервис — один сервер	339
Один микросервис — несколько серверов.....	340
Несколько микросервисов — несколько серверов	341
Наблюдаемость и мониторинг	342
Столпы наблюдаемости? Не так быстро.....	343
Строительные блоки для наблюдаемости	344
Агрегация логов	345
Агрегация метрики	355
Распределенная трассировка.....	358
Все ли у нас в порядке?	361
Оповещение.....	363
Семантический мониторинг	367
Тестирование в эксплуатации.....	369
Стандартизация.....	372
Выбор инструментов	373
Демократичность	373
Простота интеграции.....	374
Обеспечение контекста	374

Своевременность	374
Подходит для вашего масштаба.....	375
Эксперт в машине.....	375
Приступая к работе	377
Резюме.....	377
Глава 11. Безопасность.....	379
Основные принципы	380
Принцип наименьших привилегий.....	381
Глубокая оборона.....	381
Автоматизация.....	383
Встраивание безопасности в процесс доставки.....	383
Пять функций кибербезопасности.....	384
Выявление.....	385
Защита.....	387
Определение	387
Реакция.....	387
Восстановление.....	388
Основы безопасности приложений.....	388
Учетные данные.....	388
Исправление.....	395
Резервное копирование (бэкап).....	398
Повторная сборка (ребилд).....	399
Безусловное или нулевое доверие	400
Безусловное доверие	401
Нулевое доверие	401
Доверие — это спектр возможных вариантов	402
Защита данных.....	404
Данные в процессе передачи	404
Данные в состоянии покоя.....	407
Аутентификация и авторизация	410
Аутентификация между сервисами	411
Аутентификация человека	411
Общие реализации единого входа.....	412
SSO-шлюз.....	413
Детализированная авторизация	414

Проблема иерархии полномочий	415
Авторизация в вышестоящих элементах, централизованная	417
Децентрализованная авторизация	417
Веб-токены JSON	418
Резюме	422
Глава 12. Отказоустойчивость	423
Что такое отказоустойчивость	423
Надежность	424
Восстановление	425
Стабильная расширяемость	426
Непрерывная адаптивность	426
Микросервисная архитектура	427
Сбои повсюду	427
Слишком много — это сколько?	429
Снижение функциональности	430
Шаблоны стабильности	431
Тайм-ауты	434
Повторные попытки	436
Переборки	437
Автоматические выключатели	438
Изоляция	441
Избыточность	442
Промежуточное ПО	443
Идемпотентность	443
Распределение рисков	445
Теорема CAP	446
Жертвует согласованностью	447
Жертвует доступностью	448
Жертвует устойчивостью к разделению	449
AP или CP?	449
Это не «все или ничего»	450
А теперь реальный мир	450
Хаос-инжиниринг	451
Игровые дни	452
Эксперименты в эксплуатационной среде	453
От надежности к запредельному	453

Поиск виновных.....	454
Резюме.....	455
Глава 13. Масштабирование	457
Четыре оси масштабирования.....	457
Вертикальное масштабирование.....	458
Горизонтальное дублирование	461
Разделение данных.....	464
Функциональная декомпозиция	469
Сочетание моделей	471
Начните с малого	472
Кэширование.....	474
Для производительности	475
Для масштабирования	475
Для надежности.....	476
Где кэшировать.....	476
Аннулирование	482
Свежесть или оптимизация	488
Отравление кэша: поучительная история	488
Автоматическое масштабирование.....	489
Начинаем все сначала.....	491
Резюме.....	492

ЧАСТЬ III. ЛЮДИ

Глава 14. Пользовательские интерфейсы.....	494
К цифровым технологиям.....	495
Модели владения	495
Драйверы для специализированных фронтенд-команд	497
На пути к потоковым командам.....	498
Обмен специалистами.....	499
Обеспечение согласованности	501
Решение технических проблем.....	502
Шаблон: монолитный интерфейс.....	503
Когда использовать	504
Шаблон: микрофронтенды.....	505
Реализация	505
Когда использовать	505

