

# Первые десять лет Architecture as Code

Максим Смирнов



# Вместо слайда с био

<https://itexpert.ru>

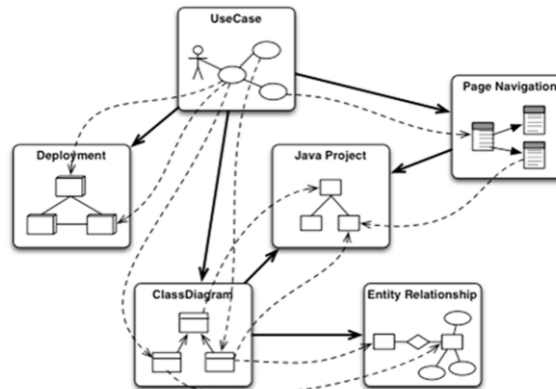
## Микросервисная Архитектура онлайн



15-18 декабря

Software architecture

## Мастерская Проектирования ИТ-решений онлайн



24-27 ноября

Solution architecture

## Модели корпоративной архитектуры онлайн



8-11 декабря

Enterprise architecture

# О чем пойдет речь (определения)



Simon Brown. Software Architecture as Code. SATURN 2015, SEI Architecture Technology User Network Conference  
(Презентация: <https://clck.ru/3MhZyq> )



Alessio Bucaioni, Amleto Di Salle, Ludovico Iovino, Patrizio Pelliccione, Franco Raimondi. Architecture as Code. 22nd IEEE International Conference on Software Architecture (ICSA 2025) <https://clck.ru/3MhaBG>

Архитектура как код (Architecture as Code, AaC) — это подход, при котором архитектура представлена в виде декларативного машиночитаемого описания, развиваемого, обновляемого и отслеживаемого с использованием инструментов непрерывной интеграции и систем контроля версий

Эта практика обеспечивает согласованность между проектированием и реализацией, непрерывную эволюцию архитектуры, создание актуальных архитектурных артефактов, активное участие разработчиков и заинтересованных сторон.

Особый интерес к AaC вызван развитием LLM

specs.c4 dynamic.c4

```

1 model {
2   customer = actor 'Customer' {
3     description 'Customer of Cloud System'
4   }
5
6   cloud = system 'Cloud System' {
7
8
9     backend = component 'Backend' {
10      description 'Backend services and API'
11
12      auth = component 'Authentication' {
13        description 'Self-Hosted Authentication'
14      }
15      auth -> cache 'keeps active sessions'
16
17      api = component 'Backend API' {
18        technology 'java/spring'
19        description '
20          Java Spring Web-service
21          provides access via REST API
22        '
23      }
24

