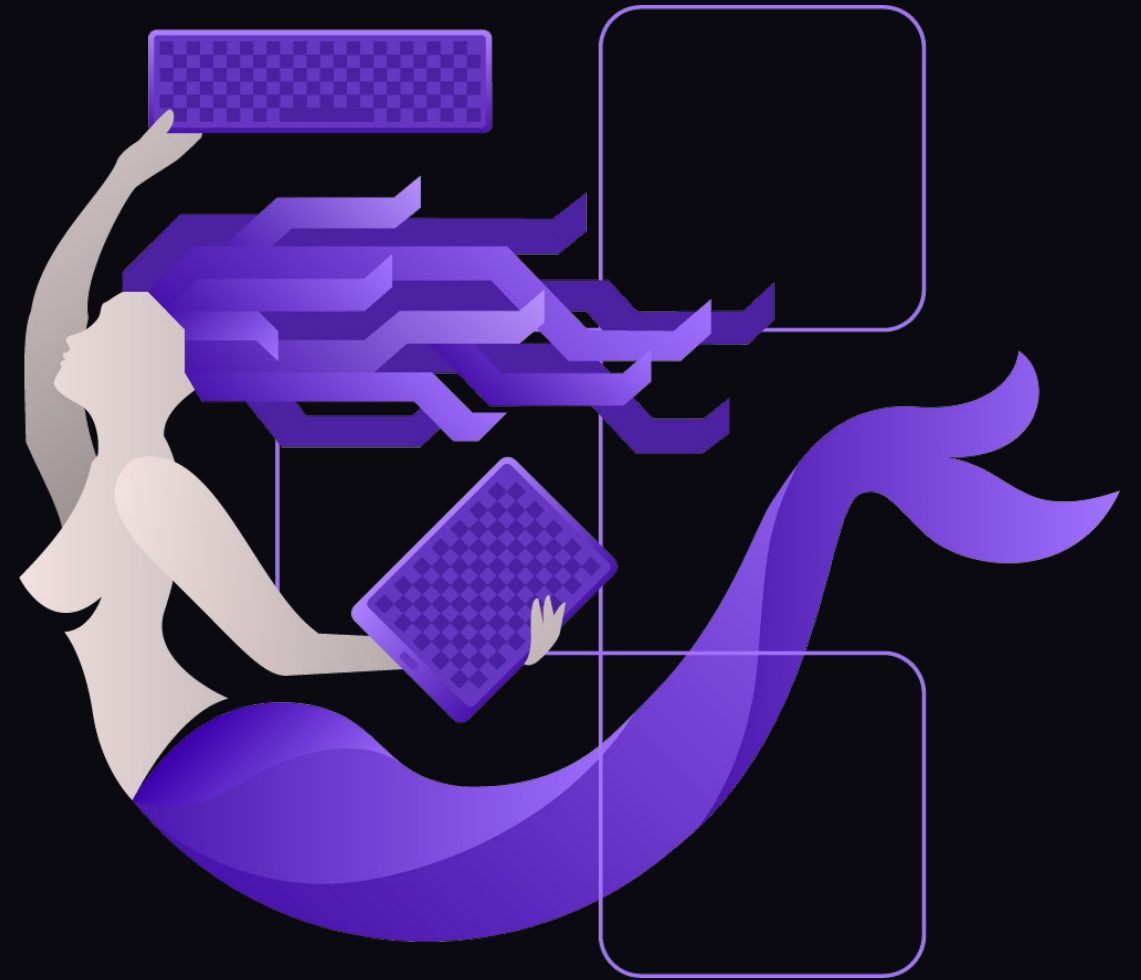


# Miserable Microservices Misconceptions

Max Małeckki



Cześć!

**Jestem**

**Max**

Od **17** **lat** psuje  
**polski internet**

Obecnie od **3 lat** pracuje dla **bitnoi.se**  
Jako

# Senior Software Engineer

**SEÑOR  
DEVELOPER**



## Disclaimer:

Przy pisaniu tej historii **nie ucierpiał**, żaden **startup** ani **software architect**.