Шаблон: декомпозиция на основе страниц.....	507
Где использовать	508
Шаблон: декомпозиция на основе виджетов.....	509
Реализация	510
Когда использовать	513
Ограничения.....	514
Шаблон: центральный объединяющий шлюз	515
Владение.....	517
Различные типы пользовательских интерфейсов.....	517
Многочисленные проблемы	519
Когда использовать	520
Шаблон: бэкенд для фронтенда (BFF).....	520
Сколько должно быть BFF.....	522
Повторное использование и BFF	524
BFF для настольных веб-сайтов и не только	527
Когда использовать	529
GraphQL.....	529
Гибридный подход	531
Резюме.....	531
Глава 15. Организационные структуры.....	532
Слабо связанные организации	532
Закон Конвея	534
Подтверждение	534
Размер команды.....	536
Понимание закона Конвея	537
Маленькие команды, большая организация.....	538
Об автономии.....	539
Сильное или коллективное владение	541
Сильное владение.....	541
Коллективное владение	543
Уровень команд или уровень организации.....	544
Баланс моделей	544
Команды поддержки.....	545
Профессиональные сообщества	547
Платформа	548
Общие микросервисы.....	551
Слишком трудно разделить.....	551

Сквозные изменения	552
Узкие места в доставке.....	553
Решение с открытым исходным кодом внутри компании.....	554
Роль доверенных коммиттеров	554
Завершенность	555
Инструментарий	555
Подключаемые модульные микросервисы	555
Обзоры изменений.....	558
Осиротевший сервис.....	562
Тематическое исследование: realestate.com.au.....	562
Географическое распределение	564
Закон Конвея наоборот	566
Люди	567
Резюме.....	568
Глава 16. Эволюционный архитектор.....	569
Что в имени твоём?.....	569
Что такое архитектура ПО	571
Делая изменения возможными	573
Эволюционное видение для архитектора	574
Определение границ системы	575
Социальная конструкция.....	578
Обитаемость	579
Принципиальный подход.....	580
Стратегические цели.....	581
Принципы.....	581
Методы	582
Сочетание принципов и методов.....	582
Пример из жизни	583
Руководство эволюционной архитектурой	584
Архитектура в потоковой организации.....	585
Создание команды.....	588
Требуемый стандарт.....	588
Мониторинг.....	589
Интерфейсы	589
Архитектурная безопасность	590
Управление и мощная дорога.....	590
Примеры.....	591

Адаптированный шаблон микросервиса.....	592
Мощная дорога при масштабировании	593
Технический долг.....	594
Обработка исключений.....	595
Резюме.....	595
Послесловие: подведем итог основных тем.....	597
Что такое микросервисы	597
Переход к микросервисам.....	598
Стили взаимодействия.....	599
Рабочий поток.....	601
Сборка	601
Развертывание.....	602
Тестирование.....	603
Мониторинг и наблюдаемость.....	603
Безопасность.....	604
Отказоустойчивость.....	604
Масштабирование	605
Пользовательские интерфейсы.....	606
Организация	606
Архитектура	607
Дополнительная литература.....	608
Взгляд в будущее	608
Заключительные слова	609
Библиография.....	610
Глоссарий.....	615
Об авторе	620
Иллюстрация на обложке.....	621

Предисловие

Архитектурный стиль микросервисов — это подход к распределенным системам, при котором используются небольшие сервисы, каждый из которых можно изменять, развертывать и выпускать независимо друг от друга. Для организаций, переходящих к менее связанным между собой системам, с автономными командами, предоставляющими функциональность, ориентированную на пользователя, микросервисы значительно повышают эффективность. Помимо этого, они обеспечивают огромное количество вариантов построения систем, дают впечатляющую гибкость, позволяя системе изменяться в соответствии с потребностями пользователей.

Однако микросервисы обладают и существенными недостатками: будучи распределенной системой, они создают множество сложностей, которые могут поставить в тупик даже опытных разработчиков.

В этой книге собраны идеи с конкретными примерами из реальной жизни, которые помогут понять, подходят ли вам микросервисы.

Кому стоит прочитать эту книгу

Область применения данной книги широка, как и возможности микросервисных архитектур. Таким образом, это издание должно понравиться людям, интересующимся аспектами проектирования, разработки, развертывания, тестирования и обслуживания систем. Те из вас, кто уже вступил на путь создания мелкомодульных архитектур для использования в новых проектах или в рамках декомпозиции существующей монолитной системы, найдут здесь множество практических рекомендаций. Это руководство также поможет тем, кто хочет тщательнее разобраться в микросервисах и наконец определиться в необходимости их применения.

