
HOW LLMS CHANGED SOFTWARE AND HARDWARE DEVELOPMENT IN A SMALL TEAM

ALEXANDER BREDIKHIN
IT4YOU
2026

CORE FOCUS — DEVELOPING DIGITAL AUDIO SIGNAL PROCESSING METHODS AND INTEGRATING THEM INTO MOBILE APPLICATIONS AND DEVICES














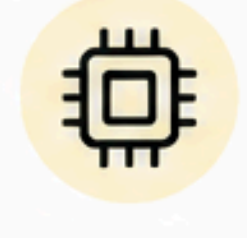



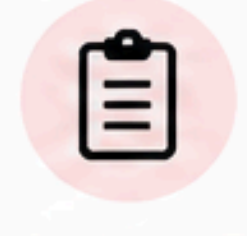





LAUNCHED APPLICATIONS

~50
APPS

TEAM EXPERIENCE

10+
YEARS

CORE TEAM

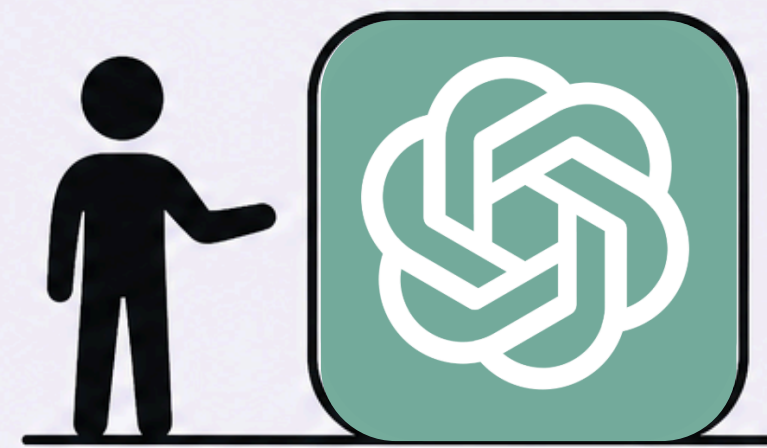
	R&D Department – 2 Mathematicians, 1 Engineer	  
	iOS Full-Stack Developers – 5	    
	Android Full-Stack Developers – 2	 
	Hardware Engineer – 1	
	Hardware Design Engineer – 1	
	Product Owners – 2	 
	QA Engineers – 2	 

NOTHING HAS BEEN THE SAME SINCE FEBRUARY 2026*

“Not a Single Line of Code Written by Hand in a Month”

— from a conversation with a senior developer on the team, April 2026

No developers were laid off 😊



GPT-5.3-Codex

—
от компании
OpenAI



Claude Opus 4.6

—
от компании
Anthropic

SOCIALAUT-REPOST REMOVER 2.0



B2C PRODUCT – 10,000 MONTHLY
INSTALLS



APP STORE



GOOGLE PLAY

A MOBILE APP THAT AUTOMATES REPOST
MANAGEMENT AND PROVIDES
ADDITIONAL MEDIA PROCESSING
CAPABILITIES

**Clean Your
TikTok Reposts**

**Remove
unwanted
reposts *fast***

**Manage what
appears on
*your profile***

CREATOR SUITE PRO

042 Total Reposts

042 Selected

Search creators...

Deselect All

@BeatFlow 2 days ago 1.4k

@StyleLens 2 days ago 1.4k

42 selected

WIPE ALL

CONTENT CLEANER

9:41

CREATOR SUITE PRO

042 Total Reposts

042 Selected

Search creators...

