

ФИНТЕХ

— РАДАР

05

№5 ОКТЯБРЬ 2023

ТЕХНОЛОГИЯ МЕСЯЦА -

УПРАВЛЕНИЕ API

API – основа для распространения цифровых продуктов среди партнеров и клиентов. Практически каждый программный продукт или облачный сервис сопровождается набором API для управления, интеграции, мониторинга и множества других целей. Это делает API одним из важнейших компонентов современных ИТ-систем.

ТАКЖЕ В ЭТОМ ВЫПУСКЕ

— АФТ-инсайты: международный опыт внедрения Open API
— Использование ИИ во всех отраслях экономик и технологий в программной инженерии

**Целью финтех-радара является
обзор актуальных технологий
и новых трендов финтеха**

По вопросам радара и с обратной
связью, пожалуйста, обращайтесь
к команде исследований и
аналитики Ассоциации ФинТех

research.analytics@fintechru.org





АССОЦИАЦИЯ ФИНТЕХ ИНФОРМАЦИОННЫЕ СЕРВИСЫ

В рамках трека «Информационный сервис» аналитики Ассоциации ФинТех реализуют следующие проекты:

1 «Дайджест зарубежных СМИ» – сбор актуальных новостей и обзор событий в области финансовых технологий на еженедельной основе по 7 основным направлениям:

- Блокчейн и криптоактивы,
- Платежи,
- Open Finance,
- Биометрия,
- Облачные сервисы,
- CBDC,
- Технологии,
- «Эхо финтеха» - резонансные новости по общей финтех-тематике.

2 Финтех-радар – периодический информационный материал, целью которого является информирование целевой аудитории об актуальных событиях в области технологий, зарождающихся трендах для принятия оперативных решений. Материал состоит из инсайтов Управления исследований и аналитики Ассоциации ФинТех, описания технологии месяца, дополнительных материалов по теме выпуска, а также дайджеста ключевых технологических событий.



МАРИАННА ДАНИЛИНА

Руководитель Управления исследований и аналитики



Опыт более 15 лет в финансовом секторе и консалтинге Big4. Профессиональный фокус на корпоративной стратегии, операционной эффективности, IT и цифровой трансформации.

Окончила МГУ имени М.В.Ломоносова и бизнес-школу Colorado Heights University (MBA)

m.danilina@fintechru.org

ДАРЬЯ ПЕТРОВА

Ведущий бизнес-аналитик по исследовательской деятельности



Ведущий бизнес-аналитик по исследовательской деятельности. Специализируется на метавселенных, CBDC, цифровой идентификации и трендах в криптовалютах.

Окончила Northumbria University (бакалавр) и Frankfurt School of Finance & Management (магистратура).

d.petrova@fintechru.org

ГРИГОРИЙ КАРУНАС

Бизнес-аналитик по информационным сервисам



Специализируется на машинном обучении, искусственном интеллекте, безопасной разработке и токенизированных активах.

Окончил МГТУ им. Н.Э. Баумана (бакалавр).

g.karunas@fintechru.org

МЕЖДУНАРОДНЫЙ ОПЫТ ВНЕДРЕНИЯ OPEN API

Аналитическая команда Ассоциации ФинТех проводит исследование **международного опыта внедрения Открытых API** с точки зрения регулирования и бизнес-эффектов.

В ходе исследования эксперты АФТ пришли к выводу о том, что **положительный эффект от внедрения регулирования Открытых API наблюдается только в тех странах, где у финансовых учреждений существует потребность в партнерстве с финтех-сектором или реализации отдельных финтех инициатив.**

 Так, в США **более 12 тысяч финансовых учреждений** пользуются **стандартом FDX**, включающий в себя инструменты для обмена банковскими, налоговыми, страховыми или инвестиционными данными. В американской системе **FDX API** по состоянию на сентябрь 2023 года зарегистрировано **53 млн пользователей** клиентских аккаунтов.

 В Бразилии около **800 участников финансового рынка** обмениваются данными в рамках режима **открытых финансов (Open Finance)**, а количество **запросов API** в среднем за неделю за первое полугодие 2022 года **увеличилось на 171%**.

 В Европейском союзе количество поставщиков данных в рамках Открытого банкинга **увеличилось на 206%** в период с 2019 по 2022 год.

> 40 млн европейских пользователей подключены к банковским услугам через API в 2023 г.

63,8 млн пользователей ожидается в 2024 г. в ЕС

132,2 млн пользователей ожидается в 2024 г. в мире

РОССИЙСКИЙ ОПЫТ: ОТКРЫТЫЕ API VS ПАРТНЕРСКИЕ API

Разница между понятиями «Партнерские API» и «Открытые API»

Наименование по PSD2, OBIE	Open API	External API
Наименование согласно принятой терминологии Банка России	Открытые API	Партнерские API
Стандарт	Отраслевой / кросс-отраслевой на уровне ФОИВа/ ФОИВов	Корпоративный / экосистемный
Среда	Присутствует	Отсутствует
	Требуется аккредитация / сертификация ПО / регистрация	Отсутствует
Юридическая конструкция	Участник - Оператор Среды	Участник - участник
	Единая (типовая)	Вариативная

Наименование по PSD2, OBIE	Open API	External API
Регулирование (обязательность/ требования)	да	нет
Оператор	Присутствует	Отсутствует
Коммерческая модель взаимодействия	На обсуждении	Формируется по каждой интеграции
Мониторинг	Централизованный	Отсутствует / на уровне участника
Система диспутов и споров	Присутствует	Отсутствует
Механизмы согласий	Обязательно, за исключением API с публичной информацией, не содержащих клиентских данных	Опционально

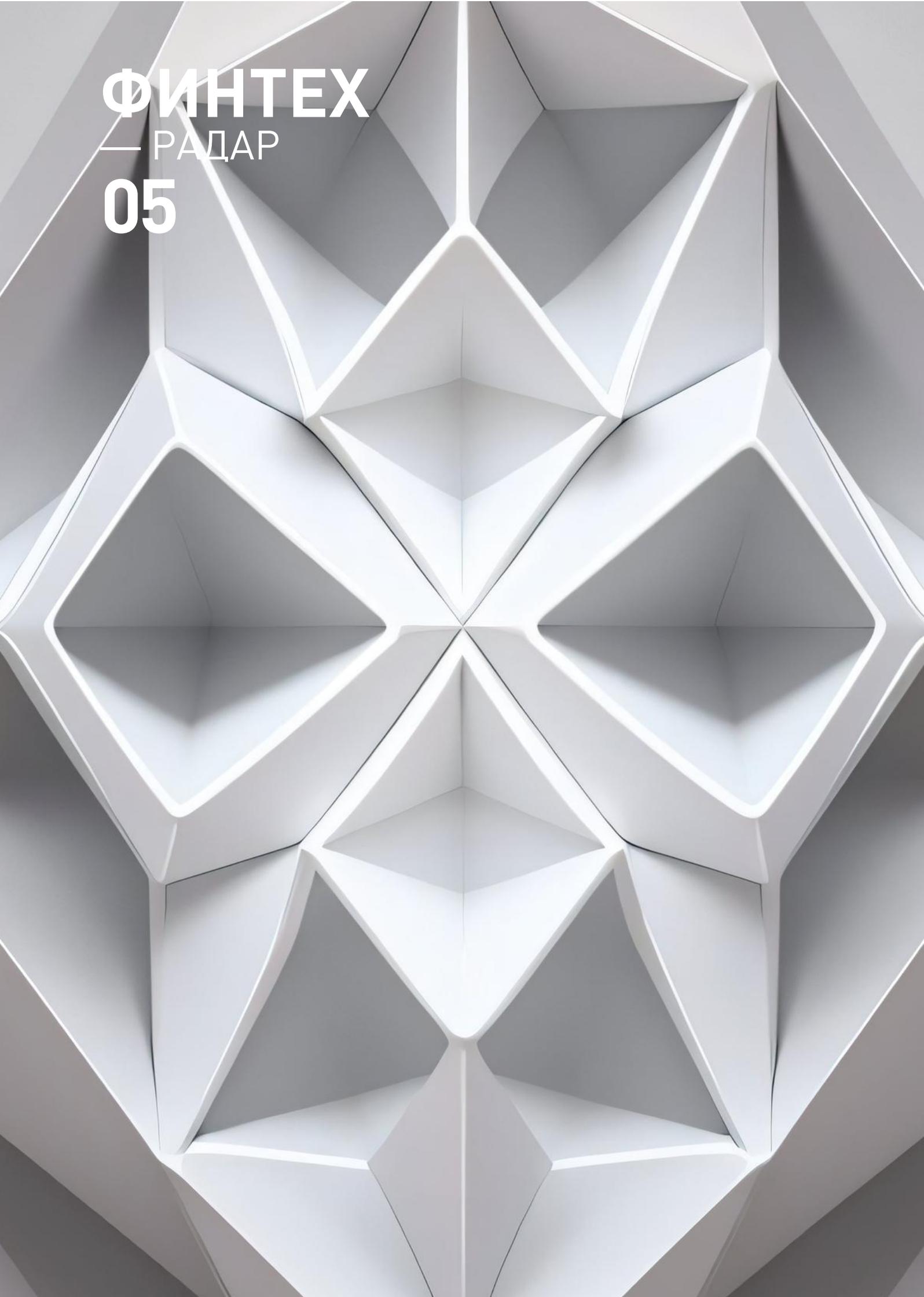
В рамках **исследования уровня технологической готовности и цифровой зрелости компаний финансового рынка**, АФТ опросила участников на предмет использования в их организациях Открытых и Партнерских API.