```



Search Ctrl + K

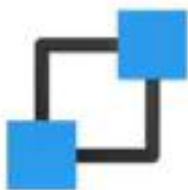


**Story 1.1 - User dashboard** id: Index

no description

**Start** ▶





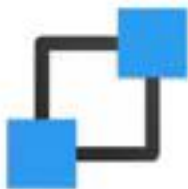
Ilograph Web

Requests



Ilograph AWS Instance

AWS Instance



Ilograph Desktop

Requests



Processing Instance

AWS Instance

Lambda



Processing Instance



Primary Instance

AWS Instance

Lambda



Primary Instance



Backup Instance

AWS Instance

Lambda



Backup Instance



Code

DNS + Origins

Get diagram

Update diagram

Request

Invite teammate

Upload icon

Purchase subscription

Claim subscription

Activate license

User collaboration

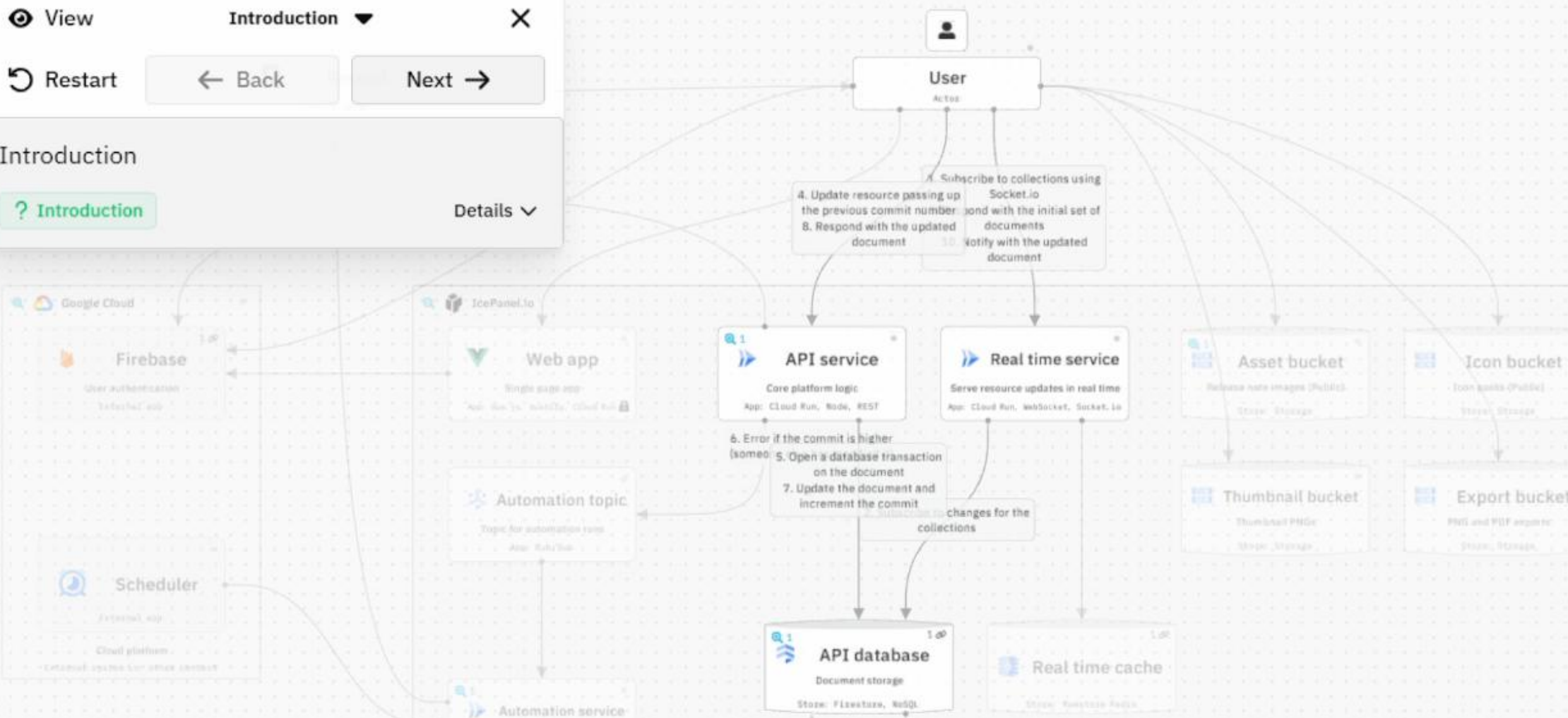
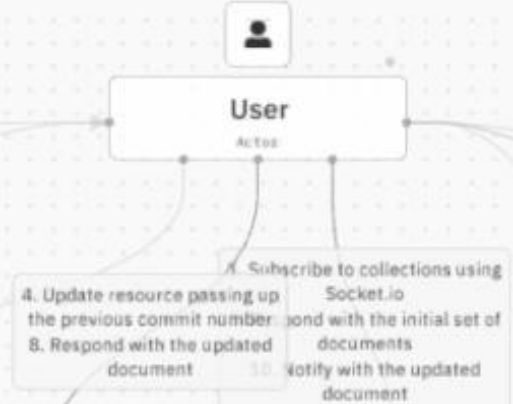
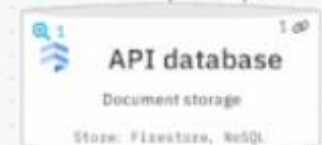
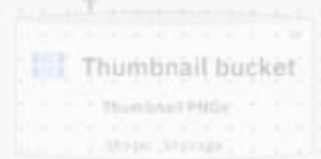
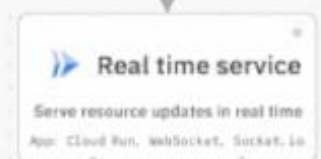
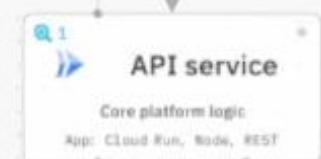
View Introduction

Restart Back Next

Introduction

Introduction

Details



# Architecture as Code

Alessio Bucaioni\*, Amleto Di Salle† Ludovico Iovino†, Patrizio Pelliccione†, Franco Raimondi†

\* Mälardalen University (Sweden), *alessio.bucaioni@mdu.se*

† Gran Sasso Science Institute (Italy), *name.surname@gssi.it*

**Abstract**—After more than thirty-five years of research and development in software architecture, several fundamental challenges remain unsolved. First, despite the importance of having a well-defined architecture description aligned with the system, inconsistencies and misalignments are still prevalent. Second, although numerous languages exist to describe architectures, none have achieved widespread use or recognition as a de facto standard. Third, while architecture is dynamic and evolving, with architectural decisions often made by non-architect stakeholders, there are no universally accepted methodologies to capture emergent aspects and incorporate them into the architecture.