Deselect All

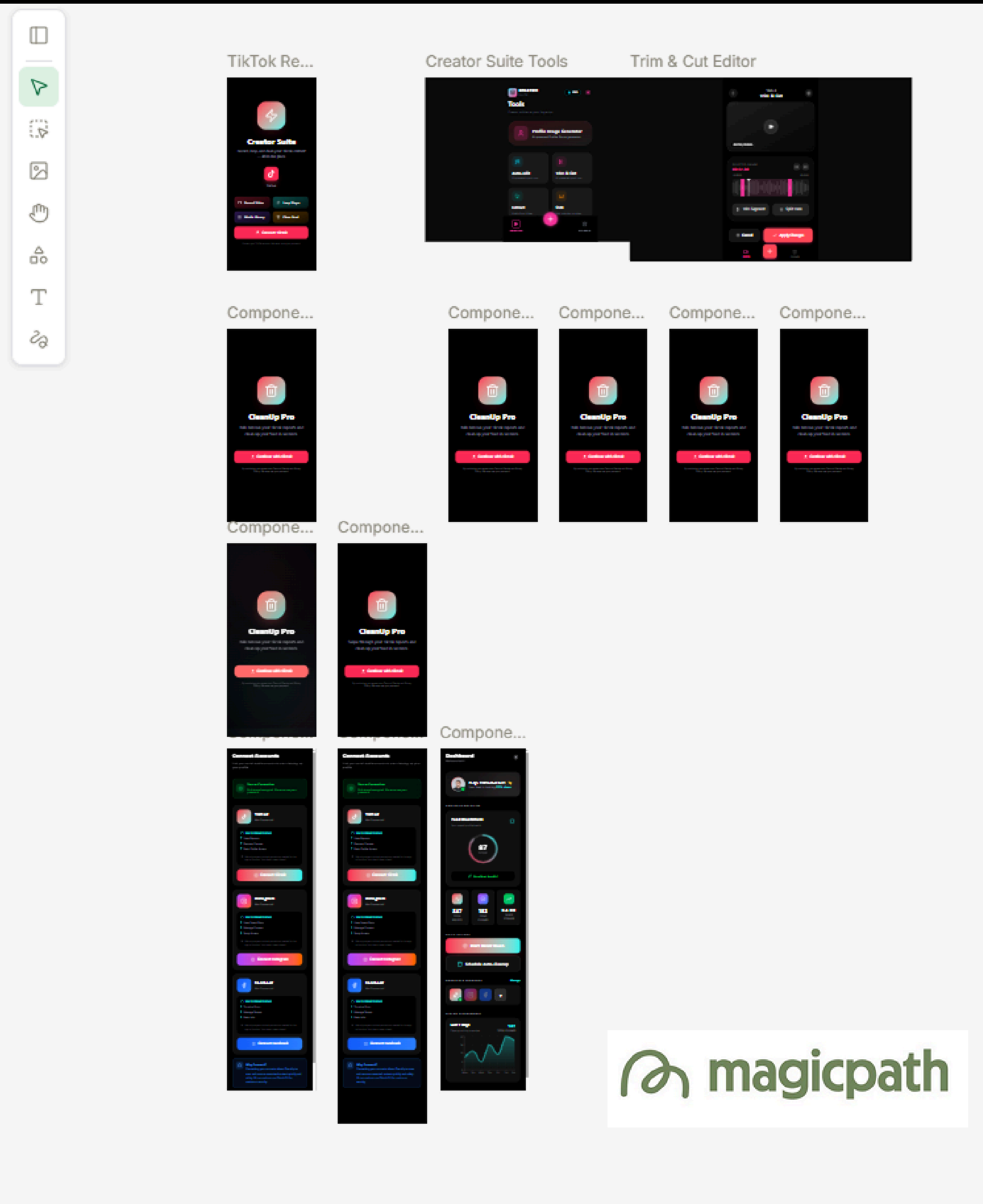
@Name 2 days ago 0.0k

@Name 2 days ago 0.0k

42 selected