33% респондентов уже применяют Открытые API

50% планирует их использование после пилотирования

Наибольшая доля партнеров, взаимодействие с которыми происходит через **партнерские API**, наблюдается среди **российских банков и страховых компаний (55%)**, в то время как зарубежные игроки финансового рынка **практически в 2 раза отстают** от показателей РФ. Это объясняется наличием в РФ развитой ИТ-инфраструктуры, позволяющей банкам и страховым компаниям быстро и эффективно интегрировать партнерские API.





ФИНТЕХ

— РАДАР

05

Технология месяца:

Управление API

В настоящее время API являются одним из самых важных компонентов современных ИТ-систем, позволяя разработчикам взаимодействовать с программным обеспечением или сервисом, обеспечивая стандартизированный способ передачи данных между приложениями. Это делает API незаменимым инструментом для интеграции различных приложений и сервисов, что ускоряет процесс разработки и снижает затраты на создание нового ПО.

API – Application Programming Interface – представляет собой программный интерфейс приложения, описывающий, каким образом программы могут передавать друг другу данные и функции. API может использоваться как составной блок для разработки новых взаимодействий между людьми, другими приложениями или умными устройствами.

В ПОСЛЕДНИЕ ГОДЫ РАСПРОСТРАНЕНИЕ API ЗНАЧИТЕЛЬНО УВЕЛИЧИЛОСЬ



Исследование Cloudflare показало, что в конце 2021 года вызовы API составляли **более 50% мирового интернет-трафика**, причем их количество выросло сильнее относительно остальных видов. Тогда, когда общий рост количества API-вызовов составил 21%, в **банковской сфере** их рост составил **70%**



Количество API-запросов на платформе Postman API Platform в 2022 году увеличилось на **33%** по сравнению с 2020 годом



По прогнозам Grand View Research, к 2030 году рынок API маркетплейсов составит **\$49,5 млрд** при среднем темпе роста 17,8% в год



В настоящее время на GitHub существует **более 3 млн** репозиторий, связанных с API

API – **основа для распространения цифровых продуктов** среди партнеров и клиентов. Практически каждый программный продукт или облачный сервис сегодня сопровождается набором API для управления, интеграции, мониторинга и множества других целей.

Быстрое распространение современных архитектур API также совпало с экспоненциальным ростом облачных вычислений и распространением мобильных устройств, где программные интерфейсы стали идеальным средством интеграции разнородных систем и облегчения обмена данными.

Фактически любое действие, выполняемое посетителем сайта, пользователем мобильного приложения или даже смарт-устройства, подразумевает **как минимум один вызов API** к удаленной бэкенд системе.

API не просто «ИТ-инструмент» – они оказывают **сильное влияние на операционную эффективность, масштабируемость и гибкость** практически любого бизнеса и напрямую влияют на его прибыльность.

Руководители компаний, занимающихся разработкой ПО, стремительно внедряют API, но при этом сталкиваются с растущей проблемой безопасного управления ими. По прогнозам Gartner, к 2025 году менее 50% корпоративных API будут управляемы, поскольку «взрывной» рост числа API превышает возможности современных инструментов управления API.

ИСТОРИЯ API

Термин API, а также общая концепция проектирования приложений как слабосвязанных компонентов, взаимодействующих через абстрактные интерфейсы, появились в 1960-х годах. Это было связано с растущей сложностью ПО и возникшей необходимостью скрыть эту сложность от разработчиков, позволив им сосредоточиться на бизнес-логике приложений и повторно использовать программный код.

Раньше API использовались исключительно разработчиками и были известны только в технической среде, тогда как сейчас они превратились в одну из основ современного цифрового бизнеса.

В начале 2000-х г. появился архитектурный стиль для разработки веб-сервисов **Representational State Transfer** (REST). Подход представлял собой быструю, масштабируемую и экономичную альтернативу более ранним корпоративным стандартам, таким как SOAP (Simple Object Access Protocol — протокол доступа к объектам). Теперь появилась возможность быстро создавать и публиковать собственные API, используя существующие стандартные протоколы, форматы (HTTP, XML и JSON) и существующие инструменты.

КАК РАБОТАЮТ API

Как упоминалось ранее, API – это программный интерфейс для приложений. Приложение может быть любым программным обеспечением, выполняющим определенную функцию. Интерфейс приложения можно рассматривать как контракт между двумя программами, определяющий правила взаимодействия между ними, используя запросы и ответы.



Потребитель API – тот, кто вызывает API, а также получает либо использует данные или услуги, предоставляемые сервисом.



Поставщик API — тот, кто создает, управляет, обслуживает и поддерживает API, позволяющий разработчикам обращаться к отдельному сервису и интегрироваться с ним.

Чтобы API мог получить нужную информацию от исходного приложения или системы, он должен обратиться к конечной точке.

Конечная точка API — это адрес, по которому API получает запросы о конкретном ресурсе на своем сервере. В API конечная точка обычно представляет собой унифицированный локатор ресурсов (URL), который указывает местоположение ресурса на сервере.

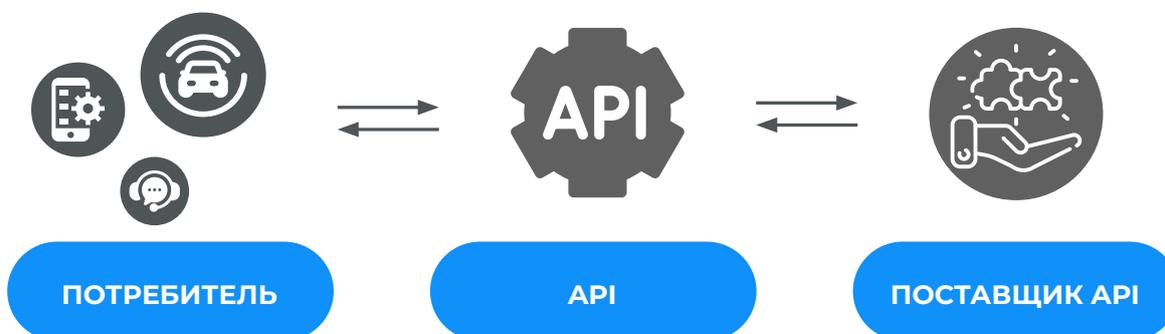


Схема 1. Процесс взаимодействия API

ЗАЧЕМ API БИЗНЕСУ

Согласно исследованию Postman “State of API Report 2023”, респонденты основными положительными эффектами от перехода к подходу API-First выделяют **ускорение интеграции с партнерами, повышение уровня удовлетворенности разработчиков и качества выпускаемого ПО, а также выпуск новых продуктов.**

Эффекты от внедрения API-FIRST подхода



Источник: Postman – State of API Report 2023

Внешние API необходимы бизнесу по разным причинам. Они помогают в системной интеграции, позволяя внедрять в приложения сторонние сервисы и тем самым **расширять функциональность и взаимодействие** с пользователями. Кроме того, они **способствуют масштабируемости**, позволяя добавлять новые функциональные возможности по мере роста компании без существенной перестройки системы.

Данные виды API также обеспечивают экономическую эффективность за счет снижения затрат на разработку всего функционала «с нуля», ускоряя процесс разработки и позволяя быстрее внедрять новые функции или услуги.

Внешние API также **обеспечивают доступ к специализированным знаниям и опыту**, которые могут превышать возможности внутренней команды предприятия. Они повышают удобство работы пользователей за счет интеграции различных сервисов и предоставляют доступ к современным средствам анализа данных.