In this paper, we explore the emerging concept of architecture as code. Inspired by the success of infrastructure as code, which enables infrastructure management in a codified, automated, and repeatable manner, architecture as code aims to bring similar benefits to software architecture. To the best of our knowledge, this is the first scientific paper to study this concept in depth within the context of software architecture, providing a comprehensive description and analysis of its characteristics. We also investigate how architecture as code is implemented and

have more severe consequences [49]. These inconsistencies are often discovered late in the development process [49]. Furthermore, over 70% of non-conformance issues between architecture descriptions and source code are due to flaws in the documentation [20]. The problem of inconsistency between architecture and implementation, as well as the challenge of achieving better alignment between architecture descriptions and implementation, has been explored in the context of architecture decay [34], architecture drift [38], and architecture erosion [17], [38]. This issue has also been recognized as an anti-pattern: creating a perfect architecture for the wrong system [31]. The problem is further complicated by the fact that architecture is a living entity, incorporating both upfront and emergent aspects and that some decisions are made in collaboration with stakeholders who do not hold the title of architects [45]. Researchers have proposed guidelines for software architects to improve the consistency and usefulness

# Что не так с описанием архитектуры

## Вызовы

1. Несоответствие системы и её архитектурного описания
2. Ни один ADL не получил широкого распространения
3. Нет методологии выявления значимых архитектурн. аспектов

## Вопросы исследования

### RQ1

- ✓ Обнаружить и классифицировать определения
- ✓ Обозначить ключевые концепции
- ✓ Выявить основные вызовы

### RQ2

- ✓ Инструменты Architecture as Code
- ✓ Поставщики AaC решений



**Давайте проведем опрос:** сначала здесь -  
в чате zoom, а затем в тг-канале

**Архитектура ИТ-решений**

[https://t.me/it\\_arch](https://t.me/it_arch)

(10-12 ваших вариантов ответа я предложу  
для голосования в тг.)

**Каковы основные цели**  
**Architecture as Code?**



# Сравнение с Infrastructure as Code

Инфраструктура как код (Infrastructure as Code, IaC) — это процесс управления (managing and provisioning) ресурсами центров обработки данных с помощью машиночитаемых файлов-определений, а не конфигурированием оборудования посредством интерактивных инструментов или непосредственно ...

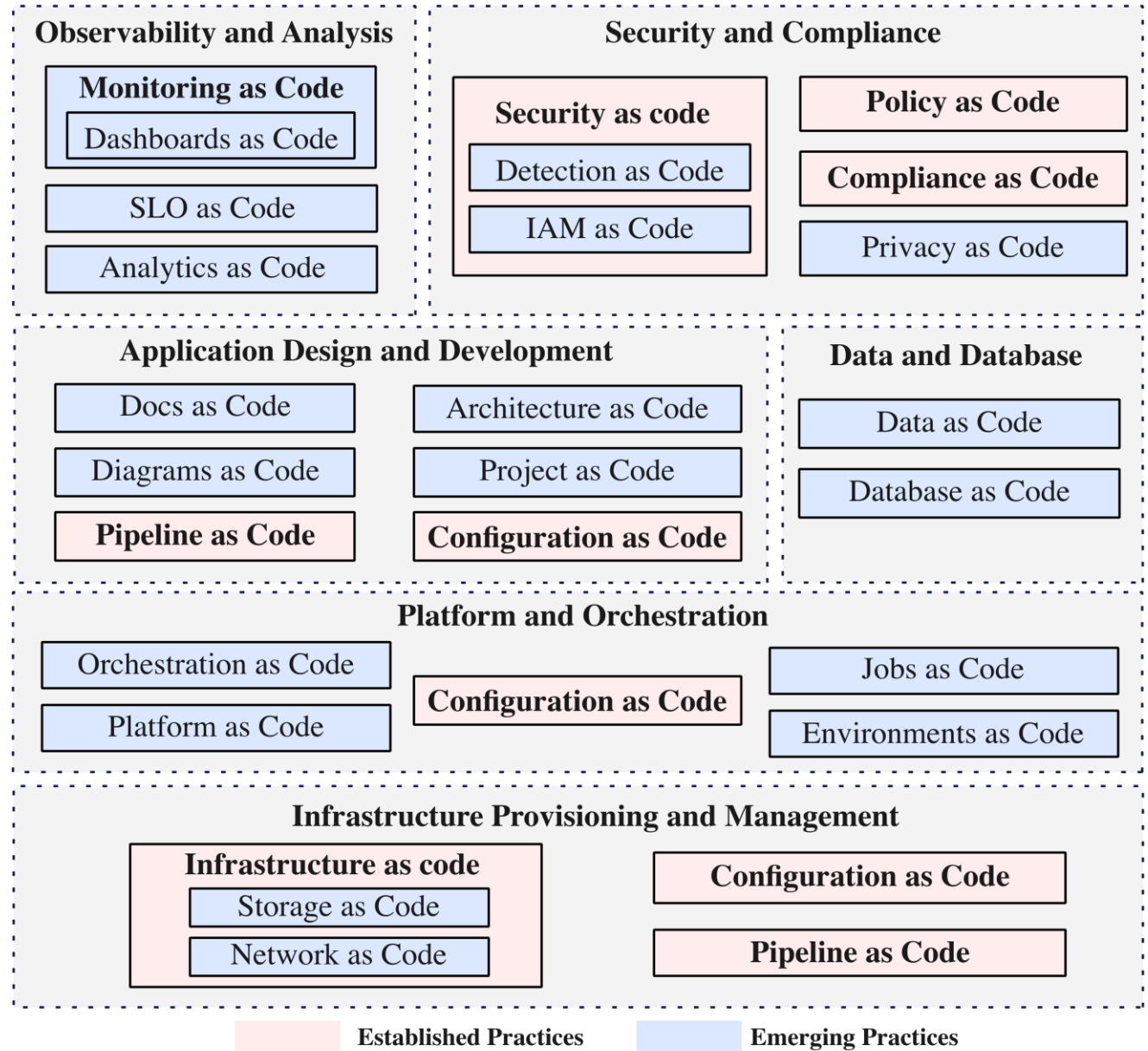
Файлы-определения могут находиться в системе контроля версий, а не поддерживаться вручную. Для управления конфигурациями могут использовать скрипты или декларативные определения, но IaC чаще использует декларативные подходы