WIPE ALL

CONTENT CLEANER



FROM PRODUCT IDEA TO UX WITH LLMS

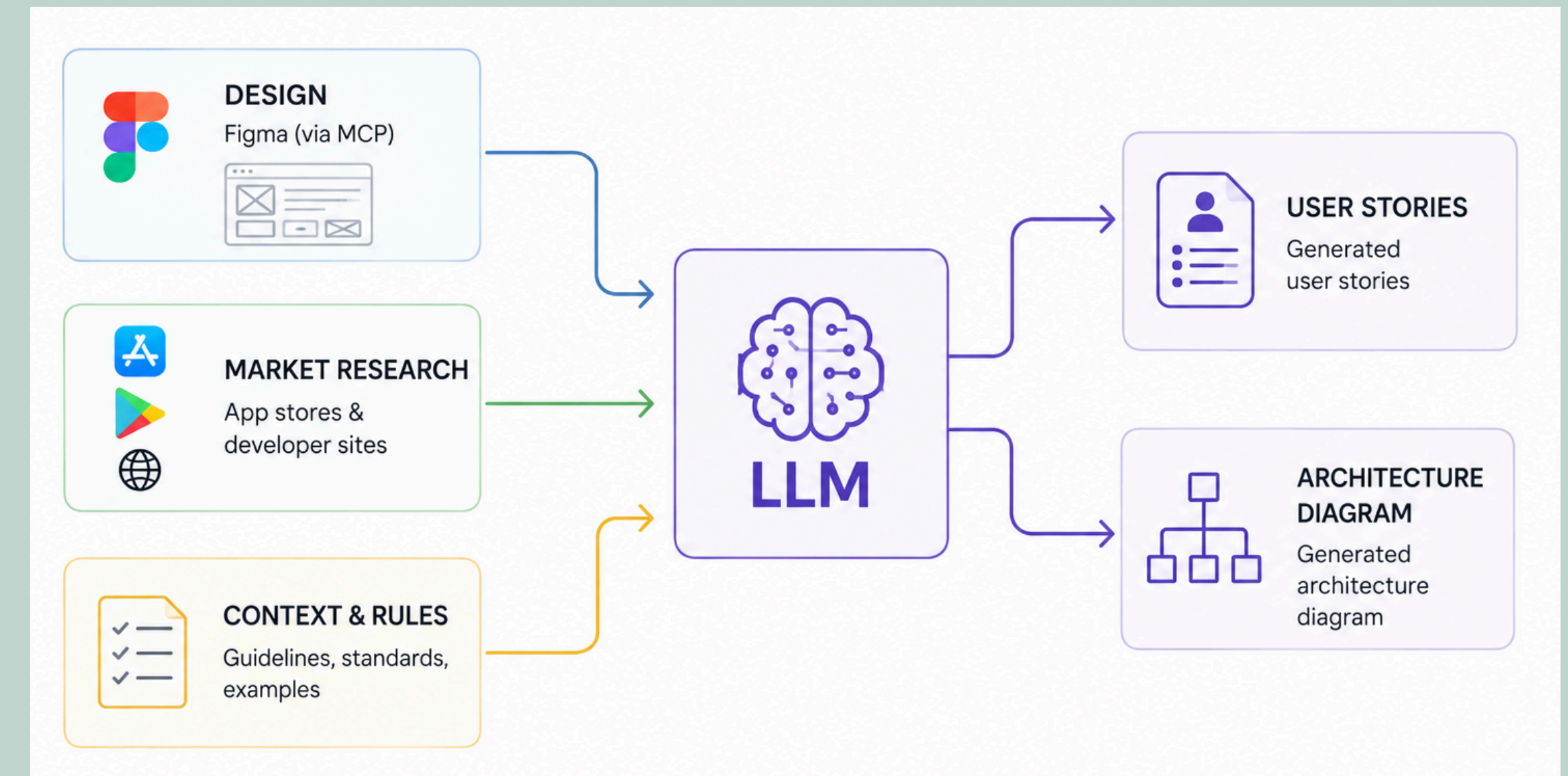