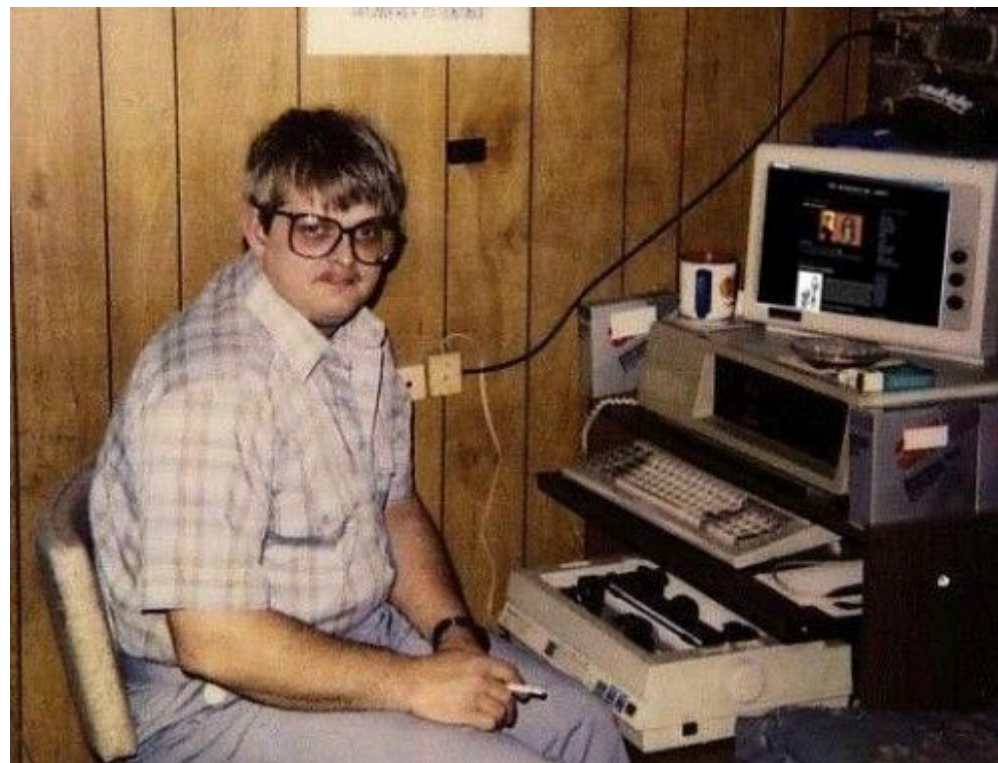
Zbieżność imion i wydażeń jest **całkowicie przypadkowa**.

Viewer discretion is advised.

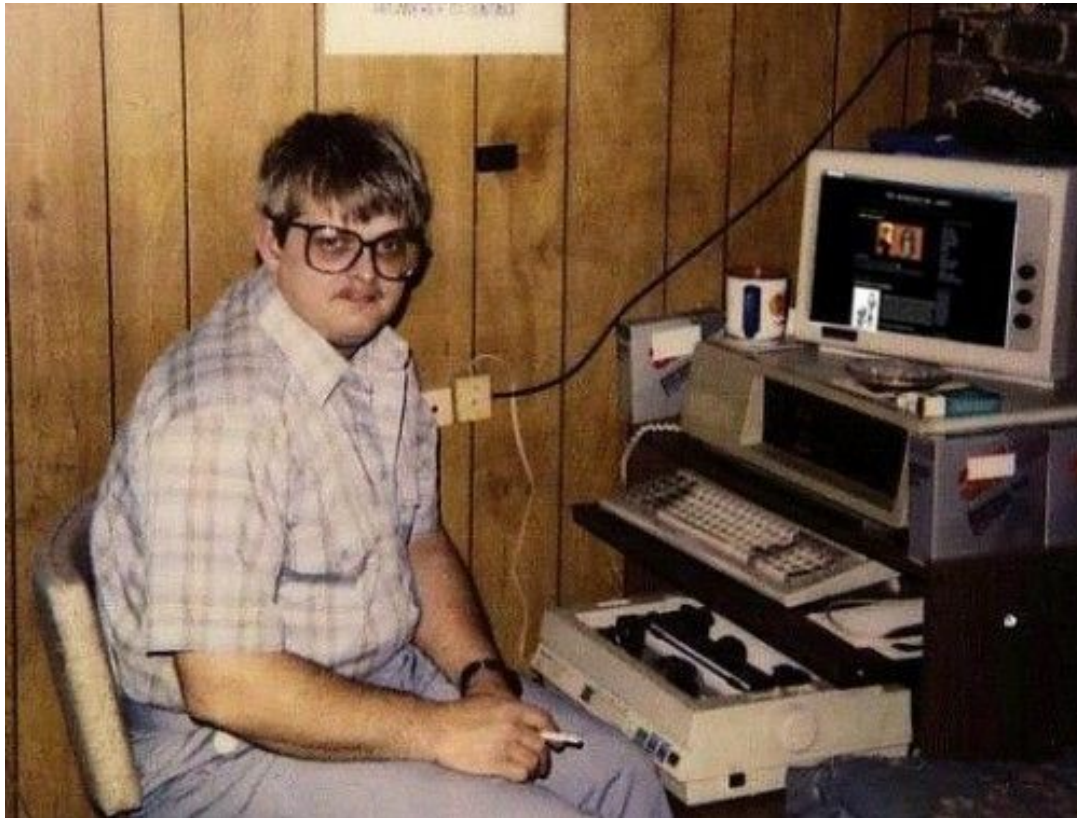
Opowiem wam dziś pewną historię.