Преимущества использования внешних API для организаций

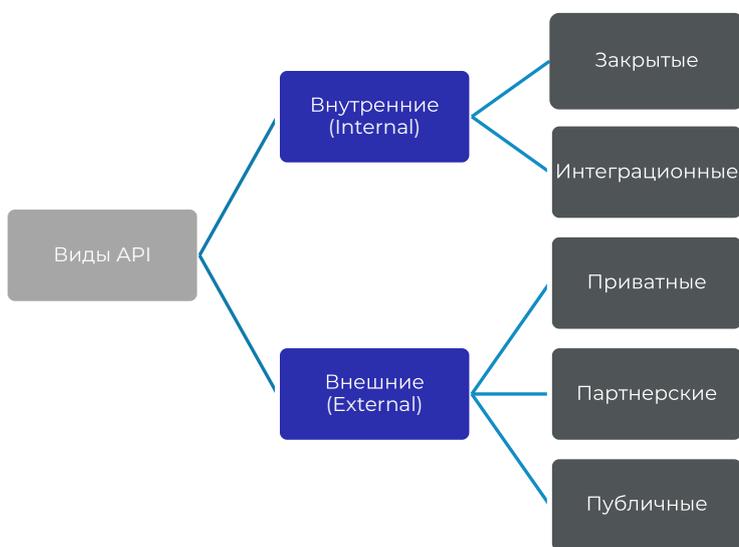
- **Расширение каналов получения доходов.** Компании могут монетизировать свои API или использовать их для привлечения клиентов к своим продуктам и услугам.
- **Снижение сложности разработки.** API могут инкапсулировать сложные процессы, облегчая разработчикам выполнение задач без необходимости вникать в их суть.
- **Гибкость проектирования ПО.** Разработчики могут выбирать, какие API использовать, что позволяет им создавать и настраивать приложения в соответствии с конкретными потребностями.
- **Улучшение процесса принятия решений.** Благодаря API, обеспечивающим доступ к данным в режиме реального времени, компании могут быстро принимать более обоснованные решения.

Преимущества использования внутренних API для организаций

- **Повышение эффективности и снижение затрат.** Применять API при внутренней разработке приложений и сервисов оказывается эффективнее и дешевле. Они позволяют разработчикам быстро извлекать функциональные возможности и данные из существующих приложений для создания новых.
- **Эффективное реагирование на запросы рынка.** Потребности клиентов быстро меняются, и планирование и разработка новых продуктов и сервисов становятся проблематичными. Интеграция API в ПО сокращает срок от разработки до вывода продукта на рынок (Time to Market, T2M), позволяя организациям быстро адаптироваться к новым приложениям для социальных сетей, веб-приложениям и аппаратному обеспечению.
- **Устраняет внутреннее соперничество и способствует развитию взаимодействия между командами разработчиков.** Повсеместное внедрение API отчасти вызвано необходимостью подтолкнуть разработчиков совместно работать на благо компании. Подход, ориентированный на API, поощряет культуру сотрудничества и обмена опытом не только среди команд разработчиков, но и среди тех, кто ими управляет.

ВИДЫ API

API могут использоваться как для взаимодействия внутри автоматизированных систем организаций, так и для подключения к партнерам и сторонним сервисам. Многие компании не нуждаются в интеграции со внешним миром, поэтому API часто используются исключительно для взаимодействия внутренних систем.



Какие виды API внедрены в вашей организации?



Источник: Google Cloud, The Digital Crunch Time: 2022 State of APIs and Applications





Внутренние API

Внутренние API – API, используемые для интеграции систем и данных внутри компании. Они в свою очередь включают в себя закрытые и интеграционные API.

Закрытые API

Закрытые API – предназначены в первую очередь для использования модулями **внутри автоматизированных систем**, а не для интеграции на уровне организации. Обычно управляются отдельными командами, а не централизованно.

Интеграционные API

Интеграционные API – предназначены для **взаимодействия** систем организации **между собой**. Включены во внутренний портал разработчиков, обязаны соответствовать стандартам организации.



Внешние API

Внешние API – API, основное предназначение которых – взаимодействие с внешними партнерами и сервисами.

Приватные API

Приватные API – **выходят из внутреннего контура** организации, но **не взаимодействуют со сторонними организациями**. Например, устройства Интернета вещей (IoT).

Партнерские API

Партнерские API – доступны **только надежным партнерам** и скрыты от сторонних разработчиков.

Публичные API

Публичные API – находятся в открытом доступе в Интернете, доступны всем сторонним разработчикам.



Схема 2. Взаимодействие различных типов API

SOAP API

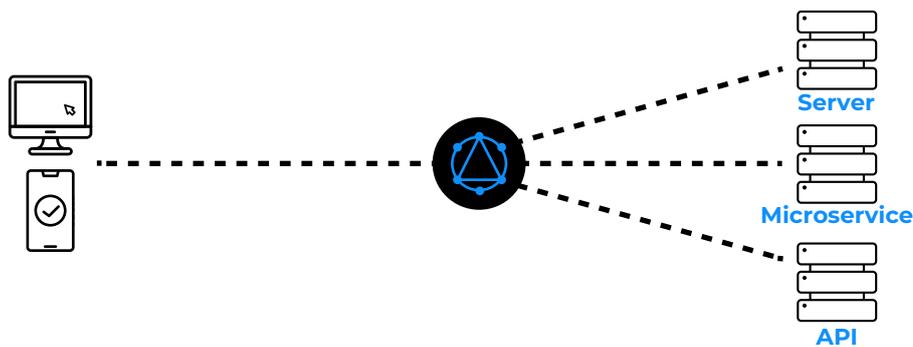
API на основе протокола SOAP (Simple Object Access Protocol – простой протокол доступа к объектам) оформляются как XML-файлы и распространены среди протоколов веб-коммуникаций. Их сложно использовать, но они обеспечивают высокий уровень безопасности и отличаются большей точностью обмена сообщениями.



Основным недостатком SOAP является то, что этот формат **излишне сложен для освоения** и отстает от новых, более гибких методов взаимодействия, дающих более устойчивые результаты.

GraphQL

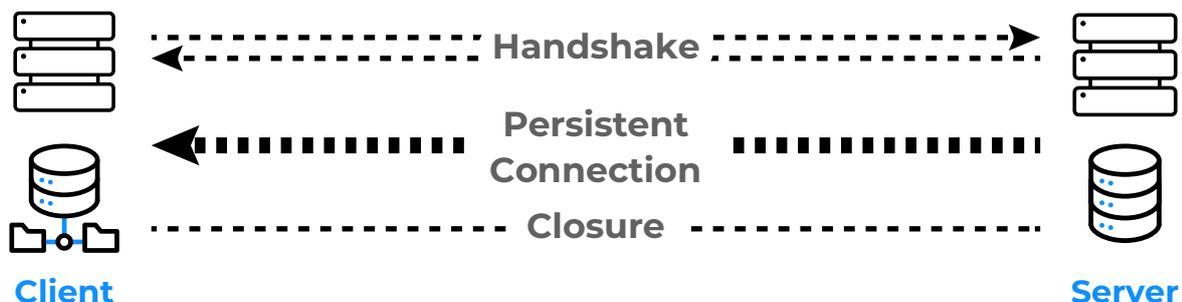
GraphQL — это язык запросов для API, позволяющий клиентам получать нужные данные без составления сложных запросов, взаимодействуя с одной конечной точкой. Такой подход позволяет сократить количество запросов между клиентом и сервером, что **повышает производительность**.



GraphQL является внутренним протоколом Facebook (запрещена на территории РФ), выпущенным в 2015 году и быстро ставшим основным типом API для JSON. В рамках этого типа API для начала необходимо создать схему, описывающую все возможные запросы и конкретные типы возвращаемых данных. Эта задача не простая, но затем API сможет принимать конкретные запросы и возвращать результат, который в точности соответствует тому, что ищет пользователь.

WebSockets

WebSockets (веб-сокеты) – это протокол двусторонней связи, обеспечивающий передачу данных между клиентом и сервером в режиме реального времени.



К плюсам веб-сокетов можно отнести связь в реальном времени и низкую задержку. Среди минусов отмечается необходимость постоянного соединения, что может привести к увеличению нагрузки на сервер, а также потенциальную уязвимость в системе безопасности.

Веб-сокеты часто используются в финансовом секторе и их распространение также растет и в других сферах.

СПЕЦИФИКАЦИИ API

Спецификации API – это сочетание протоколов, шаблонов и стилей в машиночитаемых и человекочитаемых форматах, которые могут использоваться для описания API-протоколов между производителем и потребителем API.

OpenAPI

OpenAPI — это спецификация, помогающая описывать и документировать API. Его сильные стороны включают **простоту использования, возможность точно определить формат данных, поддержку версионирования** и различные форматы сообщений.

