



Будущее банковской автоматизации - микросервисная АБС в парадигме coreless

Роман Сургунд
Бизнес-архитектор,
Рексофт
+7 985 923-89-78
surgund@reksoft.ru

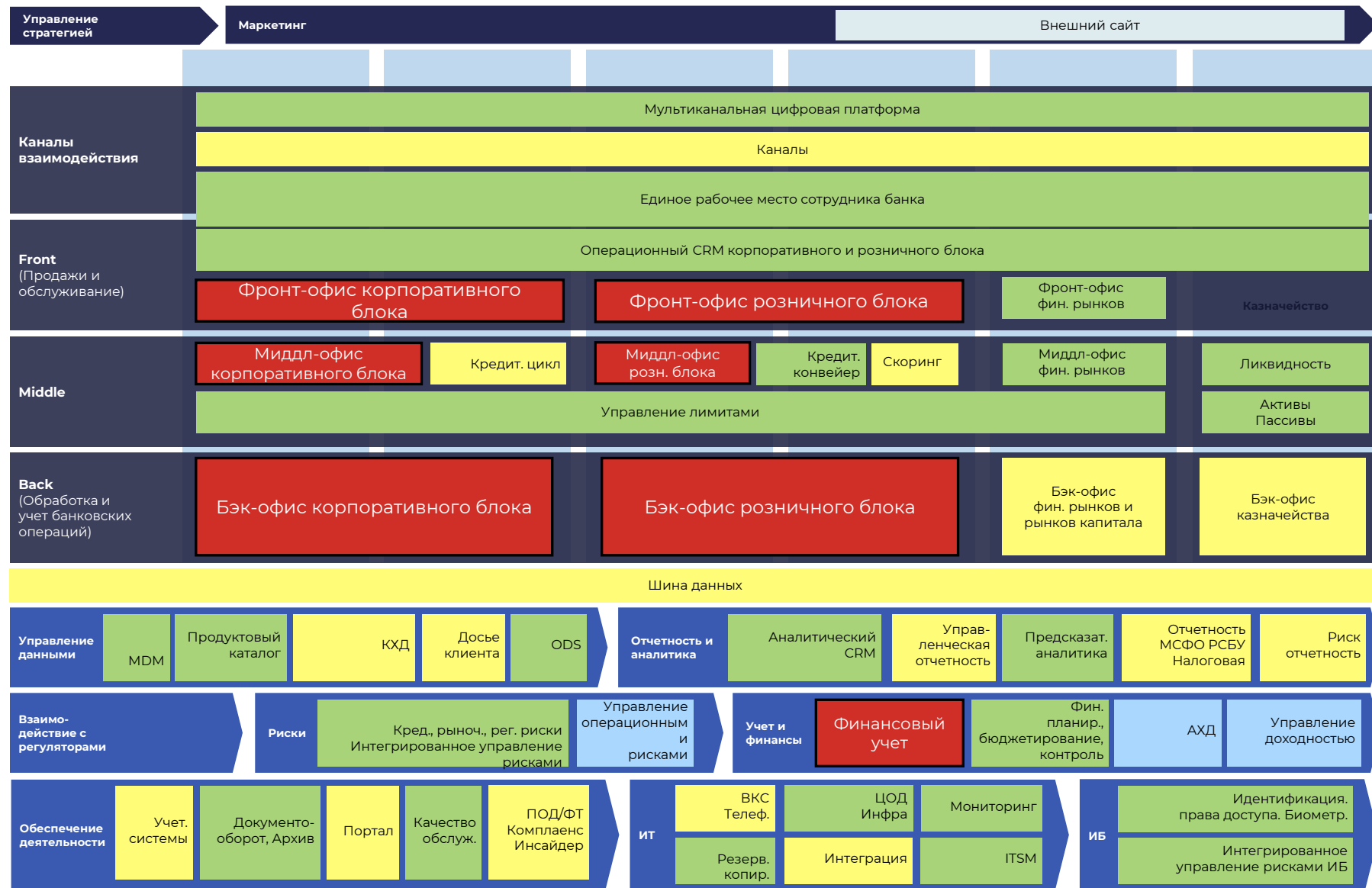
Сегодня АБС традиционно покрывает значительную часть бизнес-архитектуры банка



Традиционная АБС как правило представляет собой монолитную систему. Монолитная система чаще всего основана на проприетарных СУБД. Интеграция банковских систем осуществляется с помощью классической шины данных (ESB).