Historię **nieszczęśliwej** miłości,  
**bajkowych pieniędzy**, nienawiści i  
**niespełnionych ambicji**. Pełną zwrotów  
akcji i **mikroserwisów**.

Która zaczyna się niezwykle  
**nudno.**



# Krzysiu



- Krzysiu pracuje w projekcie X od samego początku
- Krzysiu jest Lead developerem
- Projekt na siebie zarabia.
- Biznes się rozwija.
- Krzysiu kocha to co robi.

Wszystko idzie w  
**dobrym**  
kierunku.

Projekt zarabia. Nie jest już  
**Startupem!**

Każdy pracownik **liczy** ile warte zaraz  
będą **obiecane** mu **udziały**.

**Aż tu nagle...**

**WE DON'T ALWAYS GET  
\$18MM IN VC FUNDING**

Nies

się...

**BUT WHEN WE DO, WE CREATE  
HILARIOUS INTERNET MEMES  
ABOUT IT TO TELL THE STORY**





Projekt ma **ogromny** potencjał.

**Światowe** rynki czekają.

**Jesteś zwycięzcą!**

Nasze usługi kupią **miliony** osób na całym świecie!

Trzeba tylko  
**Przeskalować  
Projekt!**

Krzysiu myśli sobie:

**„Tak jasne tylko  
przeskalować...”**

Tylko jest jeden malutki

**Haczyk...**

Krzysiu, masz na to  
**3 miesiące.**

**NIE NO SERIO**



**NIE DA RADY**

**N**

**?**

No ale **jak to**, że się

**nie da?**

Trzeba zatem:  
**znaleźć idiotę**  
który **nie wie**,  
że się **nie da**  
i to  
**zrobi.**