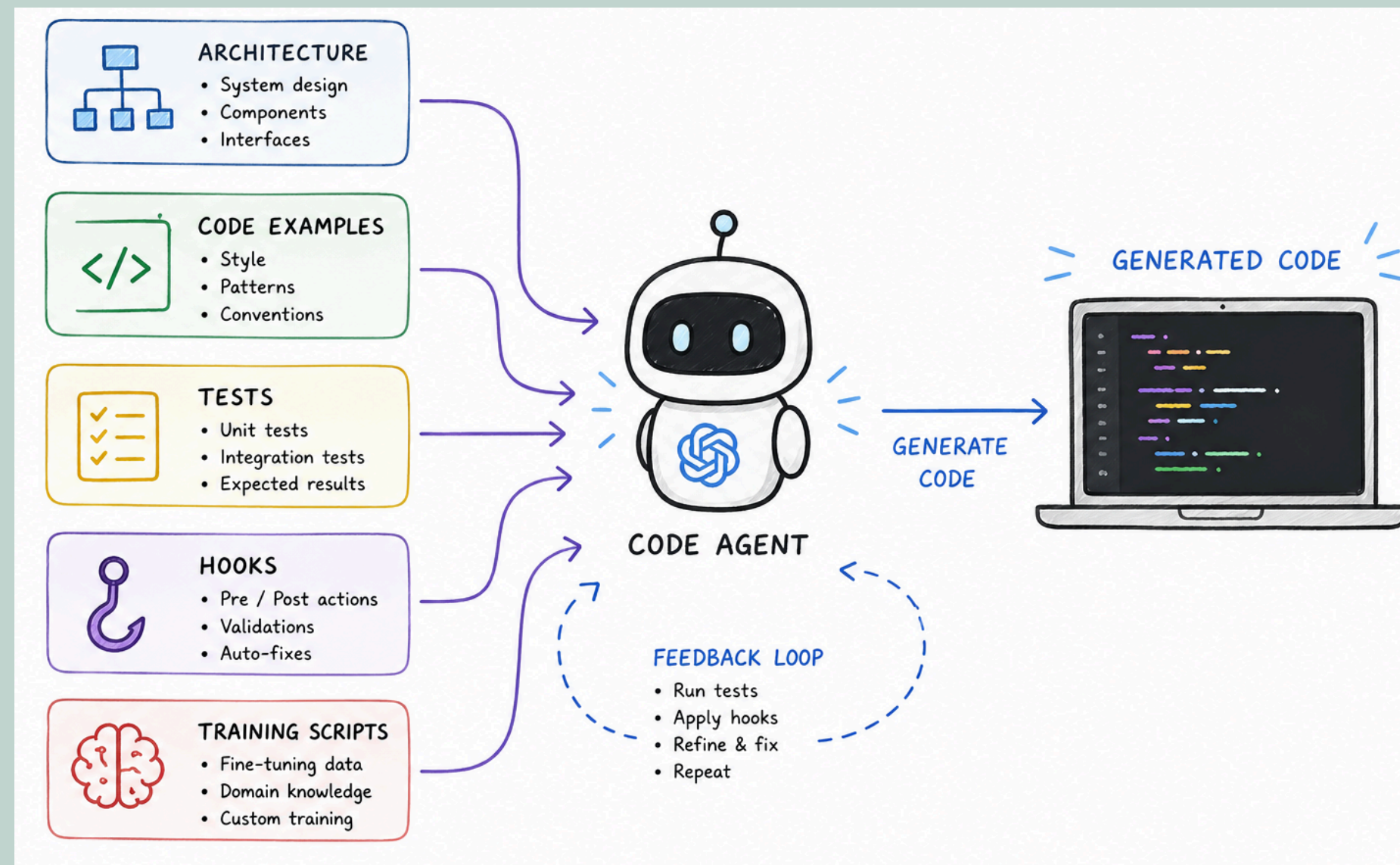
EXAMPLE: TIKTOK REPOST REMOVAL UTILITY (BULK AND AUTOMATED DELETION VIA API)