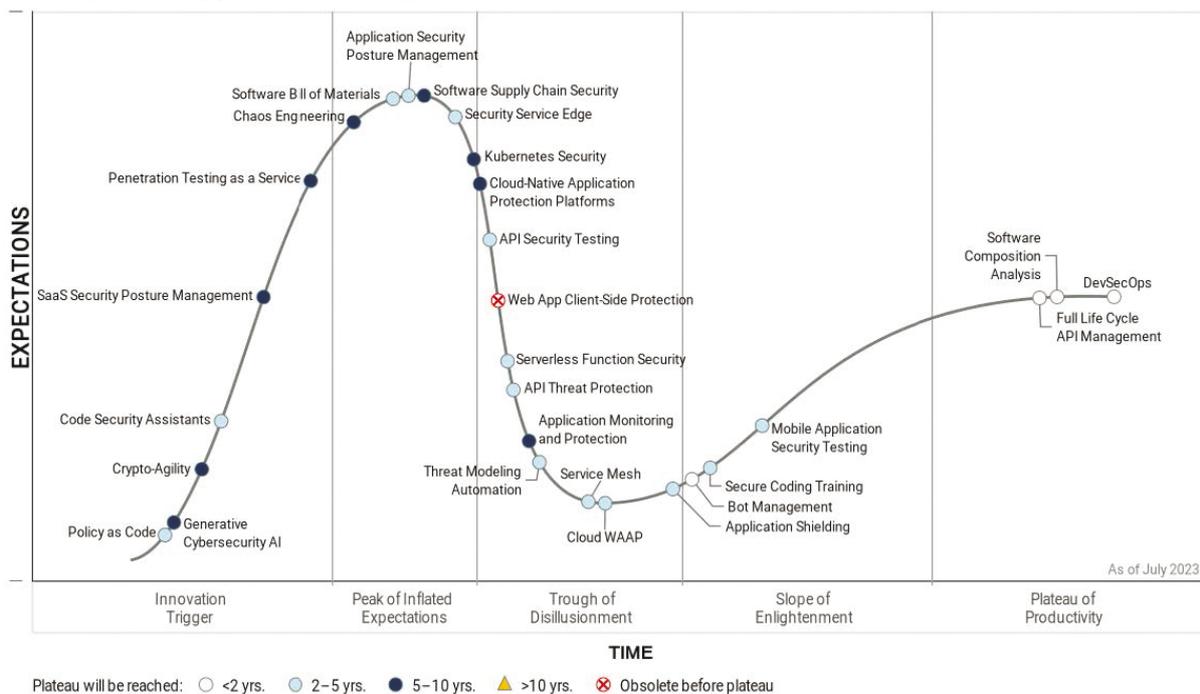
Хотя OpenAPI тесно связан с REST, его применение не ограничено только этим стилем. Он также может использоваться с другими архитектурными стилями, такими как GraphQL, gRPC или SOAP. Так, около 30% API в реестре Сбера написаны с использованием спецификации OpenAPI.

AsyncAPI

AsyncAPI — это спецификация, сосредоточенная на **описании асинхронных API**. В отличие от традиционных (синхронных) API, работающих по принципу «запрос/ответ», асинхронные API позволяют использовать **потокные данные, событийно-управляемую архитектуру и системы обмена сообщениями**.

AsyncAPI базируется на OpenAPI, используя те же концепции и структуры. Эта спецификация расширяет возможности OpenAPI, позволяя описывать особенности асинхронных API. Фактически, обе спецификации имеют одинаковую структуру и синтаксис, что облегчает их сочетание и комбинирование в проектах.

КЛЮЧЕВЫЕ ТРЕНДЫ API



GARTNER 2023 HYPE CYCLE FOR API

Событийные API

По мнению аналитического агентства Gartner, одним из главных трендов API являются **событийные API**. Событийные API (Event-driven API) описывают интерфейсы, используемые в событийно-управляемой архитектуре, включая подписки и создание каналов. Такие API отличаются от традиционных REST-интерфейсов типа «запрос/ответ» тем, что позволяют в реальном времени получать уведомления об изменениях в данных и состоянии систем. Наиболее популярным стандартом является AsyncAPI.

Наблюдаемость API

Наблюдаемость API (API Observability) представляет собой новый тренд в Нуре Cycle для API, отражающий важность анализа зависимостей и мониторинга использования API. Решения для наблюдения за API позволяют разработчикам и владельцам API видеть влияние изменений в API, включая обнаружение разрушительных изменений до того, как они произойдут. Мониторинг критически важных API является неотъемлемой частью бизнес-операций.

Начиная с обнаружения изменений и заканчивая мониторингом рабочих состояний, в процессе наблюдаемости собираются данные на каждом этапе. Наблюдаемость API прошла путь от простого анализа журналов до сложных приборных панелей, агрегирующих телеметрию по многим API, а также интеграции с репозиториями кода и интегрированными средами разработки (IDE). Системы и практики наблюдаемости API затрагивают разработчиков и команды разработчиков API-платформ.

Федеративные API-шлюзы

Кроме того, набирают популярность **федеративные API-шлюзы**. Поскольку использование API продолжает расти, организациям приходится управлять несколькими инициативами в области API, вовлекая в них облачные среды и различные команды. В связи с этим возникает потребность в решениях для управления API, охватывающих несколько API-шлюзов.

Отраслевые API

Отраслевые API также набирают обороты. В этом году в Нуре Cycle учитываются API телекомов, финансовые API, открытые банковские API, а также использование API в цепочках поставок. Открытый банкинг опережает другие отрасли благодаря наличию открытых стандартов.

Безопасность API

Безопасность API остается критически важным процессом. Повышается степень зрелости тестирования безопасности API, контроля доступа к API и защиты API от угроз. Это отражает постоянную потребность в безопасности, а также зрелость инструментов и практик обеспечения безопасности API.

Управлением полным жизненным циклом API

Управление полным жизненным циклом API является развитым технологическим подходом. В отчете об актуальных трендах и этапах развития технологий в обеспечении безопасности приложений, эксперты оценили уровень развития Full Life Cycle API Management как **«зрелый мейнстрим»**.

УПРАВЛЕНИЕ API

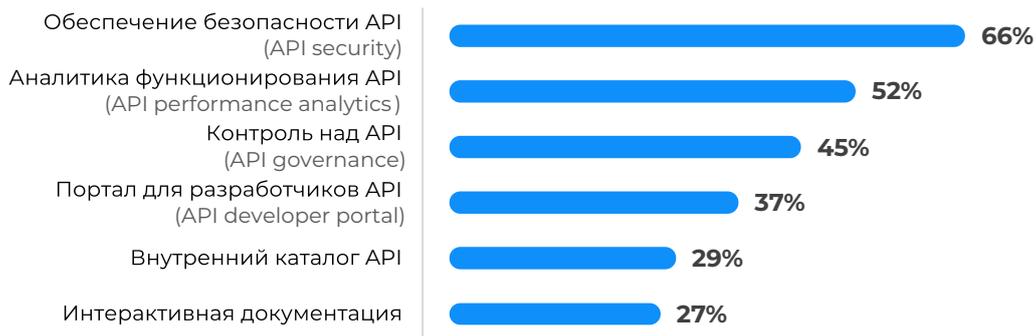
По [прогнозам](#) MarketsandMarkets, в 2022 году объем рынка управления API составил **\$4,5 млрд**, а к 2027 году составит **\$13,7 млрд** при среднем темпе роста 25,1% в год.

Управление API в организации – **комплексная задача**, в рамках которой проводятся такие мероприятия, как обеспечение безопасности, обнаружение сервисов, маршрутизация, оркестровка и аналитика функционирования, менеджмент портала разработчика, а также управление жизненным циклом каждого отдельного API.

Исследование, проведенное Google Cloud, показало, что организации наиболее важными компонентами инфраструктуры управления API считают **обеспечение безопасности (66%)** и **аналитику функционирования API (52%)**.



Какие компоненты API-инфраструктуры вы считаете наиболее важными?



Источник: Postman – State of API Report 2023

Средства управления API помогают создавать экосистемы API, публиковать и эксплуатировать API, а также собирать аналитику для мониторинга и отчетности о бизнес-ценностях. API широко используются в качестве **основного способа соединения систем, приложений и устройств; интеграции бизнес-партнеров, построения модульных и композитных архитектур программного обеспечения.**

Также растет использование API в **качестве цифровых продуктов**, прямо или косвенно монетизируемых. Инициативы по цифровой трансформации ускорили потребность в создании, управлении, эксплуатации и обеспечения безопасности API. Кроме того, они сделали управление полным жизненным циклом API важнейшей основополагающей способностью каждой организации.

Использование API для внутреннего и внешнего обмена данными приводит к необходимости управления интерфейсами с использованием **полного жизненного цикла API.**

API, объединяющие данные, сервисы и аналитические материалы, все чаще рассматриваются как **монетизируемые продукты**, позволяющие реализовать платформенные бизнес-модели. Организации, стремящиеся к ускорению роста и повышению устойчивости бизнеса, внедряют архитектуры на базе API для устранения или модернизации своих устаревших и монолитных приложений. Растет доля разработчиков, заинтересованных в использовании API. Новые подходы к API, основанные на событиях, инновации в проектировании и моделировании, такие как GraphQL, стимулируют интерес к управлению полным жизненным циклом API.

ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ API

Безопасность API – это защита доступа и операций вокруг программного интерфейса. По мнению Gartner, такие элементы обеспечения безопасности, как тестирование, контроль доступа и защита от угроз актуальны как никогда, поскольку API стали основным объектом атак для многих систем¹.

Salt Labs в исследовании [«Состояние безопасности API: Финансовые услуги и страхование»](#) отмечает, что **92%** респондентов за последний год столкнулись хотя бы с одной существенной угрозой безопасности в своих API, причем первое место в этом списке занимают неполадки с аутентификацией (48%).