Почему я написал эту книгу

С одной стороны, я написал эту книгу, потому что хотел убедиться, что информация в первом издании остается актуальной, точной и полезной. Первое издание появилось, потому что на тот момент были действительно интересные идеи, которыми я мечтал поделиться. С самого начала я писал о микросервисах

с достаточно объективной точки зрения, потому что не работал на крупного поставщика технологий. Я не продавал людям решения и надеялся, что не продавал и микросервисы, — мне просто нравилось разбираться в них и находить способы их более широкого использования.

Откровенно говоря, второе издание я написал по двум причинам. Во-первых, появилось ощущение, что на этот раз смогу сделать работу лучше: я больше узнал и, надеюсь, стал немного лучше как писатель. Во-вторых, я чувствую свою ответственность за популяризацию описанных в книге идей и поэтому хотел попробовать изложить их подробнее. Микросервисы стали для многих архитектурным выбором по умолчанию, который, на мой взгляд, трудно обосновать, поэтому я бы хотел поделиться своим видением причин.

В книге я не настаиваю на повсеместном использовании микросервисов, но и не отговариваю вас от их применения. Для меня важно донести до вас все изученные мной плюсы и минусы данного подхода.

Что изменилось с момента выхода первого издания

Первое издание я писал примерно год, на протяжении 2014 года, а выпущено оно было уже в феврале 2015-го. Это было в самом начале истории микросервисов, по крайней мере с точки зрения понимания этого термина широкими кругами в отрасли. С тех пор микросервисы стали популярны настолько, что я и предположить не мог. Чем популярнее становилась данная отрасль, тем больше появлялось возможностей и технологий для ее реализации.

По мере того как я работал с большим количеством команд после выхода первого издания, я совершенствовал свои знания о микросервисах. Иногда это означало, что идеи, существовавшие только на периферии моего сознания (например, скрытие информации), становились более ясными как основополагающие концепции, требующие более широкого освещения. Иногда новые технологии предоставляют не только новые решения, но и дополнительные сложности. Видя, как много людей стекаются в Kubernetes в надежде, что эта платформа поможет решить все их проблемы с микросервисными архитектурами, я, безусловно, задумался.

Кроме того, я написал первое издание книги «Создание микросервисов», чтобы не только рассказать о микросервисах, но и продемонстрировать, как этот архитектурный подход меняет суть разработки программного обеспечения (ПО). Поэтому, более глубоко изучив вопросы, связанные с безопасностью и отказоустойчивостью, я обнаружил, что хочу подробнее остановиться на тех темах, которые становятся все более важными для современной разработки программного обеспечения.

Таким образом, в этом, втором издании я потратил больше времени на подготовку наглядных примеров. Каждая глава была пересмотрена, и каждое предложение проанализировано. От первого издания осталось не так уж много с точки зрения непосредственно текста, но все идеи сохранились. Я старался излагать свои мысли более ясно, в то же время признавая существование нескольких способов решения проблемы. Это привело к более широкому описанию межпроцессной коммуникации, которое теперь занимает три главы. Я также потратил больше времени на изучение влияния применения таких технологий, как контейнеры, Kubernetes и бессерверные вычисления. В результате теперь есть отдельные главы о сборке и развертывании.

Я надеялся написать книгу примерно того же объема, что и первое издание, и при этом найти способ привести больше деталей. Как видно, мне не удалось достичь своей цели — это издание стало больше! Но я думаю, что смог более четко сформулировать свои идеи.

Навигация по книге

Книгу лучше читать целиком от начала и до конца, но, конечно, можно перейти к конкретным темам, которые вас больше всего интересуют. Если вы все-таки решите углубиться в конкретную главу, возможно, глоссарий в конце книги объяснит вам новые или незнакомые термины. Что касается терминологии, я использую слова «микросервис» и «сервис» взаимозаменяемо на протяжении всего повествования. Считайте, что эти два термина относятся к одному и тому же понятию, если явно не указано иное. Я также резюмирую основные концепции в послесловии, однако учтите, что вы упустите много важных деталей, если просто перейдете в конец книги!