BENEFITS

- HOURS INSTEAD OF DAYS
- NO REQUIREMENTS DOCUMENTS OR SPECIFICATIONS NEEDED
 - FASTER PROTOTYPING
 - FASTER PRODUCT VALIDATION



HOW HAS THE DEVELOPER'S ROLE AND WORKLOAD CHANGED?



A good result from an LLM is not model magic — it is the result of good engineering. Rules, examples, tests, prompts, hooks, and workflows are all created by developers.

Benefits

- **No manual UI implementation from design mockups**
- **2–3× faster than conventional development workflows**

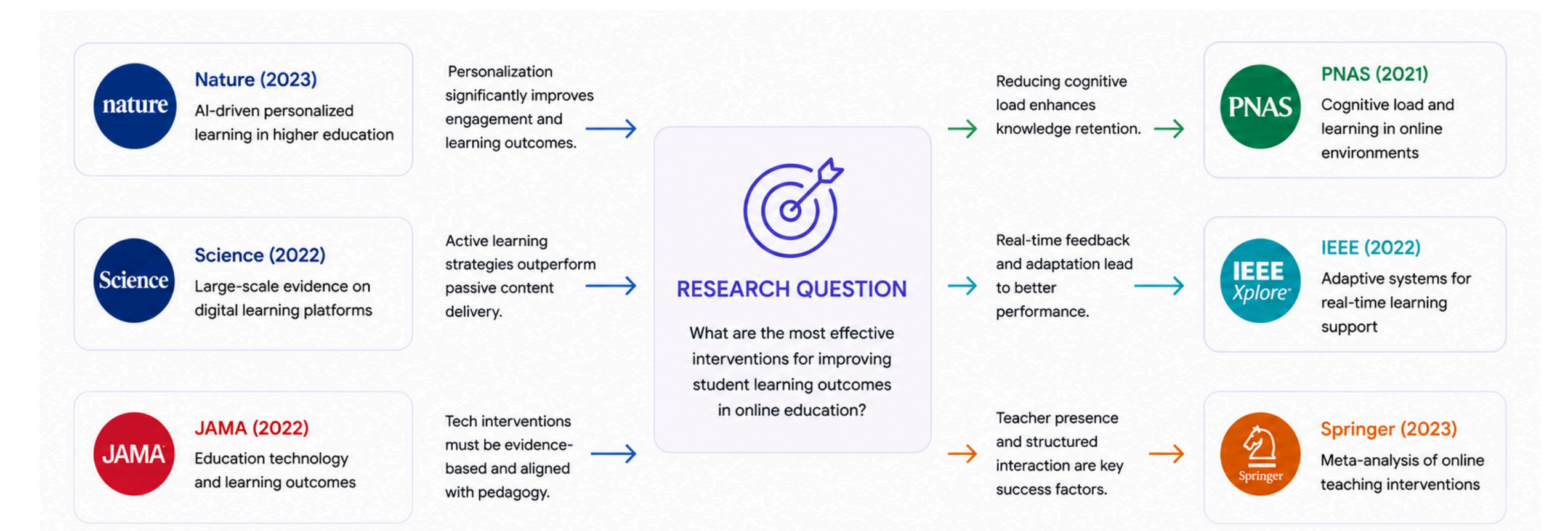
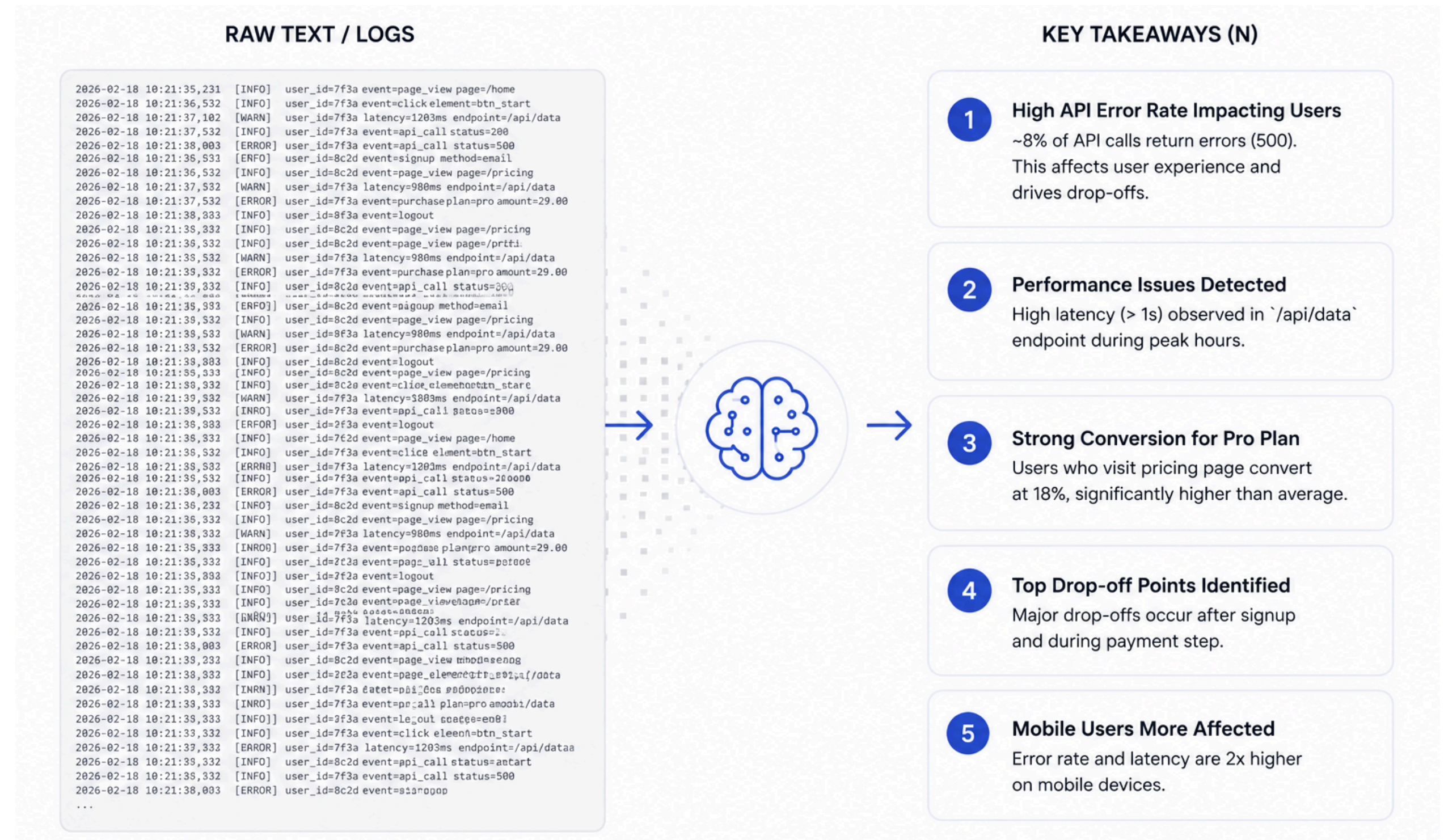
THE NEW R&D