В рамках этой задачи необходимо внедрять функциональные возможности определения личности пользователя (**аутентификация**), подтверждения доступа к ресурсам (**авторизация**), а также обеспечивать **безопасность данных при транспортировке**, операций вокруг интерфейса, а также принимать во внимание вопросы безопасности как производителей, так и потребителей API.

Для безопасного развития API в организации необходимо определить **надежную стратегию безопасности API**, оценить **текущий уровень риска** и **обеспечить безопасность** во всех средах приложений по принципу frictionless.

¹Gartner Hype Cycle for APIs 2023



ФИНТЕХ

— РАДАР

05

Управление API:



ИСТОРИЯ УПРАВЛЕНИЯ API В СБЕРЕ

Управление API в банке началось в одновременно с созданием Платформы. Платформа – это набор совместимых самостоятельно разворачиваемых или потребляемых по сервисной модели технологий и продуктов для создания и эксплуатации прикладных решений. Для разработки и использования Платформы потребовалось стандартизировать и описать API сервисов Платформы. Для описания был разработан **Реестр API**. В нем фиксировалась **техническая спецификация, поставщики и потребители API, нефункциональные характеристики**. Недостатком первой версии системы было то, что информация об API заносилась вручную командами разработки. Это приводило к дополнительным трудозатратам команд и в ряде случаев к неактуальности API.

Новый виток развития API случился в 2018 году, когда потребовалось вести учет в т.ч. API, предоставляемых неплатформенными АС, которые отличались разнообразием используемых форматов и технологий. Для управления API был разработан и внедрен «Стандарт проектирования и разработки API», регламентирующий основные моменты, связанные с управлением API.

Примерно в это же время была выпущена **новая версия Реестра API**, которая имела возможность автоматической загрузки API в Реестр. Для этого при разработке описания интерфейсов API размечались в исходном коде специальными тегами, а при сборке плагин сборщика идентифицировал API по этим тегам, обрабатывал описание интерфейса и загружал в Реестр. При внедрении этого решения, однако, не обошлось без трудностей – при разнообразии форматов API, используемых в банке, потребовалось реализовать их поддержку для автоматической обработки, а для некоторых форматов оставить возможность ручной загрузки.

Такое решение позволило **существенно повысить количество загруженных API** в Реестре и **качество их описания**, а также обеспечить **своевременную актуализацию информации** об API.

Также примерно в это же время (2017-2018 г) начали развиваться публичные API, для которых была создана специализированная платформа управления API – **SberAPI**.

В дальнейшем в банке был внедрен инструмент автоматизированной разработки и публикации заглушек для тестирования API, а также инструмент проектирования интерфейса API, который позволяет в т.ч. по описанию интерфейса внутренних API генерировать описание публичных API и публиковать их.

ВИДЫ API В СБЕРЕ

В настоящий момент выделяются следующие виды API в банке:

- 
• **Закрытые API** — API, предназначенные для использования модулями внутри АС и не предназначенные для интеграции с другими АС.
- 
• **Внутренние API** — API, предназначенные для взаимодействия АС Банка между собой.
- 
• **Публичные API** — API, предназначенные для взаимодействия АС Банка и других организаций.

К закрытым API в настоящее время специальных требований не предъявляется, т.к. с одной стороны – для них ограничено число потребителей и, как правило, поставщик и потребитель находятся в одной либо в смежных командах. С другой стороны, у таких API могут быть особенности, обусловленные технологией, производительностью и т.п. Их регистрация в Реестре API на текущий момент также является необязательной.

Внутренние API **обязательно должны регистрироваться в Реестре API**, а также удовлетворять требованиям стандарта Банка по разработке и проектированию API.

К публичным API также предъявляются **требования Стандарта**, однако их учет ведется отдельно от внутренних – непосредственно в системе **SberAPI**.

ЛАНДШАФТ API

ВНУТРЕННИЕ API

На текущий момент в реестре загружено **более 10 000** версий API.

Изначально, основной тип API, используемый в банке – это удаленный вызов на языке **Java**. **Основной недостаток** данного типа API – **языковая зависимость**, тормозящая развитие разработки и переход на новые технологии. На текущий момент актуален следующий тренд – **планомерный переход на независимые от языка типы API**, такие как:

- SOAP
- GraphQL
- REST (Open API, JSON-RPC)
- gRPC

Отдельное положение занимают API, основанные на передачи событий через Kafka или MQ, в банке принято называть их **событийные API**. Рассмотрим каждый тип подробнее.

GRAPHQL

Данная спецификация в основном используется для дизайна пользовательского интерфейса (UI) и изредка (всего несколько случаев) для интеграций. Основные сложности в использовании для интеграций следующие:

- необходимость использования ограничения стоимости запросов, т.к., возможно, потребитель попросит огромный объем данных.
- сложность контроля за «особыми» данными (коммерческая тайна и персональные данные клиентов). Потребитель может менять состав данных без уведомления поставщика.

SOAP

Устаревший формат, в основном используется для обмена с внешними организациями или в старых, legacy AC.

JSON-RPC

Особенность реестра в том, что он не выделяет JSON-RPC в отдельный тип API. Одна из причин заключается в том, что JSON-RPC не имеет стандартную спецификацию. Наибольшее распространение получил формат open-RPC. В данный момент, коллеги предпочитают другие типы спецификаций.

gRPC

Очень востребованный формат для высоконагруженных сервисов, а также получил распространение в технических сервисах платформы. За счет того сообщения передаются в бинарном (не человеко-читаемом) формате, что позволяет передавать больше сообщений в единицу времени, но несколько затрудняет отладку и разбор ошибок.

Второй недостаток – отсутствие развитых средств ограничения и точного описания контракта, что затрудняет использование.

OPEN API

OpenAPI на текущий момент является наиболее распространенным форматом. Основные достоинства – простота, развитые возможности описания (вплоть до применения инкапсуляции), возможность установки точных ограничений на формат, поддержка версионирования и различных форматов передаваемых сообщений.

СОБЫТИЙНЫЕ API

С событийными API возникла сложность – при создании Реестра и Стандарта не было понятно какой формат лучше выбрать, поскольку многие команды до создания Реестра уже вывели свои событийные API в ПРОМ. Схемы использовали разные: от бинарного AVRO, до XSD и JSON schema. Через некоторое время был найден формат описания Async API. Однако формат был новый, и возникли некоторые сомнения в том, что формат будет развиваться и будет стабильное комьюнити, однако это не подтвердилось. В данный момент Async API поддерживает или планирует поддерживать различные форматы в том числе бинарные – AVRO, PROTO.

ПУБЛИЧНЫЕ API

В данный момент, около 700 API выставлено банком как публичные на единой площадке – это API Банка и Экосистемы (**SberAPI**). На эту же площадку будут выставлены и обязательные публичные API, спецификации которых опубликовал Банк России. Для публичных API банк использует OpenAPI и изредка (где невозможно OpenAPI) SOAP. Основные функции площадки:

- публикация (информирование) о наличии API и тарифах по их использованию;
- идентификация, аутентификация и авторизация доступа;
- пре тарификация вызова API (миттеринг);
- проверки вызовов (валидация) на соответствие контракту.

СТАНДАРТ ПРОЕКТИРОВАНИЯ И РАЗРАБОТКИ API

Основным документом, регламентирующим управление API в банке, является **Стандарт проектирования и разработки API**. В стандарте определяется:

- классификация API;
- состав контракта API;
- правила регистрации API в реестре;
- правила версионирования API и требования к обратной совместимости;
- ответственность поставщика и потребителя API;
- правила наименования API, методов и структур;
- правила проектирования интерфейсов API (включая используемые технологии и форматы);
- жизненный цикл API и процедура устаревания API.

Стандарт покрывает следующие типы API: **Java API, Open API Specification, GraphQL, gRPC, событийные, JSON-RPC**.

С точки зрения контракта регламентируются:

- требования, направленные на повышение **качества описаний API** (наименование API, методов, параметров и используемых объектов данных; детальное описание API)
- требования, направленные на **повышение точности контракта** (ограничения по использованию общих типов, позволяющих передавать любые данные – например, generic, any – а также использованию строковых типов для передачи структур; ограничения на размер сообщений);
- правила **версионирования API** (использование SemVer);
- требования о **способах передачи определенных структур**, например, даты или момента времени.