Технология разработки/доработки таких систем, как правило, заключается в использовании небольшого количества low-code настроек и огромном количестве написания кода вручную

Единая АБС банка





Грани магического кристалла будущего - архитектура ОМС Orchestration - Microservices - Codegeneration



**Микросервисная
архитектура**



**Автоматическая
генерация кода**

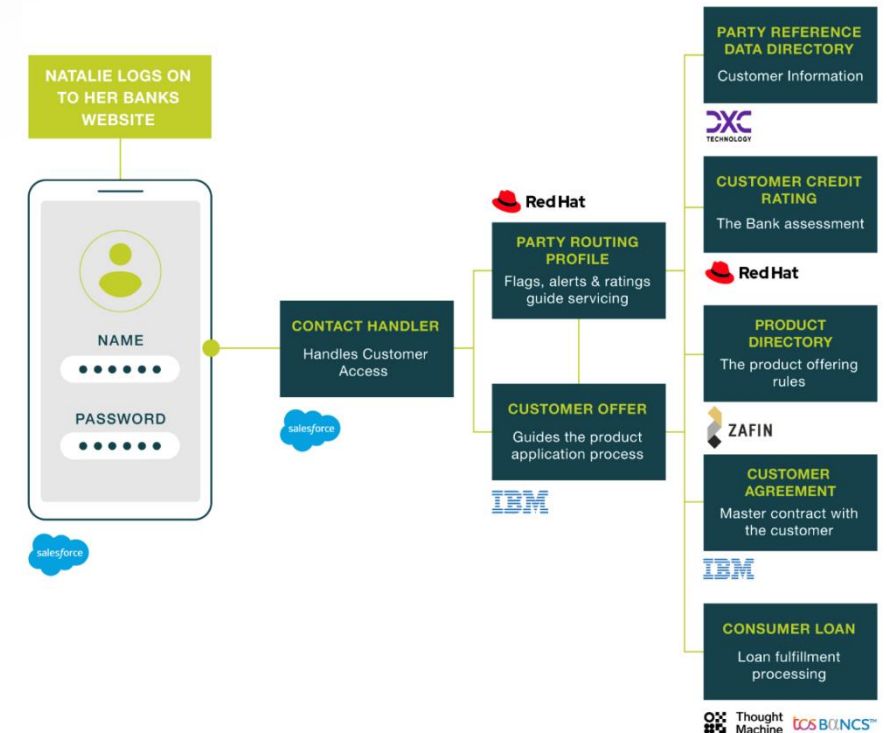
**Оркестровка как
подход к интеграции**

Видение будущего - BIAN coreless banking...



Инициатива, выдвинутая ведущей ассоциацией банков и поставщиков решений, предоставит банкам платформу для сотрудничества с ведущими поставщиками программного обеспечения в разработке перспективной и совместимой банковской инфраструктуры микросервисов на основе архитектуры BIAN.

Coreless banking - это предоставление банковских услуг, не зависящих от традиционных АБС. Это новый способ построения цифрового пути клиента из заранее определенных модульных бизнес-сервисов. В отличие от монолитных приложений и сервисов, где все программные системы тесно интегрированы, в coreless banking нет зависимости от монолитных основных банковских движков. **Каждый из банковских сервисов определяется как отдельная модульная бизнес-функция, которая определяется и поддерживается независимо от всех других функций,** чтобы банки и финтех-партнеры могли продолжать внедрять инновации, а затем предлагать их комбинации клиентам.