✓ RAPID EXPERIMENT
EVALUATION

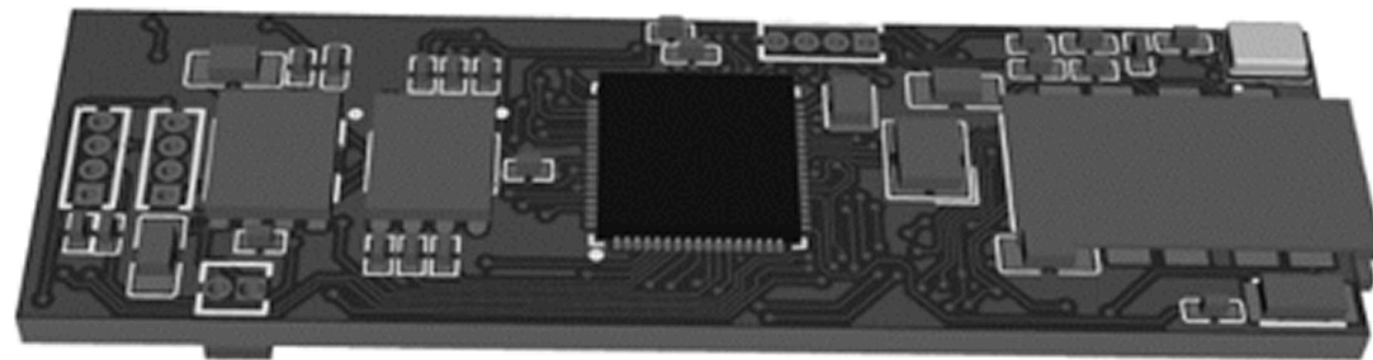
✓ SCIENTIFIC &
TECHNICAL ANALYSIS

✓ ACCELERATED
RESEARCH CYCLES

✓ FASTER HYPOTHESIS
VALIDATION



HARDWARE DEVELOPMENT IN THE LLM ERA



THE ACTION ATS2835P MICROCONTROLLER, BASED ON A 32-BIT RISC ARCHITECTURE AND FEATURING AN INTEGRATED CEVA DSP, PERFORMS REAL-TIME DIGITAL AUDIO PROCESSING.

BENEFITS

- ✓ MINUTES INSTEAD OF DAYS OF MANUAL CALCULATIONS
- ✓ FASTER ADOPTION OF CHEAPER AND MORE EFFICIENT HARDWARE COMPONENTS

Floating-Point C++ (Original)

```
#include <cmath>

float biquadProcess(
    float x,
    float b0, float b1, float b2,
    float a1, float a2,
    float& z1, float& z2)
{
    float y = b0 * x + z1;
    z1 = b1 * x - a1 * y + z2;
    z2 = b2 * x - a2 * y;
    return y;
}

// Example usage
float y = biquadProcess(x,
    0.2929f, 0.5858f, 0.2929f,
    -0.0000f, -0.1716f,
    z1, z2);
```

- ✓ Uses float/double variables
- ✓ Floating-point arithmetic
- ✓ Higher memory and CPU usage