ЖИЗНЕННЫЙ ЦИКЛ API

Жизненный цикл API состоит из нескольких основных шагов:

- 01 • Анализ и сбор требований.**
 На данном шаге выявляются функциональные и нефункциональные требования потребителей к API на основе бизнес-процесса, архитектуры, технического задания, а для существующих и дорабатываемых API – также по результату анализа мониторинга работы API
- 02 • Принятие решения.**
 Определяется решение: доработка существующего API, создание нового, вывода старого API из эксплуатации. На этом этапе происходит в т.ч. анализ информации об API в Реестре
- 03 • Проектирование и разработка.**
 Создание нового API по требованиям из первого шага, тестирование, проверка на соответствие стандарту и вывод на ПРОМ, а также при необходимости запуск процедуры устаревания для API, которые заменяются разрабатываемым API
- 04 • Эксплуатация.**
 Мониторинг и контроль SLA и исправление ошибок, связанных с работой API
- 05 • Устаревание.** При вводе в эксплуатацию новой версии API или нового API, заменяющего текущий, версия API объявляется устаревшей, к ней применяются ограничения по новым подключениям, и начинает течь срок до вывода из эксплуатации, в течение которого потребители обязаны перейти на обновленную версию (по умолчанию – 12 месяцев с момента выпуска новой версии)
- 06 • Вывод из эксплуатации.**
 После перехода всех потребителей на новую версию API, заменяющую текущую, можно удалить версию API

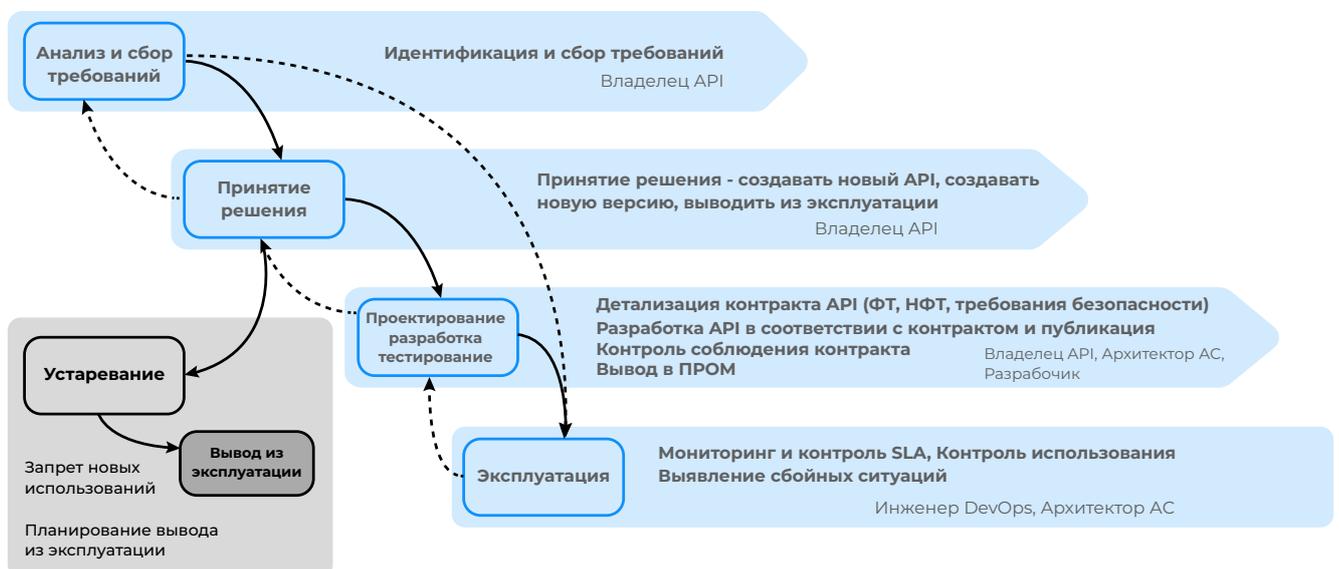


Рисунок 2: Жизненный цикл API в Сбере

ПРОЦЕДУРА УСТАРЕВАНИЯ API

Процедуре устаревания часто уделяется незаслуженно мало внимания, отчего могут возникнуть высокие издержки с точки зрения обеспечения безопасности. Российские эксперты среди новых угроз безопасности **выделяют Zombie API** – неактуальный программный интерфейс, который вследствие отсутствия регулярных обновлений становится легкоуязвимым объектом атаки. Исследование Salt Labs **показало**, что половина респондентов (48%) в области финансовых услуг и страхования выделяют Zombie API как основную проблему в области безопасности – в **3,5 раза выше**, чем утечку конфиденциальных данных.

Эксперты Сбера также поделились внедренной у них процедурой устаревания:

- При необходимости разработать и ввести в эксплуатацию замещающий API (совершенно новый API или новая мажорная версия).
- Пометить все части контракта, как устаревшие. К примеру, для этих целей в OpenAPI есть `deprecated: true`.
- Сообщить потребителям о необходимости перехода на новую версию в течение 12 месяцев или в момент, когда у старой версии не осталось потребителей. В сообщении должна быть указана причина вывода и на «что» будет заменен данный API. Потребители могут согласовать более длительное время поддержки.
- Повторно обратиться к потребителям за 3 месяца до окончания времени поддержки.
- По окончании времени поддержки необходимо удалить неэксплуатируемые части API.

РЕЕСТР API

Реестр API содержит список API в привязке к АС. Для каждого API имеется информация обо всех версиях, для каждой версии – список методов с входными и выходными параметрами, нефункциональные характеристики. Имеется возможность получить полную спецификацию API на соответствующем IDL. Также существует возможность поиска API в т.ч. по АС, названию и описанию API.

Наполнение реестра новыми версиями для поддерживаемых форматов API проводится в автоматическом режиме из исходного кода, используя инструменты непрерывной интеграции и доставки (CI/CD).

Реестр API используется **для следующих целей**:

1. Учет API, включая их функциональные и нефункциональные характеристики
2. Поиск API потенциальными потребителями – для этого реализованы широкие возможности поиска, в т.ч. полнотекстовый поиск
3. Учет потребителей API, включая их нефункциональные требования
4. Привязка API к описанию архитектуры (позволяет отследить, как именно реализуется проектируемое архитектурное решение)

В настоящее время ведется работа по более тесной интеграции Реестра API и системы детального проектирования архитектуры АС, в результате появятся следующие возможности:

- автоматизированное получение информации о потребителях
- плановая и фактическая информация о нагрузке в разрезе каждого потребителя
- согласование плановой нагрузки потребителя
- информация о стендах и версии API установленной на нем



ИНСТРУМЕНТ РАЗРАБОТКИ КОНТРАКТА API

Для разработки контрактов API в банке создали собственный инструмент. При создании контрактов API возникла проблема поддержки архитектурных стандартов и требований безопасности, особенно для публичных API. Вторая возможность инструмента — это командная разработка контракта API, что позволит поставщику и потребителю применять подход API First и быстрее согласовать контракт.

АНАЛИЗ И ОЦЕНКА ИСПОЛЬЗОВАНИЯ API

В настоящее время в Сбере API оцениваются по следующим критериям:

- 1. Соответствие Стандарту** – на основе описания API в Реестре выполняются автоматические проверки требований Стандарта, результат проверки предоставляется команде-разработчику поставщика API.
- 2. Рейтинг зрелости API**, учитывавший техническую составляющую описания контракта API. Он рассчитывается автоматически по ряду критериев, в т. ч. соответствию правилам описания контракта, правилам именования API и т.п. Рейтинг зрелости отличается от соответствия Стандарту, т.к. в нем учитываются также рекомендации и лучшие практики по проектированию.
- 3. Число потребителей и заявленная нагрузка.** Критерий является косвенным признаком качества API, поскольку отражает его «популярность» и бизнес-необходимость.

ПЕРСПЕКТИВЫ РАЗВИТИЯ УПРАВЛЕНИЯ API

Опираясь на опыт крупных организаций, можно выделить следующие перспективные направления управления API:

1. Создание интегрированной среды управления API для поддержки всех основных этапов жизненного цикла. Например:

- Интеграция системы публикации заглушек с Реестром API даст возможность переходить из UI Реестра сразу к заглушкам. Это позволит более детально ознакомиться с API и протестировать его использование.
- Интеграция инструмента разработчика API с Реестром позволит формировать заготовку API в Реестре на основе разработанной спецификации, тем самым лучше поддерживая принцип API-First. Эта интеграция позволит использовать спецификацию в детальной архитектуре и учитывать потребителей на более раннем этапе, еще до того момента, когда произойдет сборка кода API на стороне поставщика. Затем, на момент сборки, описание API будет детализировано на основе данных из исходного кода.
- Учет фактического использования API в Реестре проводится на основе данных промышленной среды с помощью автоматического анализа логов интеграционных инструментов, а также систем поставщиков и потребителей. Учет позволит сравнивать данные фактического потребления API с проектируемыми и своевременно принимать решения по развитию, оптимизации и масштабированию API.

2. Внедрение инструментов API Management и API Gateway для внутренних API. Оно позволит повысить управляемость API, поскольку их количество при использовании микросервисной архитектуры достаточно велико.

3. Тарификация использования API внутри организации. Обладая данными о фактических вызовах API конкретными потребителями, можно эффективно аллоцировать ресурсы на потребителей смежных подразделений. Такое решение позволит:

- Сильнее заинтересовать поставщиков API в подключении новых потребителей;
- Анализировать потребности в конкретном сервисе его потребителей, а также расходы на один вызов;
- Находить «убыточные» сервисы и анализировать необходимость их трансформации;
- Более точно планировать нагрузку, и, как следствие, экономить на инфраструктуре.