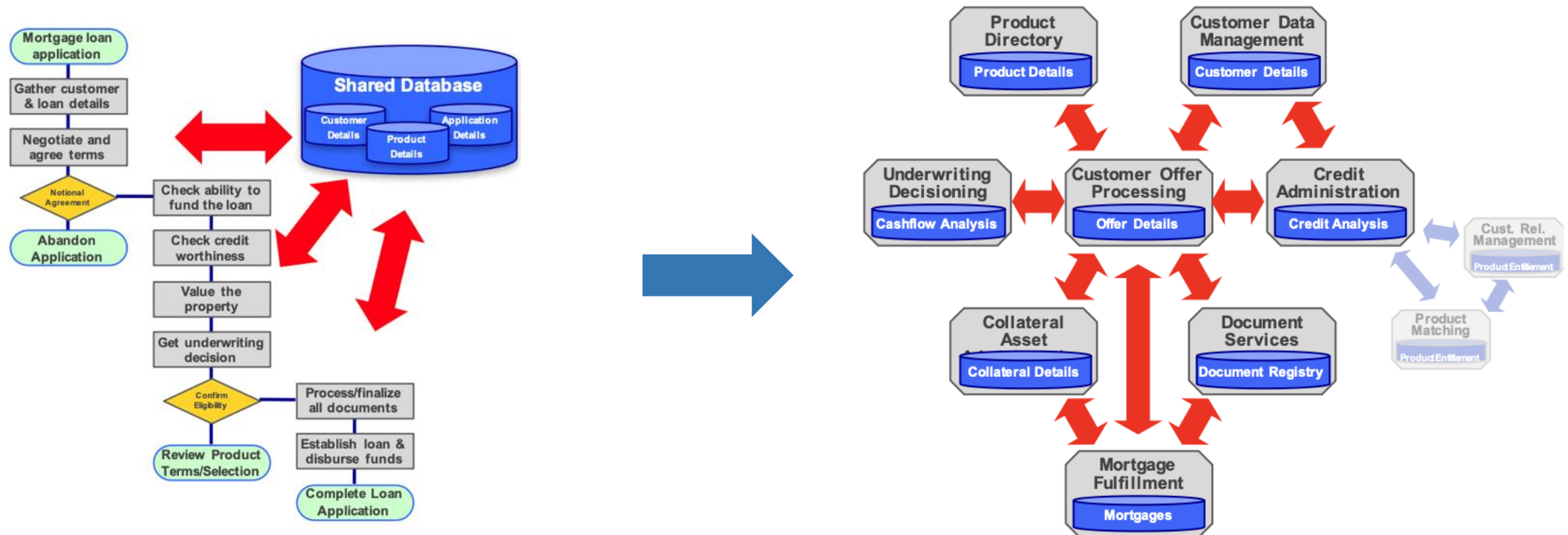
<https://bian.org/deliverables/bian-coreless-banking/>



... через компонентный сервисный подход ...



Функциональные разделы **BIAN Service Domain** определяют высокоинкапсулированные и автономные компоненты, которые хорошо подходят для реализации в высокораспределенных облачных/контейнерных технических средах, которые сегодня все чаще используются с **поддержкой API**. Действительно, без надежного подхода к разделению бизнеса, подобного тому, который используется в BIAN, крайне сложно разрабатывать **высокораспределенные системы любого масштаба**, главным образом потому, что работа с общей бизнес-информацией быстро становится чрезмерно сложной.