Wittig, Andreas; Wittig, Michael (2016). *Amazon Web Services in Action*. Manning Press. ISBN 978-1-61729-288-0

# Всё остальное as code

Understanding  
Everything as Code:  
A Taxonomy and  
Conceptual Model

- 7 Jul 2025



<https://arxiv.org/abs/2507.05100v1>

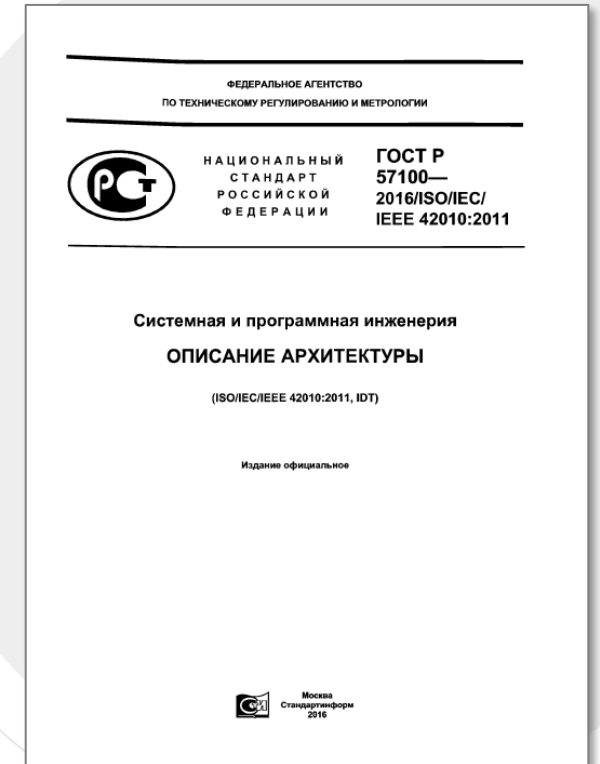
# Ожидания от Architecture as Code

1. Интеграция проектирования в процессы разработки и развертывания
2. Автоматическое создание архитектурных диаграмм из исходного кода приложений (или включение архитектуры в код, DaC)
3. Контроль соответствия кода архитектуре
4. Проверка и валидация архитектуры
5. Автоматическое создание представлений (диаграмм) из общей архитектурной модели
6. Создание и оценка архитектур AI-решениями



# Что было до Architecture as Code

1. Формальные языки описания архитектур (Architecture Description Language, ADL)
2. Системная инженерия, базирующаяся на моделях (MBSE)
3. Интеграция записей архитектурных решений (ADR) в репозитории исходного кода
4. Стандарты обмена описаниями архитектур (Например, ArchiMate® Model Exchange File Format)



# Ограничения и проблемы

1. Архитектура описывает разные аспекты системы, с разных точек зрения, в разные моменты времени и с разными целями (Descriptive vs Prescriptive Architecture)
2. Распространению IaS предшествовало появление API для конфигурирования оборудования. У приложений нет таких API
3. Синхронизация кода приложений и архитектурных описаний остается непростой задачей
4. Инструменты AaC не справляются с большими моделями
5. Ответы на вопрос: “Зачем?”. Как вписать в AaC ADR и/или элементы мотивационных моделей.
6. Современная архитектура предприятия ориентирована на выявление и планирование изменений, что делает востребованным модели реализации и миграции
7. Поддержка совместной работы