## Me Spending My Parents Money

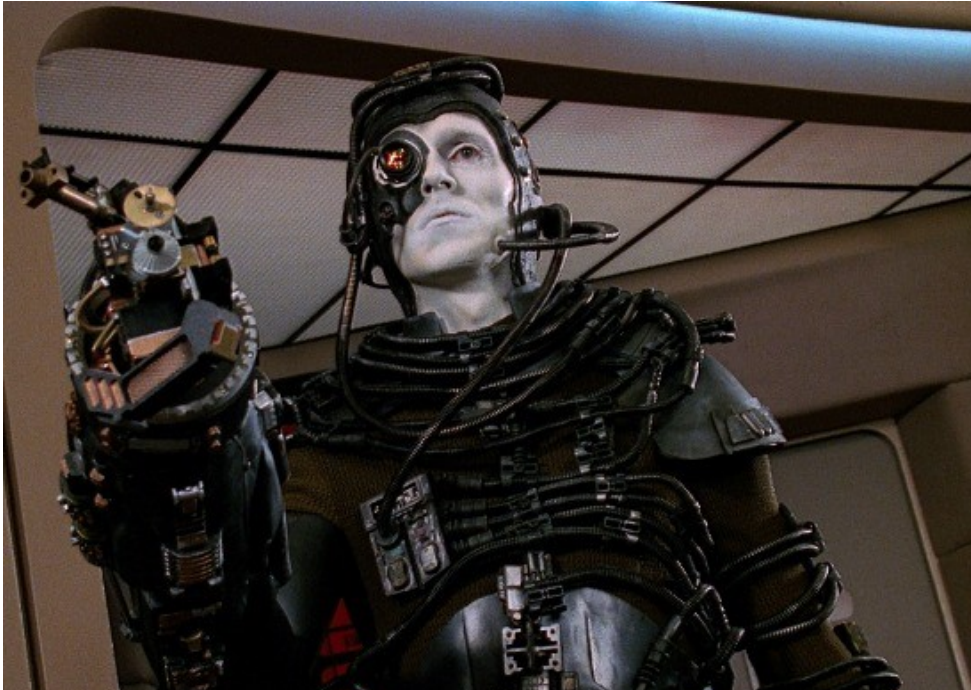


## Me Spending My Own Money





# Zenon – the architect

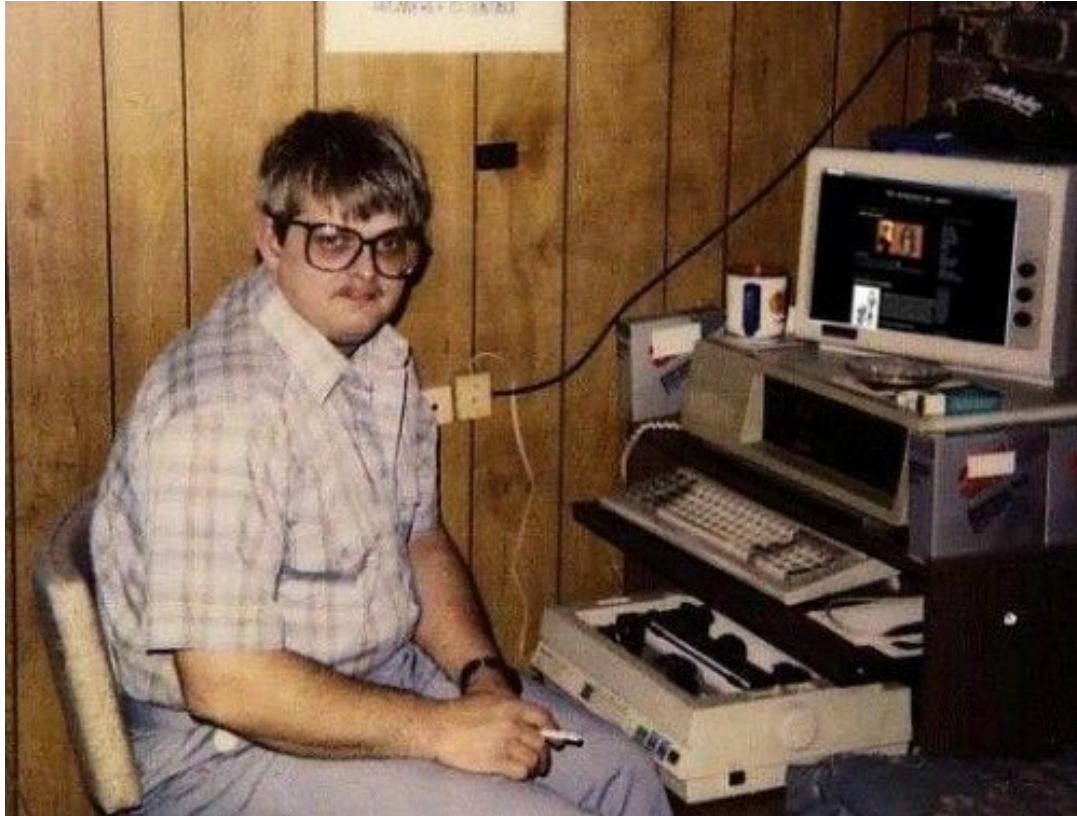


- Choć wcześniej pracował jedynie przy „łopacie”
- Zgodził się na awans na **Architekta**.
- Typowy chciwy „Yes Man”
- „Nikt wam tyle nie dowiezie ile ja wam mogę obiecać”
- Inwestor wybrał go bo był **najtańszy**.
- Oto nasz „pożyteczny idiota”