... и переход на микросервисную архитектуру

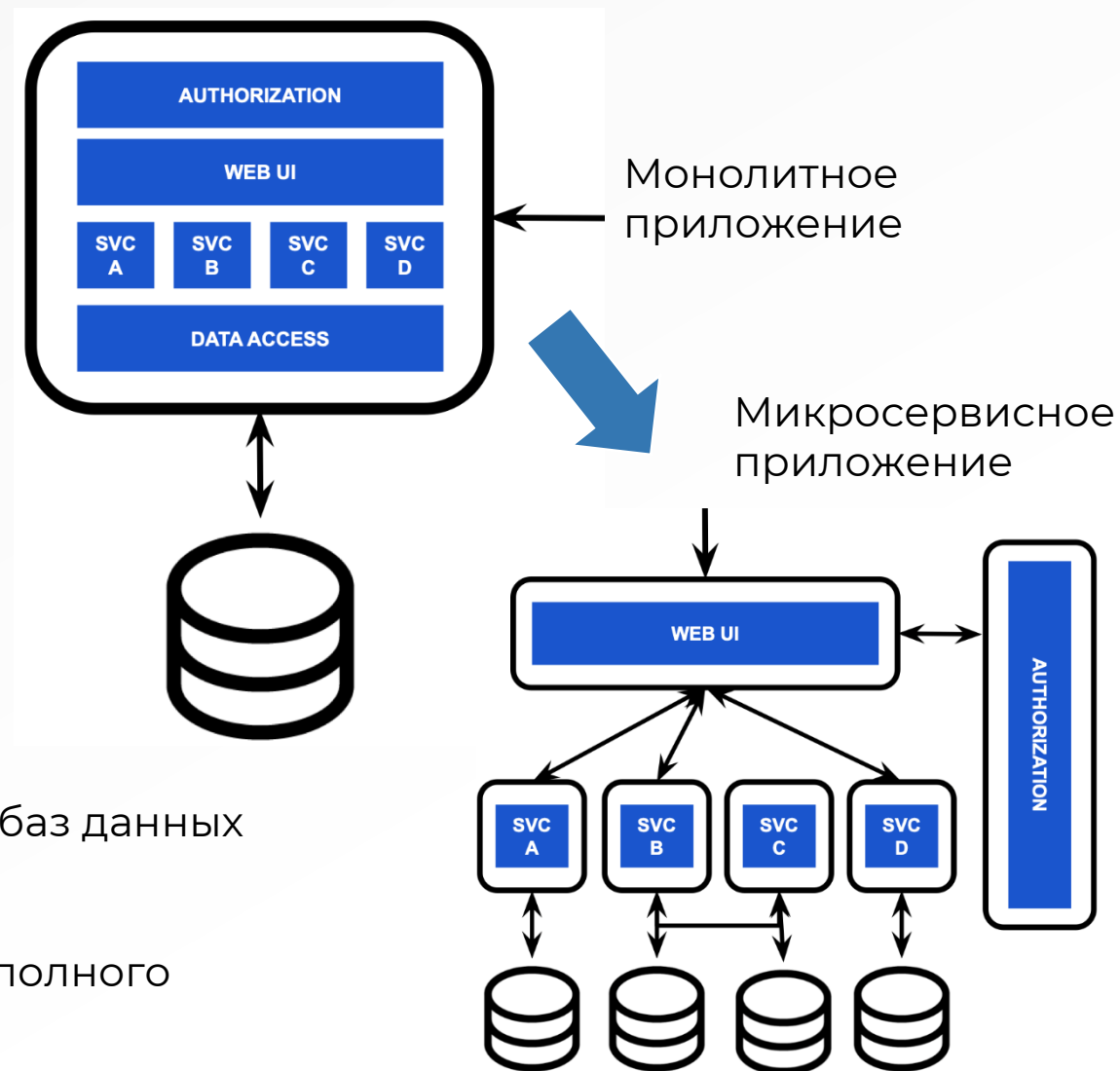
Термин "микросервисная архитектура" появился в последние несколько лет для описания особого способа разработки программных приложений в виде наборов **независимо развертываемых микросервисов**.

Фокус на сокращении времени создания ценности

- Небольшие и простые микросервисы
- Ориентированность на API
- Меньше и быстрее тестировать
- Быстрое время запуска
- **Развертывание независимо** в контейнерах

Что надо учитывать при применении микросервисного подхода:

- Архитектура базы данных с разделами / несколько баз данных
- Сложнее внедрить изменения во многих сервисах
- Сложность интеграции
- **Большое количество** микросервисов для полного функционала



Микросервисная АБС как результат развития архитектуры банковских систем

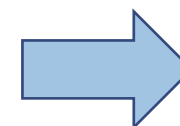
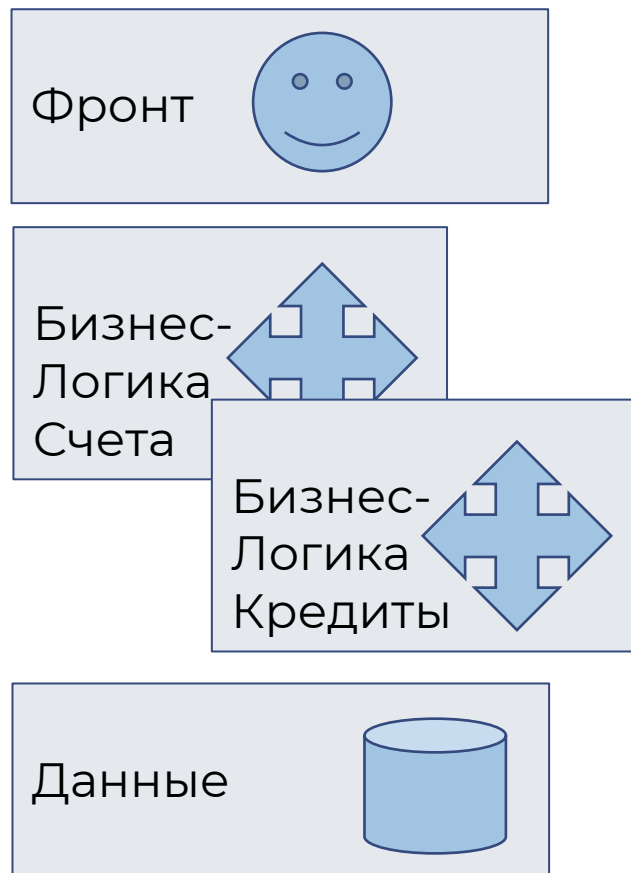
Монолит
Клиент-сервер