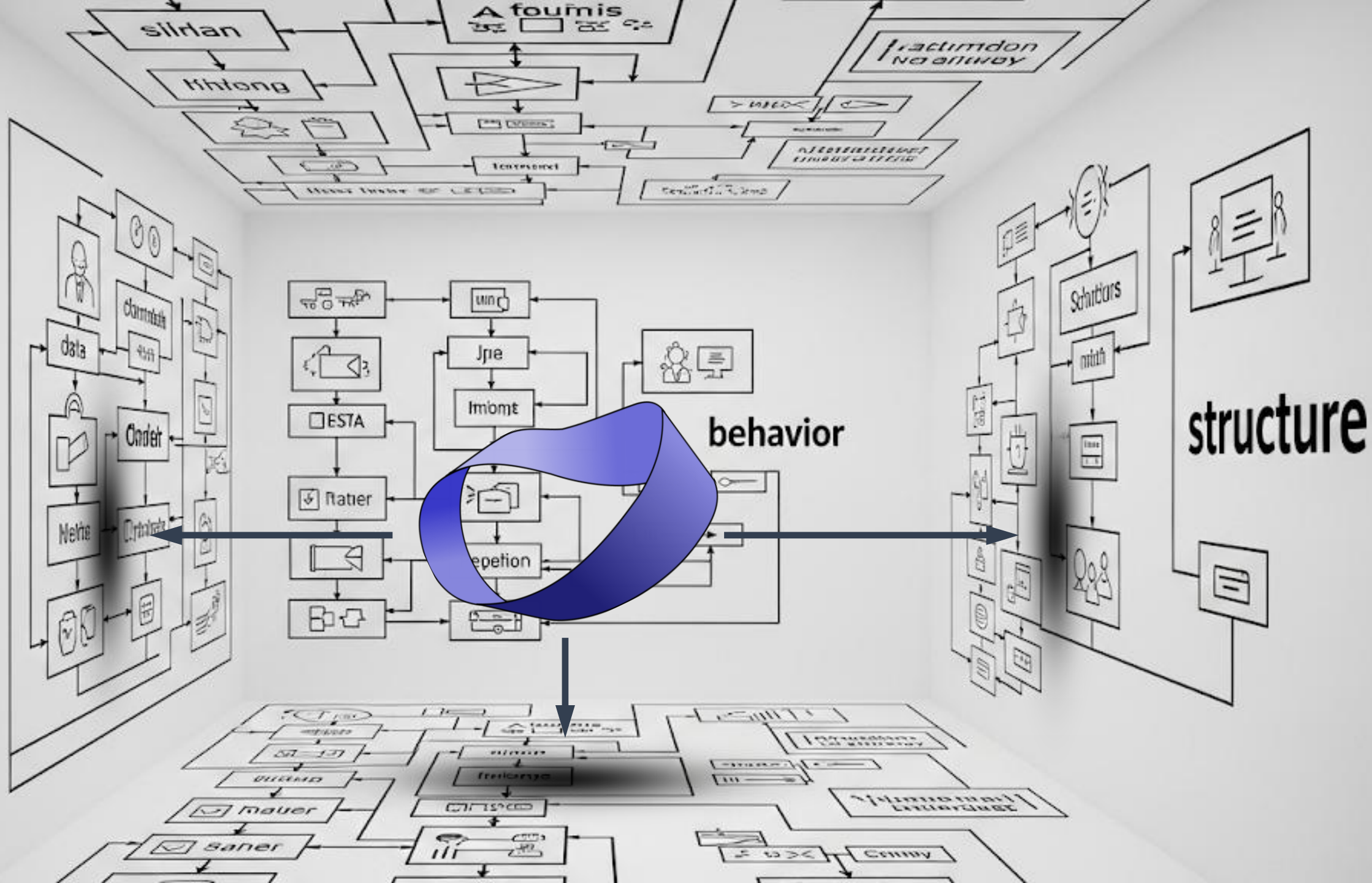
# Что принесет пользу прямо сейчас

*Diagram as Code* (DaC) — это методология, при которой диаграммы программных систем, архитектур или инфраструктурных компонентов генерируются программами, с использованием декларативного или императивного кода, а не создаются вручную в графических редакторах

- ✓ Использование репозитория ПО и инструментов CI/CD
- ✓ Версионирование, автоматическое обновление при изменениях
- ✓ Формирование диаграмм по запросу, с требуемым уровнем детализации и характеристиками диаграммы
- ✓ Возможно использование разных нотаций и инструментов визуализации
- ✓ Автоматическая проверка ограничений и валидация
- ✓ Поддержка сложных и динамических диаграмм

Насколько  
хорошо  
диаграммы  
описывают  
архитектуру?

data



# Результаты опроса

Что?

**ERD/CI. 11% (186)**

Как?

**SEQUENCE**  
36% (574)

**BPMN** 18% (294)

**FLOW** 13% (216)

Где?

**C4 Container**  
22% (353)

**Context** 18% (291)

**Deploy.** 10% (158)

**В свободном формате** 37% (594)

**Табл./текст** 17% (282)

\* тг-канал Архитектура ИТ-решений [https://t.me/it\\_arch/1720](https://t.me/it_arch/1720)