# 3 miesiące.

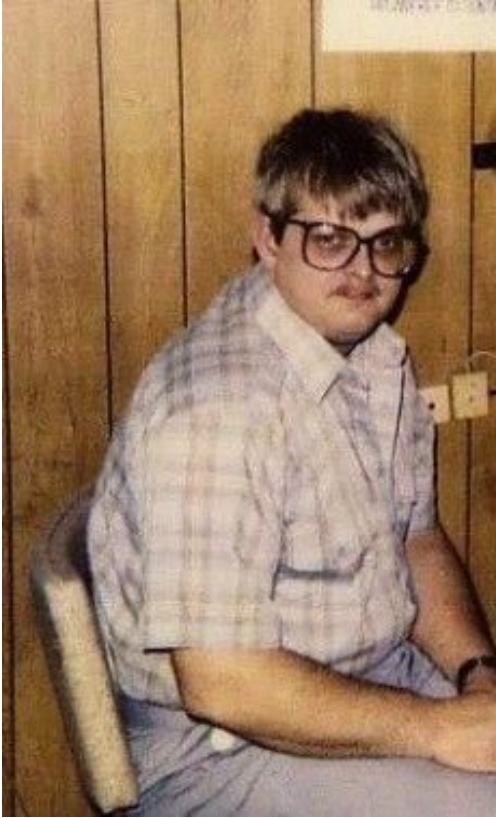
**Czas start**

# A co z Krzysiem?

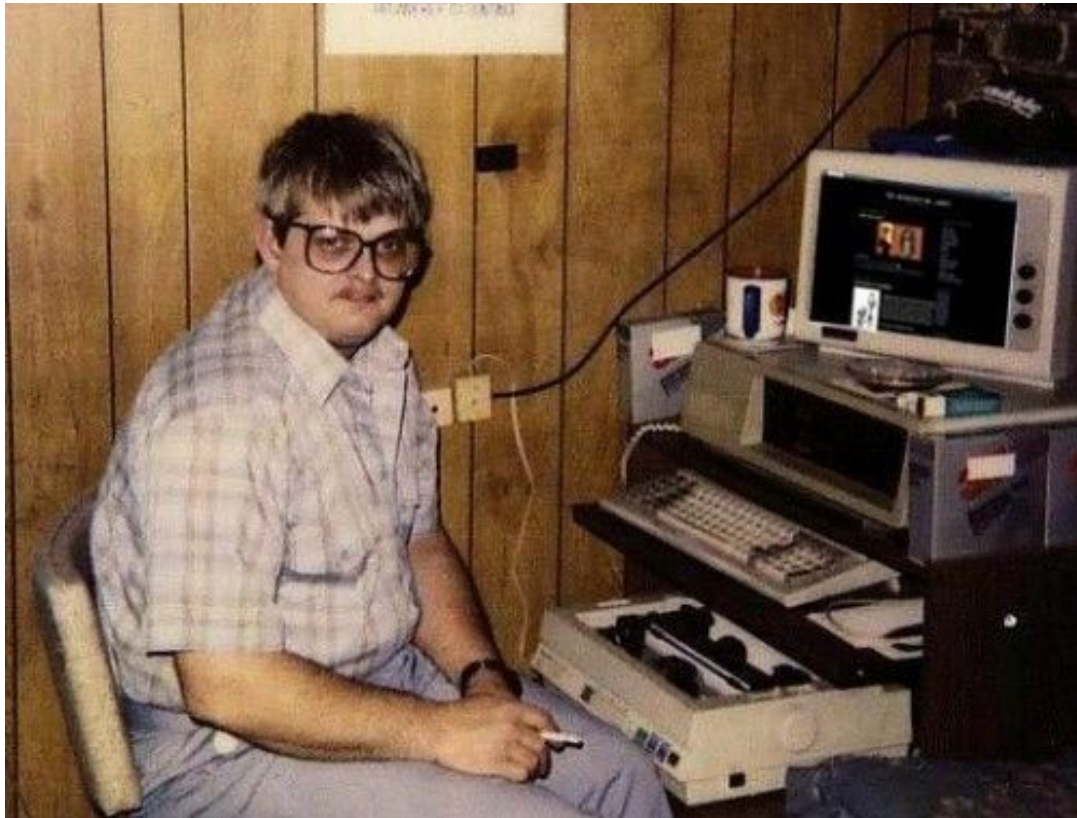


- Krzysia **zdegradowano** do roli Maintainera wygaszanego projektu...
- Jako **ekspert domenowy**, pojawia się na spotkaniach organizowanych przez Architekta.
- Jest co coraz bardziej **nie na rączkę temu Architektu**, bo pytaniami podważa jego bajkowe wizje przedstawiane inwestorowi.