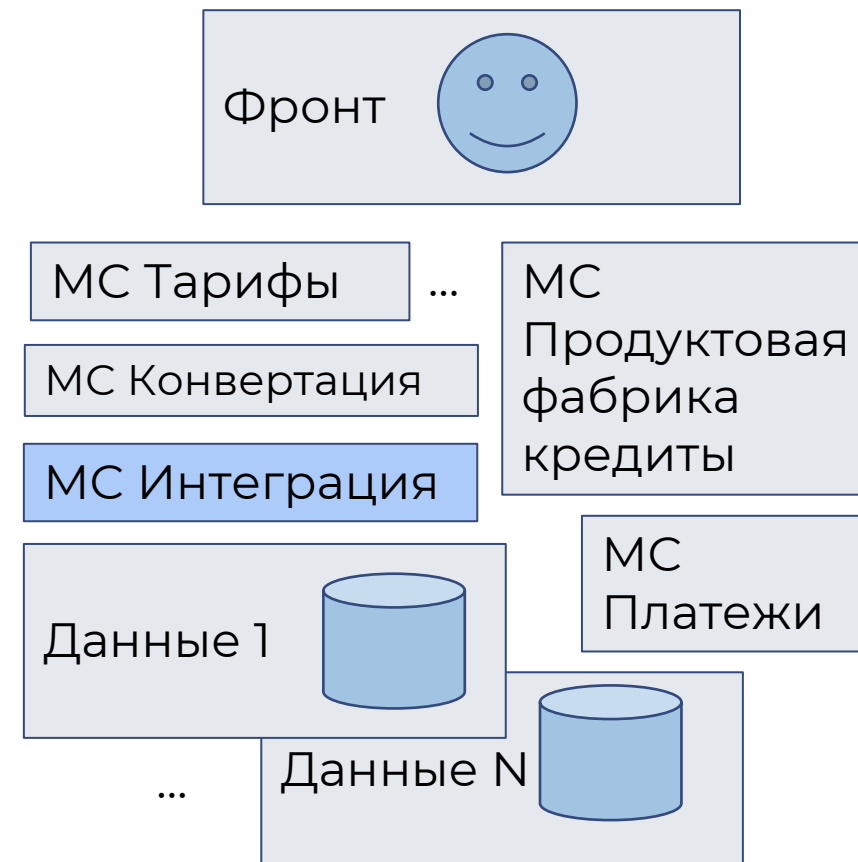
Трех-звенная
архитектура



Модульная
архитектура



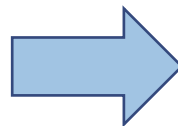
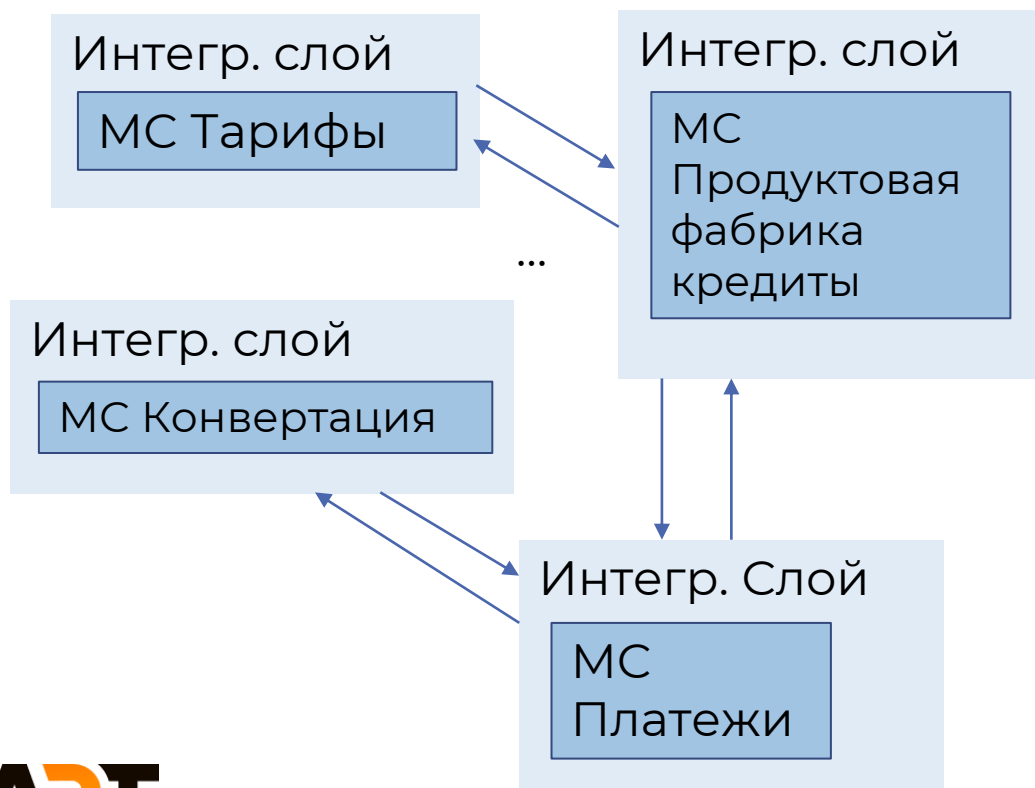
Микросервисная
архитектура



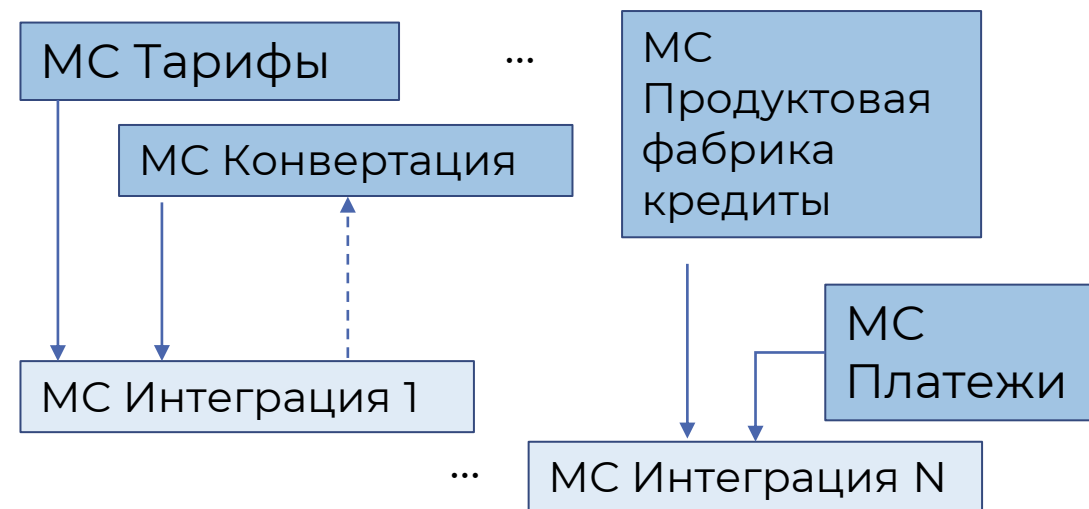
Подходы к интеграции в микросервисной архитектуре

Создание микросервисов – в настоящее время процесс несложный и быстрый. Основные трудозатраты приходится на вопросы интеграции уже готовых микросервисов между собой и с другими системами.

Хореография



Оркестровка



*Перенос интеграционной логики в специальные микросервисы.
Концепция оркестровки становится реальной*

ACG – Современный стандарт разработки кода



Сегодня тренд задает самое эффективное средство программирования – **ACG** (automatic code generation).

В отличие от параметрического low-code подхода, при использовании ACG настраиваются не только параметры исполнения кода, но и **правила генерации программного исполняемого кода** - в соответствии требованиями, соответствующими определенной предметной области (шаблоны).