Diagram Type Optional

Sequence Entity Relationship Cloud Architecture Flowchart

Diagram Generation Prompt Required

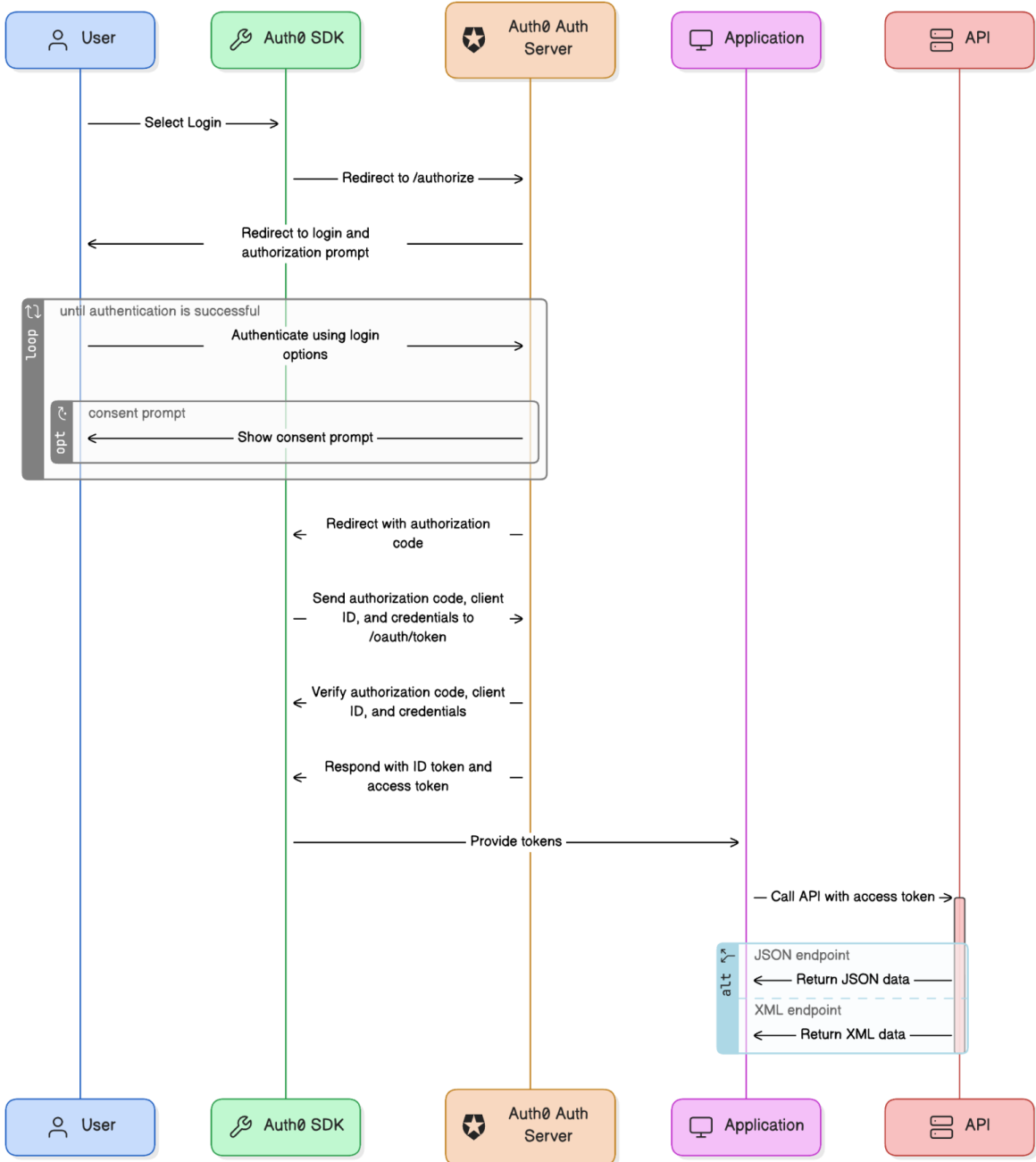
User selects Login within application.  
Auth0's SDK redirects user to Auth0 Authorization Server (/authorize endpoint).  
Auth0 Authorization Server redirects user to login and authorization prompt.  
User authenticates using one of the configured login options, and may see a consent prompt listing the permissions Auth0 will give to the application.  
Repeat this step until authentication is successful.  
Auth0 Authorization Server redirects user back to application with single-use authorization code.  
Auth0's SDK sends authorization code, application's client ID, and application's credentials, such as client secret or Private Key JWT, to Auth0 Authorization Server (/oauth/token endpoint).  
Auth0 Authorization Server verifies authorization code, application's client ID, and application's credentials.  
Auth0 Authorization Server responds with an ID token and access token (and

Great prompt length!