Книга разбита на три отдельные части: «Основы», «Реализация» и «Люди». Давайте рассмотрим, какие вопросы охватывает каждая из них.

Часть I. Основы

В этой части я подробно описываю некоторые ключевые идеи, лежащие в основе микросервисов.

Глава 1 «Что такое микросервисы». Это общее введение в микросервисы, в нем я привожу ряд тем, которые будут подробно описаны позже в книге.

Глава 2 «Как моделировать микросервисы». В этой главе рассматривается важность таких понятий, как скрытие информации, связность и связанность, а также использование предметно-ориентированного проектирования для определения правильных границ ваших микросервисов.

Глава 3 «Разделение монолита на части». Здесь приведены некоторые рекомендации о том, как взять существующее монолитное приложение и разбить его на микросервисы.

Глава 4 «Стили взаимодействия микросервисов». В последней главе этой части мы обсудим различные типы связи микросервисов, включая асинхронные и синхронные вызовы, а также стили взаимодействия «запрос — ответ» и событийную архитектуру.

Часть II. Реализация

Переходя от концепций более высокого уровня к деталям реализации, в этой части мы рассмотрим методы и технологии, которые могут помочь получить максимальную отдачу от микросервисов.

Глава 5 «Реализация коммуникации микросервисов». В этой главе мы подробно рассмотрим конкретные технологии, используемые для реализации взаимодействия между микросервисами.

Глава 6 «Рабочий поток». В ней предлагается сравнение саг и распределенных транзакций и обсуждается их полезность при моделировании бизнес-процессов с использованием нескольких микросервисов.

Глава 7 «Сборка». В этой главе микросервис сопоставляется с репозиториями и сборками.

Глава 8 «Развертывание». В этой главе мы обсудим множество вариантов развертывания микросервиса, в том числе использование контейнеров, Kubernetes и FaaS.

Глава 9 «Тестирование». Здесь обсуждаются проблемы тестирования микросервисов, в том числе проблемы, вызванные сквозными тестами, и то, как могут помочь контракты, ориентированные на потребителя, и продакшен-тестирование.

Глава 10 «От мониторинга к наблюдаемости». В этой главе мы переходим от изучения деятельности по статическому мониторингу к более широкому взгляду на улучшение наблюдаемости микросервисных архитектур. Здесь приводятся некоторые рекомендации относительно инструментария.

Глава 11 «Безопасность». Микросервисные архитектуры создают большую площадь для внешних атак, но также дают нам больше возможностей для глубокой обороны. В этой главе мы рассмотрим правильный баланс между уязвимостью и защитой.

Глава 12 «Отказоустойчивость». В ней предлагается более широкий взгляд на то, что такое отказоустойчивость, и на ту роль, которую микросервисы могут сыграть в повышении отказоустойчивости ваших приложений.

Глава 13 «Масштабирование». В этой главе я описываю четыре оси масштабирования и показываю, как их можно использовать совместно для масштабирования микросервисной архитектуры.

Часть III. Люди

Идеи и технологии ничего не значат без людей и организаций, которые их используют.

Глава 14 «Пользовательские интерфейсы». В этой главе рассматриваются принципы совместной работы микросервисов и пользовательских интерфейсов, начиная с перехода от выделенных команд разработки пользовательского интерфейса (фронтенд-разработки) к использованию BFF и GraphQL.

Глава 15 «Организационные структуры». В предпоследней главе основное внимание уделяется тому, как потоковые команды и команды поддержки могут работать в контексте микросервисных архитектур.

Глава 16 «Эволюционный архитектор». Микросервисные архитектуры не статичны, поэтому вам, возможно, потребуется пересмотреть ваше отношение к системной архитектуре — тема, подробно рассматриваемая в этой главе.

Условные обозначения

В этой книге используются следующие типографические обозначения.

Курсив

Обозначает новые термины и важные понятия.

Моноширинный шрифт

Используется в листингах кода, а также в тексте, обозначая такие программные элементы, как имена переменных и функций, базы данных, типы данных, значения и ключевые слова.

Рубленый шрифт

Применяется для выделения URL, адресов электронной почты.



