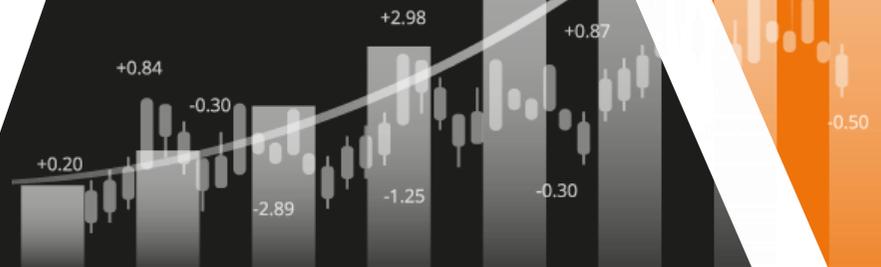




Микросервисная АБС

© ART-Финтех, 2023



Традиционная АБС:

Традиционная АБС:

- монолит = одна база данных = огромные расходы на «железо»
- проприетарные базы данных = лицензионные платежи
- много ручного кода = большие команды = дорогая поддержка

Доработки: долго и дорого

Вывод новых продуктов на рынок: долго и дорого

Сбой АБС = остановка работы банка!



Актуальные требования к АБС:



Короткий
time to market
для новых
продуктов



Возможность
создавать
сложные и
гибкие
продукты



Быстродействие
системы

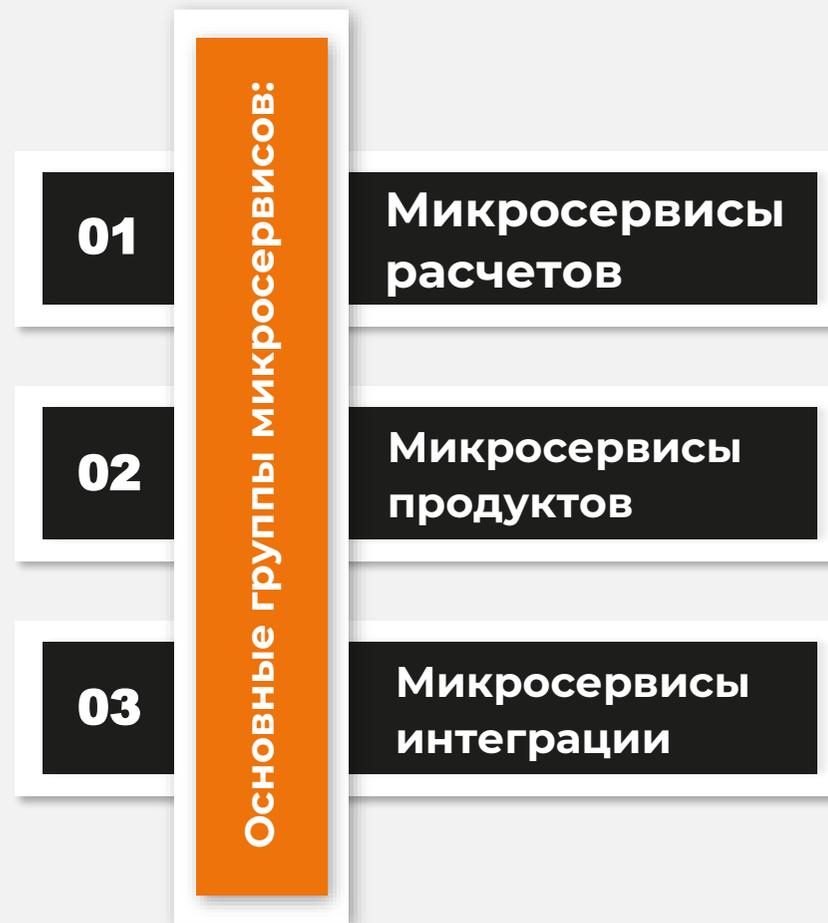


Сокращение
расходов на
команду и
«железо»

Время - деньги!

Микросервисная АБС:

- один продукт - один микросервис
- одна задача - один микросервис



Два подхода к разработке АБС, которые стремятся удовлетворить современные бизнес-требования

1. Low-code разработка, первые поколения

Плюс:

- мало ручного написания кода

Минусы:

- монолитный закрытый «движок» = «черный ящик»
- ограниченный набор возможностей (только то, что «положили в коробку»)
- низкая производительность
- привязка к вендору



Два подхода к разработке АБС, которые стремятся удовлетворить современные бизнес-требования

2. ACG: Automatic Code Generation

Автоматическая генерация кода на основании low-code технологий

Отличия от предыдущих поколений low-code:

- генерация чистого исполняемого кода
- отсутствие «черных ящиков»
- рекордная на рынке производительность
- открытый код = нет привязки к вендору



Структура АСГ:



Генераторы: программисты настраивают шаблоны генерации кода

Конструкторы: аналитики настраивают параметры шаблона

Интерфейсы: бизнес-специалисты настраивают «фронты»

Применение АСГ на практике:

- 90% работы с системой выполняется аналитиками = экономия времени и зарплаты разработчиков
- создание сложного тарифа «с нуля» - за 1-2 дня
- быстродействие: 100 000 транзакций в секунду - и это не предел!
- простое тестирование
- улучшенная безопасность