<https://docs.eraser.io/docs/ai-diagrams>

Generate

User Authentication Flow

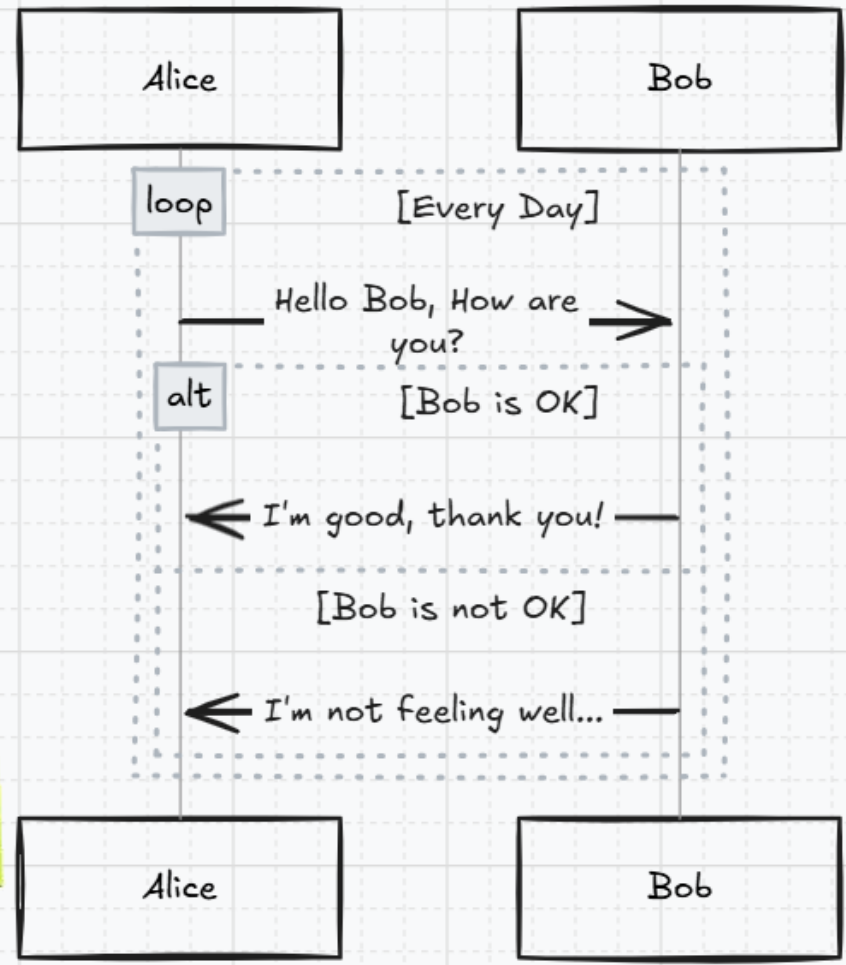


- Открыть Ctrl+O
- Сохранить как...
- Экспортировать... Ctrl+Shift+E
- Онлайн взаимодействие...
- Палитра команд Ctrl+/
- Найти на холсте Ctrl+F
- Справка ?
- Очистить холст и сбросить ц...
- Excalidraw+
- GitHub
- Подписаться
- Чат в Discord
- Sign up

Простая диаграмма последовательности с Alice и Bob, циклом и альтернативным сценарием