FLOATING-POINT TO FIXED-POINT

Fixed-Point C++ (Optimized)

```
#include <cstdint>

// Q1.15 format (16-bit fractional)
int16_t biquadProcess(
    int16_t x,
    int16_t b0, int16_t b1, int16_t b2,
    int16_t a1, int16_t a2,
    int32_t& z1, int32_t& z2)
{
    // 64-bit accumulator for intermediate results
    int32_t y = ((int32_t)b0 * x >> 15) + z1;
    z1 = ((int32_t)b1 * x >> 15) - ((int32_t)a1 * y >> 15) + z2;
    z2 = ((int32_t)b2 * x >> 15) - ((int32_t)a2 * y >> 15);
    // Saturate to 16-bit output
    if (y > 32767) y = 32767;
    else if (y < -32768) y = -32768;
    return (int16_t)y;
}

// Example usage (coefficients in Q1.15)
int16_t y = biquadProcess(x,
    9629, 19258, 9629, // b0, b1, b2
    0, -5636, // a1, a2
    z1, z2); // state
```

- ✓ Uses int16_t / int32_t variables
- ✓ Fixed-point arithmetic with bit shifts
- ✓ Lower memory usage and faster on MCUs (no FPU required)

Key Changes

- Integer types
int16_t / int32_t
- Bit shifts instead of floating-point divisions
- Scaled coefficients (Q1.15 format)
- Faster and more efficient on resource-constrained MCUs

- ✓ MORE DEVICES, LOWER DEVELOPMENT COSTS
- ✓ FASTER MICROCONTROLLER CODE MIGRATION AND REFACTORING
- ✓ RAPID FLOATING-POINT → FIXED-POINT CONVERSION

VERSION CONTROL AND PROJECT KNOWLEDGE MANAGEMENT

awesome-project / README.md

Awesome Project

build passing coverage 92% code style black license MIT

A modern, high-performance toolkit for building awesome things.

Table of Contents

- Overview
- Features
- Installation
- Quick Start
- Configuration
- Project Structure
- Contributing
- License

Features

- Fast and efficient
- Easy to use
- Well documented
- Highly extensible
- Cross-platform support

Project Structure

```
awesome-project/  
├── src/           # Source code  
├── tests/        # Unit tests  
├── docs/         # Documentation  
├── examples/    # Example projects  
└── README.md    # You are here
```

Requirements

Component	Version	Notes
Python	>= 3.8	Required
CMake	>= 3.16	Build system
Git	>= 2.20	Version control

Quick Start

```
1 git clone https://github.com/octo-org/awesome-project.git  
2 cd awesome-project  
3 pip install -r requirements.txt  
4 python setup.py install  
5 awesome --help
```

Automated Reports & Detailed
Architecture Documentation in
Markdown

```
user@devbox awesome-project % git status  
On branch main  
Changes to be committed:  
  (use "git restore --staged <file>..." to unstage)  
    modified:   src/auth/oauth2/login.cpp  
    modified:   src/auth/oauth2/token.cpp  
    new file:   docs/architecture/auth-flow.md  
  
user@devbox awesome-project % git commit -m "feat(auth): add OAuth2 login\n- Implement token refresh\n- Update documentation"  
[main a1b2c3d] feat(auth): add OAuth2 login  
3 files changed, 142 insertions(+), 23 deletions(-)  
create mode 100644 docs/architecture/auth-flow.md  
user@devbox awesome-project %
```

feat(auth): add OAuth2 login

John Developer committed just now

3 files changed +142 -23

Summary

Add OAuth2 login flow with token refresh support.

Motivation

Improve authentication by integrating OAuth2 provider and keep user sessions fresh.

Changes

- Add OAuth2 login flow
- Implement token refresh
- Update documentation

Files Changed

File	Insertions	Deletions
src/auth/oauth2/login.cpp	+78	-5
src/auth/oauth2/token.cpp	+45	-10
docs/architecture/auth-flow.md	+19	0

Stats

+142 insertions
-23 deletions

Related Issues

Closes #42
Related to #17

Version Control Systems & Commit
History

GROWING ENGINEERS FASTER

- The Android version of the TikTok application was developed by an engineer for whom it started as a learning project.
- AI identifies bugs before they reach the main testing phase.
- Junior developers learn from AI explanations
- Less time on syntax, more time on product goals

WILL ROBOTS REPLACE
PROGRAMMERS? PLEASE DON'T

ASK. 😊