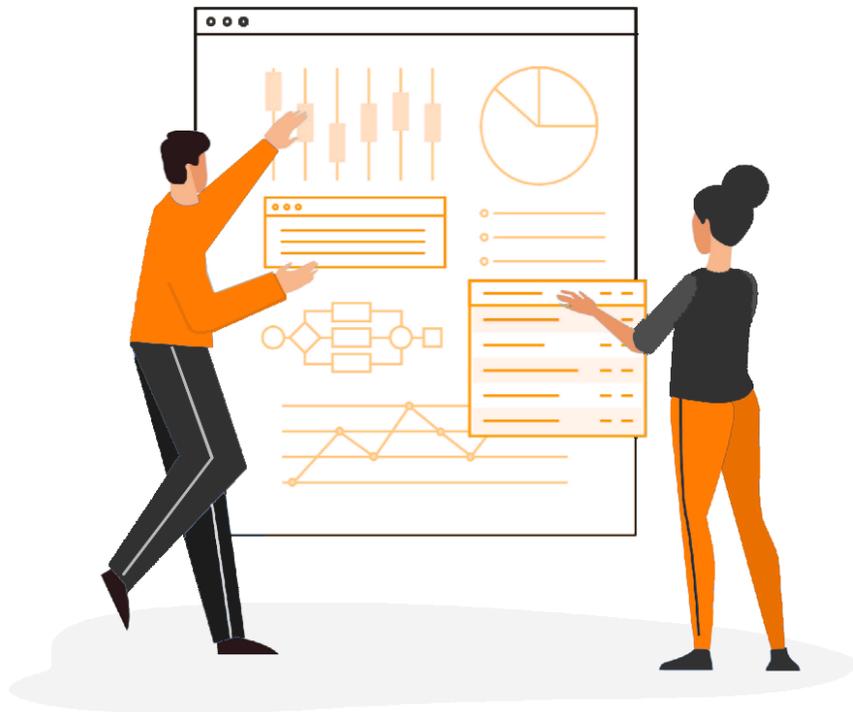
Применение ASG на практике:

Работа Микросервисной АБС невозможна без умной шины нового поколения.

Два способа взаимодействия микросервисов: хореография и оркестровка.

Сейчас на рынке используется только хореография.

Два способа взаимодействия микросервисов:



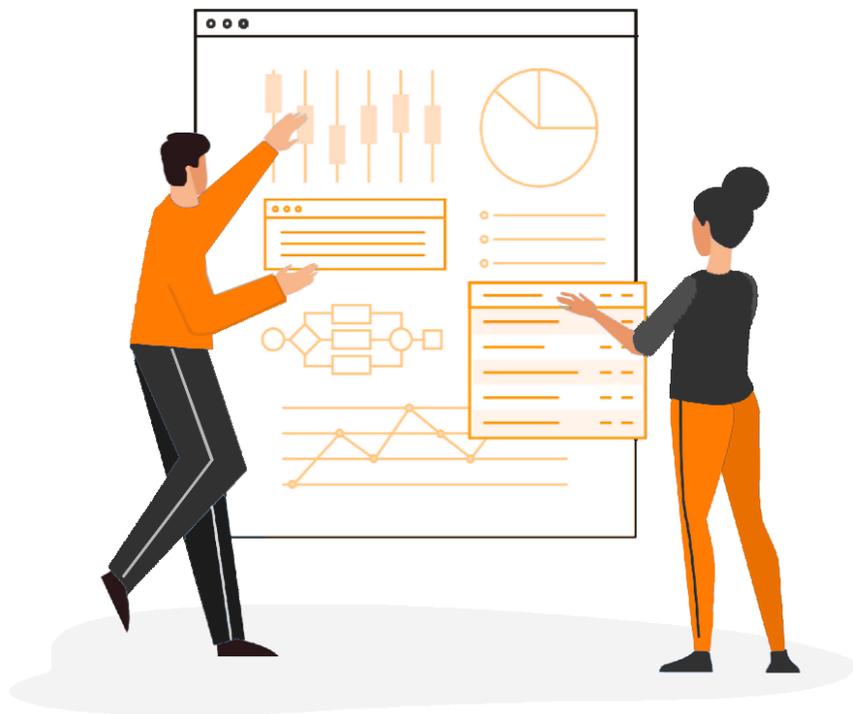
1. Хореография

Правила взаимодействия микросервисов друг с другом описываются в рамках каждого микросервиса.

Проблемы хореографии:

- любые изменения требуют ручную перепрограммировать все микросервисы, задействованные в бизнес-процессе
- длительное тестирование и сложный code review
- пониженная безопасность
- увеличение количества микросервисов = замедление вывода на рынок

Два способа взаимодействия микросервисов:



2. Оркестровка

Создание отдельных интеграционных микросервисов, которые управляют другими микросервисами.

Оркестровщик:

- не требует доработок вручную
- оптимизированное быстродействие
- визуальная нотация бизнес-процессов упрощает разработку и поддержку
- настройка интеграций - силами бизнес-аналитиков

Решает все проблемы хореографии =
верный выбор для микросервисной АБС

Структура Микросервисной АБС

Микросервисы на платформе АСГ:

- SMART ESB (Node.js)
- Тарификатор (Java)
- Конвертации (Java)
- Начисления процентов (Java)
- Формирования проводок (Account engine) (Java)
- Генератор визуальных форм (Angular)
- ETL (Node.js)
- Защищенный файловый обмен (Node.js)
- Система автоматического тестирования (Node.js)
- Конструктор пакетов услуг и продуктов (Java)
- Ряд продуктовых сервисов (Node.js)
- И т.д.

Прочие микросервисы:

- Центр авторизации и ролевая модель (Node.js)
- Генератор отчетов (Node.js)





**Внедрение Микросервисной АБС уже
стартовало в нескольких банках.**

**Начать переход сейчас = стать
технологическим лидером рынка!**



+7 (495) 565 33 43



sales@art-bank.info

www.art-bank.info