На выходе - уже результирующий программный код микросервиса, который будет выполнять **бизнес-действия в предметной области**.

Компания получает в полное распоряжение ресурс, эквивалентный **огромному количеству программистов**. Это позволяет IT-подразделениям и специалистам бизнес-областей решать задачи абсолютно нового уровня.

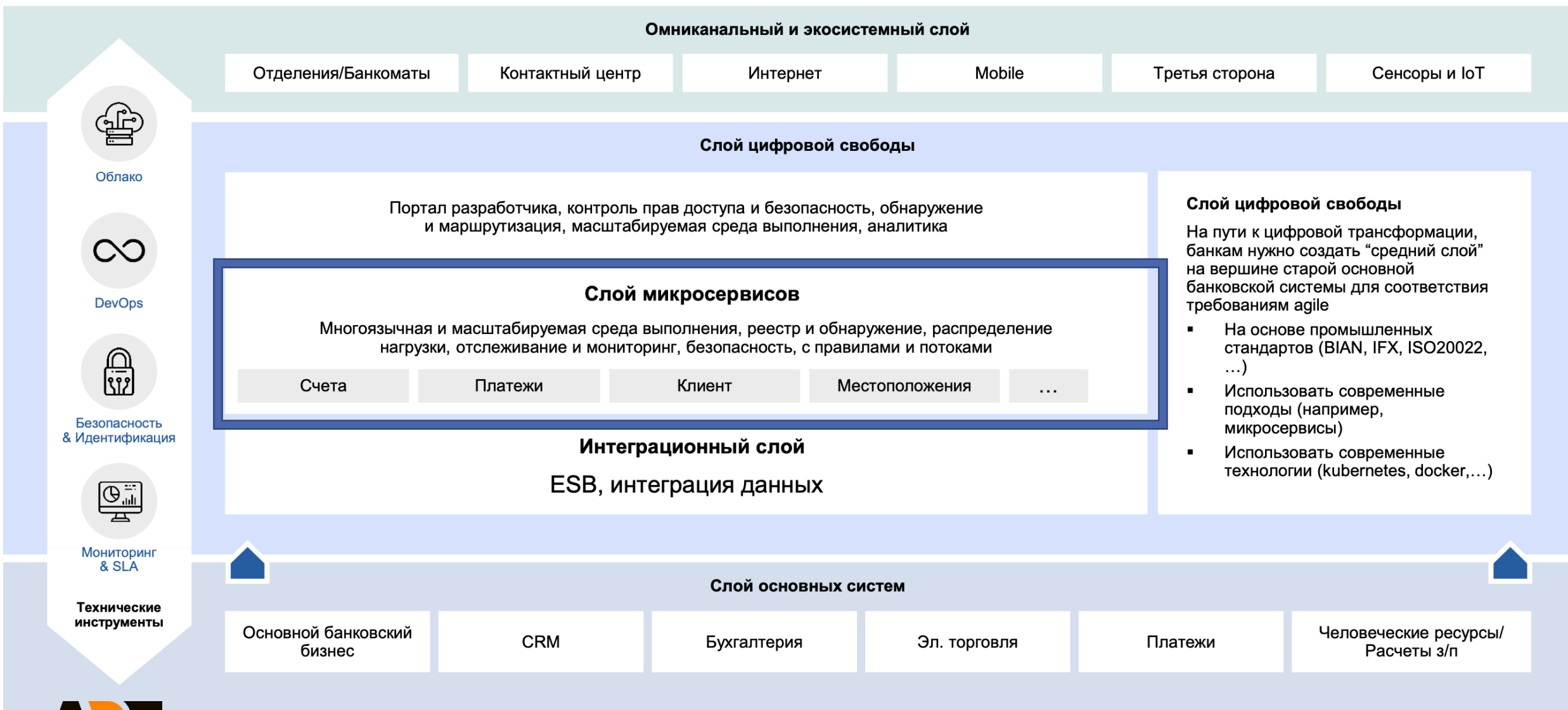


Используемые технологии

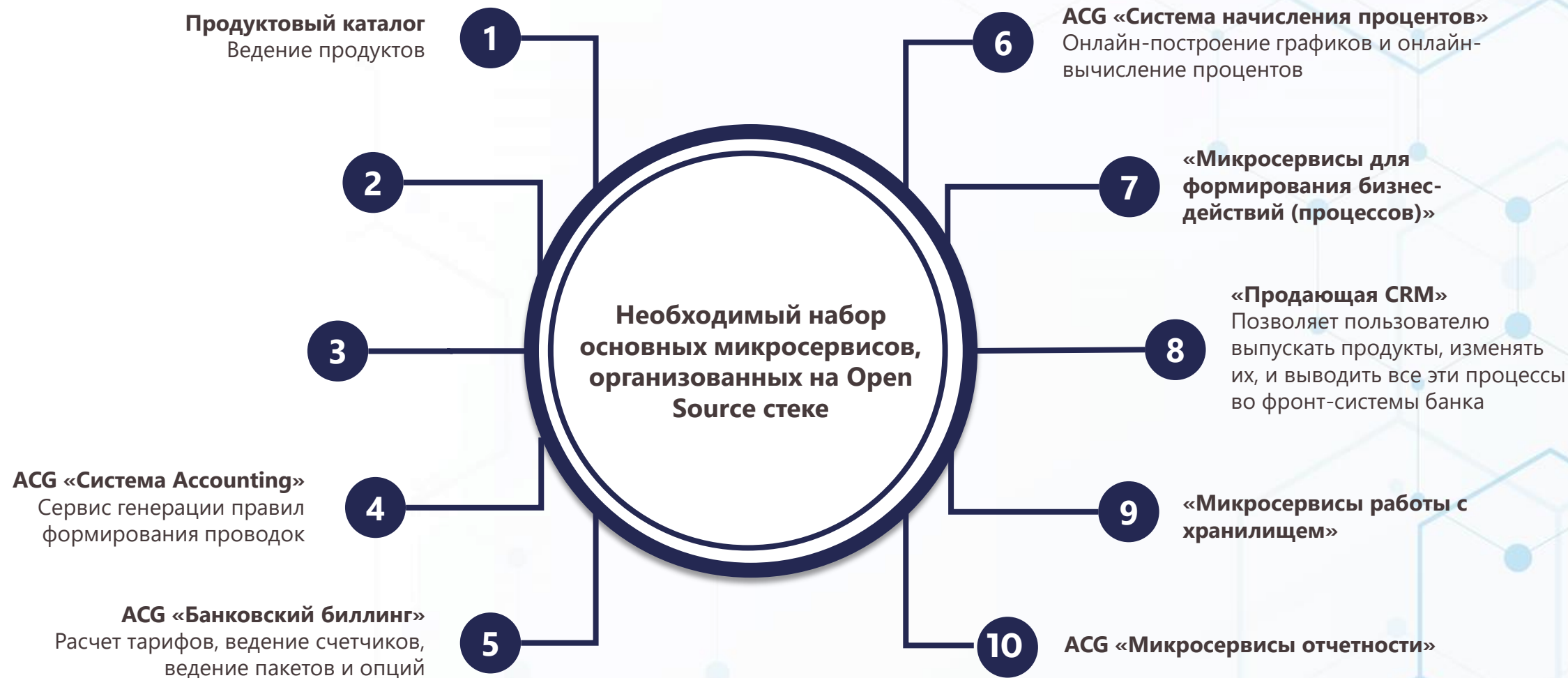
- Генерация нативного кода на нескольких распространенных языках (Сейчас Java, Node.js/Typescript/PL SQL, будут добавляться другие).
- Компиляция в микросервисы на Java или Node.js.
- Запуск в Kubernetes/OpenShift/ Docker Swarm



- Нет интерпретации. Нативный код в рантайме
- НЕ БЫСТРО, А ОЧЕНЬ БЫСТРО – десятки тысяч операций в секунду на простейшей технике
- Стабильно. Работает годами
- Нет риска «черных ящиков». Видно 100% рабочего кода.
- Микросервисы в высшем понимании. Перенос интеграционной логики в специальные микросервисы. Концепция оркестровки становится реальной
- Open Source, Российское ПО
- Работает в больших банках, платежной системе и крупном ритейле. Опробовано, подтверждено



Готовая реализация микросервисной АБС





Благодарим за внимание!

Роман Сургунд
Бизнес-архитектор
Рексофт

+7 985 923-89-78
surgund@reksoft.ru

Владимир Ломов
Советник Генерального
директора,
ГК "АРТ-БАНК"

+7 926 606-36-19
vladimir.lomov@art-bank.ru