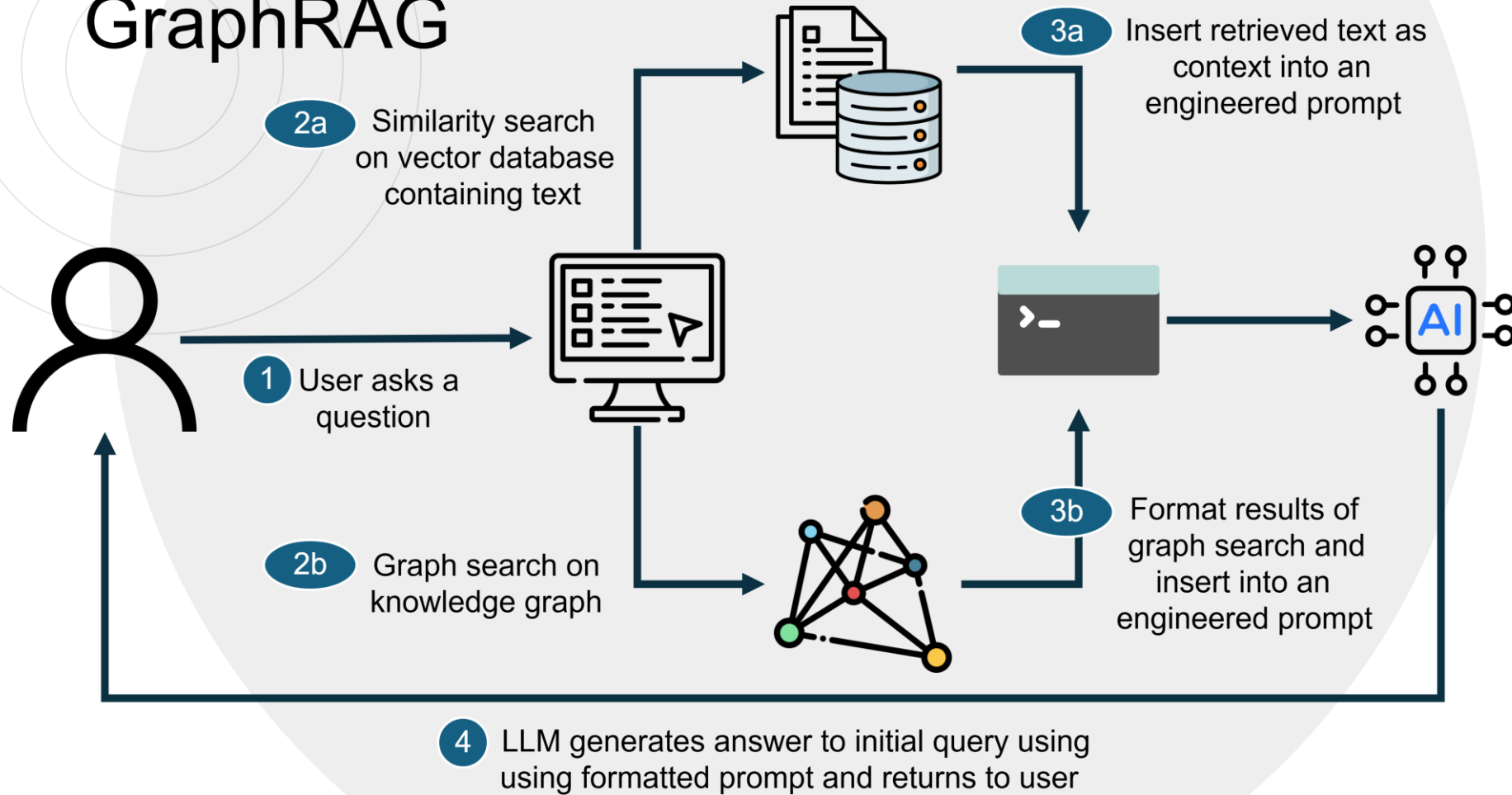


To move canvas, hold `Scroll wheel` or `Space` while dragging, or use the hand tool

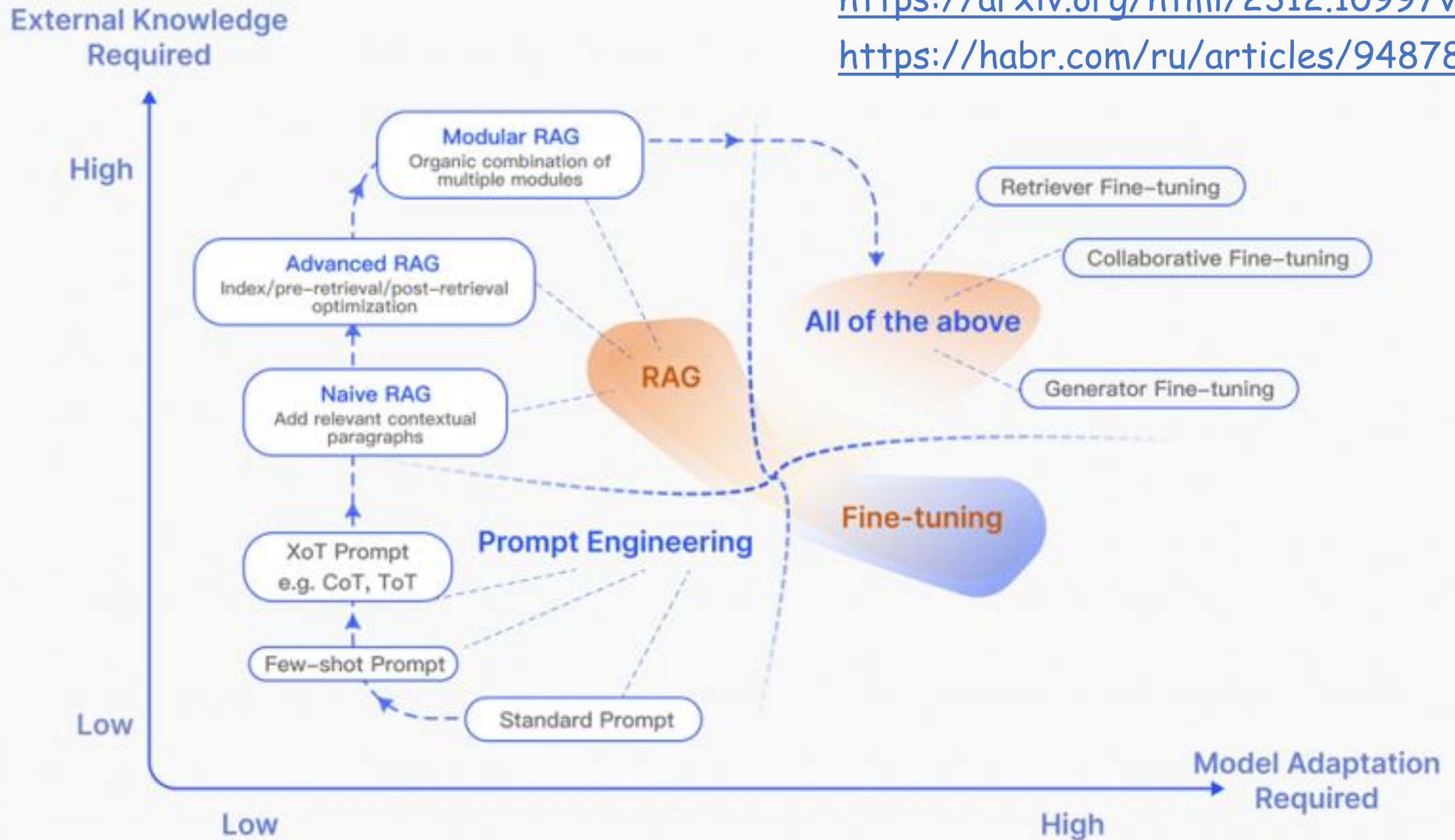


# Поговори с Architecture Repository

## GraphRAG



<https://arxiv.org/html/2312.10997v5>  
<https://habr.com/ru/articles/948786/>





# Вопросы

Первые десять лет  
Architecture as Code

Максим Смирнов