Обозначает совет или предложение.



Обозначает примечание общего характера.



Обозначает предупреждение или предостережение.

Благодарности

Я постоянно удивляюсь той поддержке, которую получаю от своей семьи, особенно от моей жены Линди Стивенс. Откровенно говоря, без нее этой книги бы не было. Спасибо ей. Я также благодарю своего отца, Джека, Джози, Кейна и весь клан Джилманко Стейнс.

Большая часть этой книги была написана во время глобальной пандемии, которая все еще продолжается, пока я пишу эти строки. Возможно, это мало что значит, но я хочу выразить свою благодарность Национальной службе здравоохранения Великобритании и всем людям в мире, которые обеспечивают нашу безопасность, работая над вакцинами, излечивая больных, доставляя нам еду и помогая тысячами различных способов, которые мне не так очевидны. Эта благодарность также предназначена всем вам.

Второго издания не было бы без первого, поэтому я хотел бы еще раз сказать спасибо всем, кто помогал мне в сложном процессе написания моей первой книги, включая технических рецензентов Бена Кристенсена, Мартина Фаулера, Венката Субраманьяма, Джеймса Льюиса за наши многочисленные поучительные беседы, команду O'Reilly в лице Брайана Макдональда, Рэйчел Монаган, Кристен Браун и Бетси Валишевски, и за отличные отзывы читателей Ананда Кришнасвами, Кента Макнила, Чарльза Хейнса, Криса Форда, Эйди Льюиса, Уилла Темза, Джона Ивса, Рольфа Рассела, Бадринатха Янакирамана, Дэниела Брайанта, Иэна Робинсона, Джима Уэббера, Стюарта Глидоу, Эвана Боттчера, Эрика Суорда и Оливию Леонард. И спасибо Майку Лукидесу, я думаю, за то, что он втянул меня в эту неразбериху в первую очередь!

Для второго издания Мартин Фаулер снова вернулся в качестве научного редактора, и к нему присоединились Дэниел Брайант и Сара Уэллс, которые не пожалели своего времени на отзывы. Я также хотел бы поблагодарить Ники Райтсон и Александра фон Цитцервитца за помощь в доведении технического обзора до конца. Что касается O'Reilly, то весь процесс контролировался моим потрясающим редактором Николь Таше, без которой я бы точно сошел с ума, Мелиссой Даффилд, которая, похоже, справляется с моей рабочей нагрузкой лучше, чем я. Благодарю также Деба Бейкера, Артура Джонсона и остальную производственную команду (мне жаль, что я не знаю всех ваших имен, но спасибо вам!), а также Мэри Трезелер за то, что в трудные времена брала штурвал в свои руки.

Кроме того, огромное спасибо за неоценимую помощь ряду людей, в том числе (в произвольном порядке) Дэйву Кумбсу и команде Туго, Дэйву Хэлси и команде Money Supermarket, Тому Керхову, Эрике Доерненбург, Грэму Тэкли, Кенту Бек, Кевлин Хенни, Лоре Белл, Адриане Муат, Саре Тарапоревалла, Уве Фридрихсу, Лиз Фонг-Джонс, Кейну Стивенсу, Гилманко Стейнсу, Адаму Торнхиллу, Венкату Субраманьяму, Сюзанне Кайзер, Яне Шауманн, Грейди Бучу, Пини Резник, Николь Форсгрэн, Джезу Хамблу, Джин Ким, Мануэлю Паис,

Мэтью Скелтону и команде «Саут Сидней Рэббитоуз». Наконец, я хотел бы поблагодарить восхищенных читателей ранней версии книги, предоставивших бесценные отзывы. Среди них Фелипе де Мораис, Марк Гарднер, Дэвид Лозон, Ассам Зафар, Майкл Блетерман, Никола Мусатти, Элеонора Лестер, Фелипе де Мораис, Натан Димауро, Даниэль Лемке, Сонер Экер, Риппл Шах, Джоэл Лим и Химаншу Пант. И наконец, привет Джейсону Айзексу.

От издательства

Ваши замечания, предложения, вопросы отправляйте по адресу comp@piter.com (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На веб-сайте издательства www.piter.com вы найдете подробную информацию о наших книгах.