РЕКОМЕНДАЦИИ ПО УПРАВЛЕНИЮ API:



Андрей Пылин
Руководитель направления
Департамента корпоративной
архитектуры, Сбер

Автоматизируйте учет API и его использования в Реестре - только таким образом удастся достичь полноты, качества и актуальности данных.

Больше внимания уделяйте учету нефункциональных требований к API, это позволит своевременно реагировать на увеличение требований по производительности, доступности, задержке.



Роман Яньков
Исполнительный директор
Департамента корпоративной
архитектуры, Сбер

Заинтересуйте поставщиков API в подключении новых потребителей. Это позволит повысить качество API и сократить T2M по выпуску новой функциональности.

**Особая благодарность Роману Янькову и Андрею Пылину
за помощь в подготовке материала в качестве соавторов**



ЧЕК-ЛИСТ ПО УПРАВЛЕНИЮ ЖИЗНЕННЫМ ЦИКЛОМ API:

1. Использовать управление полным жизненным циклом API для **реализации стратегии API** и удовлетворения как технических, так и бизнес-требований к API.
2. Рассматривать **API как продукты** и ввести бизнес-метрики для оценки их эффективности.
3. Отобрать **функционально широкое решение для управления API**, включая микросервисы, многошлюзовые и многооблачные архитектуры.
4. Убедиться, что выбранное решение охватывает **весь жизненный цикл API**.
5. Использовать управление полным жизненным циклом API для **обеспечения управления всеми API**, включая интерфейсы сторонних поставщиков (частных или публичных).
6. Решения для управления полным жизненным циклом API подходят для реализации стратегии одного поставщика, а не для выбора наиболее подходящего инструментария для различных этапов жизни API. В связи с этим необходимо убедиться, что выбор решения **соответствует целям конкретной организации**.

ВЫВОДЫ

В современной цифровой экосистеме управление API играет критическую роль. Оно позволяет организациям оптимизировать свои бизнес-процессы, ускоряя разработку, обеспечивая безопасность и контроль. API отличаются от других программных продуктов и требуют особого отношения для обеспечения безопасного использования – это касается не только внешних API, но и внутренних.

Управление API должно происходить в соответствии с жизненным циклом API.

Традиционно внимание сильно сконцентрировано на первых стадиях, тогда как устаревание и вывод из эксплуатации остаются в стороне. Недостаточное внимание к этим стадиям приводит к **Zombie API** – устаревшим и необслуживаемым интерфейсам, которые выделяются экспертами как одна из наиболее сильных уязвимостей в области защищенности API.

Оценка использования API по **возможности должна проводиться всесторонне: и по техническим, и по экономическим, и по продуктовым метрикам**. Только учитывая все эти факторы ответственные за развитие API-продуктов могут делать выводы относительно эффективности конкретного интерфейса и предпринимать меры для улучшения ситуации как с конкретным API, так и со всем API ландшафтом организации.

При использовании **API необходимо учитывать конкретные требования проекта**. Например, для проектов, связанных с финансовыми операциями, необходимо использовать API с высокой степенью безопасности и защиты данных. Для проектов, связанных с обработкой больших объемов данных, необходимо выбирать API с высокой производительностью и возможностью масштабирования.

Управление API может стать **ключом к созданию новых бизнес-моделей**. С помощью API компании могут предоставлять свои услуги третьим сторонам, открывая новые возможности для монетизации. Также оно помогает организациям быть более конкурентоспособными в цифровую эпоху. Таким образом, управление API является важным аспектом разработки ИТ-систем и требует внимательного подхода к выбору типа API в соответствии с конкретными задачами.

ПОЧИТАТЬ ДОПОЛНИТЕЛЬНО ПО ТЕМЕ



Преобразование
организации по принципу
API-First



Пособие для владельцев API



Отчет об использовании
API и микросервисов за
2022 год



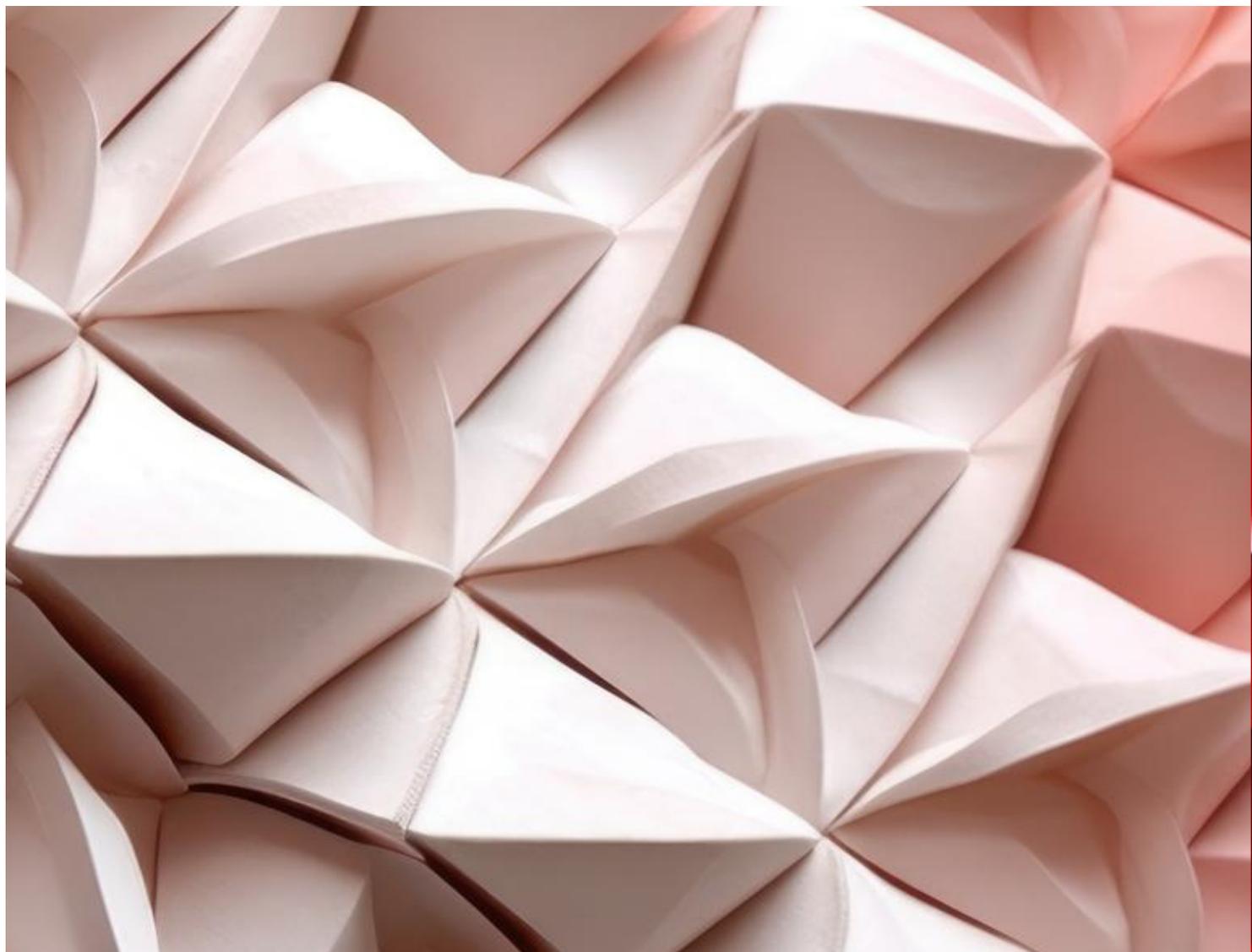
Отчет по защищенности
API в сфере финансовых
услуг и страхования



Ландшафт трафика API



API как продукт



The background is a vibrant, abstract composition of red and dark red geometric shapes. A central, three-dimensional pyramid is the focal point, rendered with sharp edges and a glossy, reflective surface. The lighting creates bright highlights on the edges and deep shadows in the recesses, giving it a sense of depth and volume. The overall aesthetic is modern and technological.

ФИНТЕХ

— РАДАР

05

ПОДРАЗДЕЛЕНИЕ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ КИТАЙСКОГО БИГТЕХА ALIBABA GROUP ВЫПУСТИЛО ДВЕ МОДЕЛИ ИИ С ОТКРЫТЫМ ИСХОДНЫМ КОДОМ.



Компания выложила в открытый доступ две большие языковые модели (LLM) – Qwen-7B и Qwen-7B-Chat. Каждая модель имеет 7 млрд параметров. Код, весовые коэффициенты и документация по обеим моделям будут находиться в свободном доступе для ученых, исследователей и коммерческих организаций по всему миру. Однако для компаний, число ежемесячных активных пользователей которых превышает 100 млн, необходимо будет получить лицензию Alibaba на использование моделей. Событие представляет собой первый случай, когда крупная китайская технологическая компания предоставляет открытый доступ к своим LLM.

GARTNER ОПУБЛИКОВАЛ РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ ИСПОЛЬЗОВАНИЯ ТЕХНОЛОГИЙ В МАРКЕТИНГЕ (MARTECH)



По данным исследования, 63% руководителей маркетинговых департаментов планируют инвестировать средства в генеративный ИИ в ближайшие 24 месяца. Опрос 405 руководителей отделов маркетинга, проведенный компанией в мае и июне 2023 года, показал, что в текущем году использование возможностей общего стека маркетинговых технологий в их организациях снизится в среднем всего до 33%. Снижение происходит второй год подряд – в 2022 году этот показатель составлял 42%.

Согласно аналитикам Gartner, повысить уровень использования маркетинговых технологий в организации довольно сложно – лишь 11% респондентов сообщили, что в 2023 году использование маркетинговых технологий увеличится более чем на 10% по сравнению с показателями прошлого года. Респонденты определили наиболее распространенные препятствия для масштабного применения martech-стека: сложность текущей экосистемы, проблемы с клиентскими данными и негибкое управление.

ЕС АНОНСИРОВАЛ ЗАПУСК ИНИЦИАТИВЫ ПО ПРЕДОСТАВЛЕНИЮ СТАРТАПАМ В ОБЛАСТИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА УСКОРЕННОГО ДОСТУПА К ЕВРОПЕЙСКИМ СУПЕРКОМПЬЮТЕРАМ.



Президент Европейского союза Урсула фон дер Ляйен призвала к созданию «новой глобальной структуры для искусственного интеллекта, построенной на трех столпах: защитные механизмы, управление и руководство инновациями». При описании необходимых защитных механизмов фон дер Ляйен сослалась на Закон об искусственном интеллекте (AI Act), назвав его «образцом для всего мира». Что касается столпа управления, президент призвала мировое сообщество сформировать совет по управлению, подобный Международной группе экспертов по изменению климата. В поддержку принципа руководства инновациями эксперт объявила об инициативе ЕС по ускорению доступа к европейским суперкомпьютерам для стартапов в области искусственного интеллекта, желающих обучать модели и проводить исследования.

OPEN AI ПРЕДСТАВИЛА МОДЕЛЬ DALL-E 3

DALL-E 3 представляет собой последнюю версию ИИ-инструмента для преобразования текста в изображение. Обновленная версия понимает значительно больше нюансов и деталей, чем предыдущие модели, что позволяет легко воплощать идеи в точные изображения. DALL-E 3 встроен в ChatGPT, что позволяет использовать последний в качестве помощника для мозгового штурма и уточнения промпта. Инструмент стал доступен пользователям ChatGPT Plus и Enterprise в начале октября.



GARTNER ПРЕДСТАВИЛ РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ ГЛОБАЛЬНЫХ РАСХОДОВ КОМПАНИЙ НА ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ И УПРАВЛЕНИЕ РИСКАМИ

Согласно новому прогнозу компании, в 2024 году мировые расходы организаций на обеспечение безопасности и управление рисками составят \$215 млрд. К концу текущего года издержки составят \$188,1 млрд. Ожидается, что в 2024 году наиболее высокими темпами будут расти расходы на обеспечение конфиденциальности данных и облачную безопасность: каждый сегмент увеличится более чем на 24% по сравнению с предыдущим годом. Конфиденциальность остается одним из приоритетов организаций, поскольку продолжают появляться нормативные акты, влияющие на обработку персональных данных, в том числе связанные с использованием искусственного интеллекта. По прогнозам Gartner, к 2025 году 75% населения Земли будут обладать персональными данными, на которые будут распространяться современные нормы конфиденциальности.



Расходы на услуги в области безопасности – консалтинг, ИТ-аутсорсинг, внедрение и аппаратную поддержку – составят порядка \$90 млрд в 2024 году, что на 11% больше, чем в 2023 году. Ожидается, что в 2024 году на услуги по обеспечению безопасности будет приходиться 42% всех расходов организаций на безопасность и управление рисками, и они останутся самой крупной статьей расходов в данной операционной области.

BAIN & COMPANY ПРЕДСТАВИЛА ОТЧЕТ ПО ТЕХНОЛОГИЯМ ЗА 2023 ГОД

Согласно результатам опроса, более трех четвертей (78%) респондентов сообщили об использовании искусственного интеллекта во всех отраслях. Компании, ожидающие внедрения ИИ, рискуют остаться позади конкурентов. Существующие инструменты искусственного интеллекта позволяют ускорить выполнение задач на 20% без снижения качества, и этот показатель, скорее всего, будет расти по мере развития технологий. Внедрение ИИ распространяется на все отрасли, повышая их производительность и эффективность. Самой высокой степенью использования ИИ отличаются системные интеграторы: 89% из них внедряют или оценивают возможности ИИ. За ними следуют направление маркетинга/рекламы – 86% и технологическая отрасль – 85%. Во всех отраслях 78% используют или оценивают по крайней мере один из шести основных вариантов использования базовой модели.



Применение ИИ широко распространено среди различных отраслей

Доля респондентов, в настоящее время использующих или оценивающих по крайней мере один из шести основных вариантов использования базовой модели ИИ, %



GARTNER ОПУБЛИКОВАЛ РЕЗУЛЬТАТЫ ОПРОСА ОРГАНИЗАЦИЙ ПО ПРИМЕНЕНИЮ ГЕНЕРАТИВНОГО ИИ



В ходе недавнего опроса более чем 1400 руководителей высшего звена компаний, 45% респондентов сообщили, что находятся в режиме пилотирования генеративного ИИ, а еще у 10% решения на основе генеративного ИИ уже находятся в промышленной эксплуатации. Это значительно больше, чем в опросе Gartner, проведенном в марте-апреле 2023 года, в котором только 15% респондентов тестировали генеративный ИИ, а 4% – внедряли его в свою деятельность.

78% респондентов считают, что преимущества генеративного ИИ перевешивают его риски. 45% предприятий масштабируют инвестиции в генеративный ИИ на несколько бизнес-функций, причем 22% – более чем на три различные функции. Разработка программного обеспечения является функцией с самым высоким уровнем внедрения и инвестиций в генеративный ИИ (21%), за ней следуют маркетинг (19%) и обслуживание клиентов (16%).

ПРОГРАММА РАЗВИТИЯ ООН (UNDP) ОПУБЛИКОВАЛА В СВОЕМ БЛОГЕ ДОКУМЕНТ, ОПИСЫВАЮЩИЙ СИСТЕМУ УПРАВЛЕНИЯ ЦИФРОВЫМИ ИДЕНТИФИКАТОРАМИ, ОСНОВАННУЮ НА ПРАВАХ ЧЕЛОВЕКА.



Концепция направлена на создание стандартной модели законов и политики, способствующих переходу на цифровые идентификаторы. Она также опирается на уроки, извлеченные из экосистемы управления идентификацией в системе ООН, основанной на регистрации актов гражданского состояния. Концепция состоит из девяти компонентов: нормативно-правовая база, отсутствие дискриминации, доступ к информации, юридическая подотчетность, дееспособные институты, ценность для пользователя, закупки и борьба с коррупцией, защита данных.





АССОЦИАЦИЯ ФИНТЕХ ИССЛЕДОВАНИЯ И АНАЛИТИКА

МАРИАННА ДАНИЛИНА

Руководитель Управления исследований и аналитики

E: m.danilina@fintechru.org



ДАРЬЯ ПЕТРОВА

Ведущий бизнес-аналитик по исследовательской деятельности

E: d.petrova@fintechru.org



ГРИГОРИЙ КАРУНАС

Бизнес-аналитик по информационным сервисам

E: g.karunas@fintechru.org



TELEGRAM КАНАЛ



WWW.FINTECHRU.ORG

Информация, содержащаяся в настоящем материале, предназначена только для информационных целей и не является профессиональной консультацией или рекомендацией. Ассоциация ФинТех не дает обещаний или гарантий относительно точности, полноты, адекватности, своевременности или актуальности информации, содержащейся в материале.

Ассоциация ФинТех оставляет за собой право вносить изменения в информацию, содержащуюся в материале, однако не берет на себя обязательств по обновлению такой информации после даты, указанной в настоящем документе, несмотря на то что информация может стать устаревшей, неточной или неполной.

Ассоциация ФинТех не проводила независимую проверку данных и предположений, использованных в настоящем материале.

Ассоциация ФинТех не несет никакой ответственности за любой ущерб, который может быть причинен в любой форме любому лицу вследствие использования, неполноты, некорректности, неактуальности любой информации, содержащейся в материале.

Материалы полностью или частично нельзя распространять, копировать или передавать какому-либо лицу без предварительного письменного согласия Ассоциации ФинТех.