# Konflikt, niekompetencja czy urażone ambicje?



# Krzysiu chce dobrze.

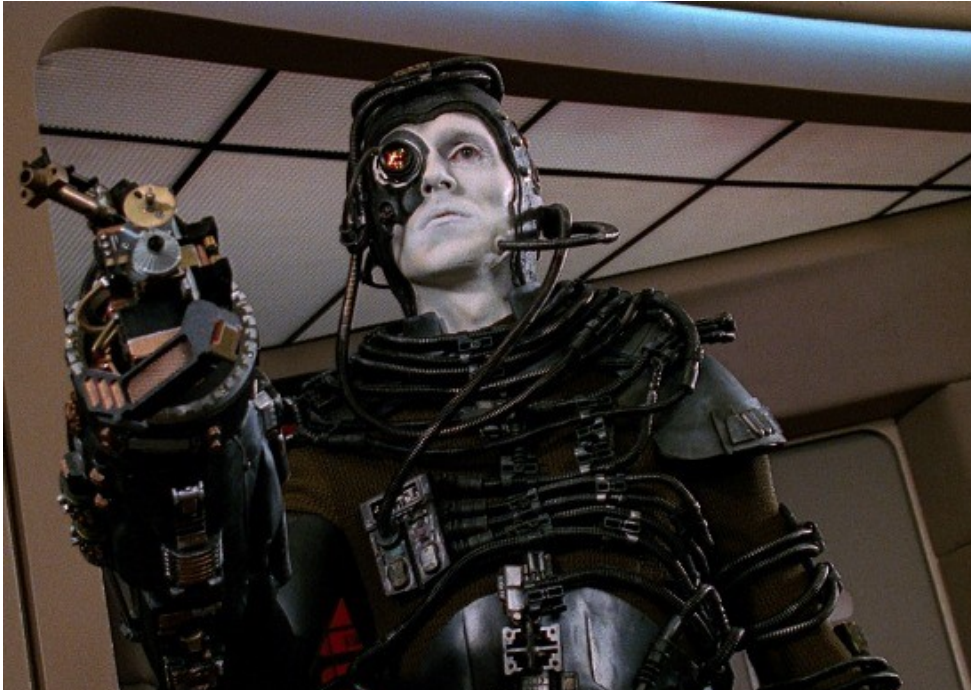


- Krzysiu stara się być **obiektywny**.
- Jego uwagi nie służą **podkopywaniu autorytetu Architekta**.
- Krzysiu chce **dobrze dla projektu**.
- Widzi **spore ryzyko** gdy plan Architekta zostanie wcielony w życie.
- Niestety **CEO** zaślepiiony pieniędzmi inwestora nie chce słuchać Krzysia.

To jaki właściwie jest **plan** Panie

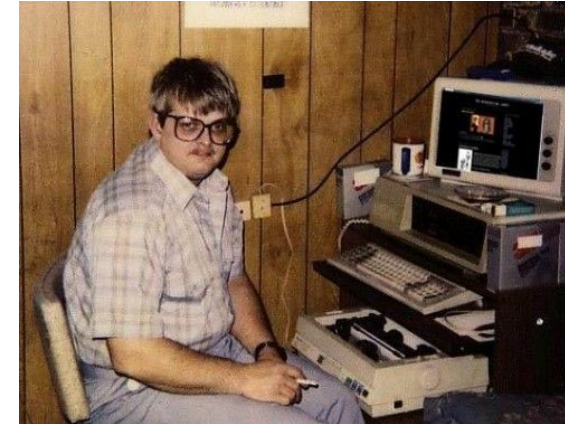
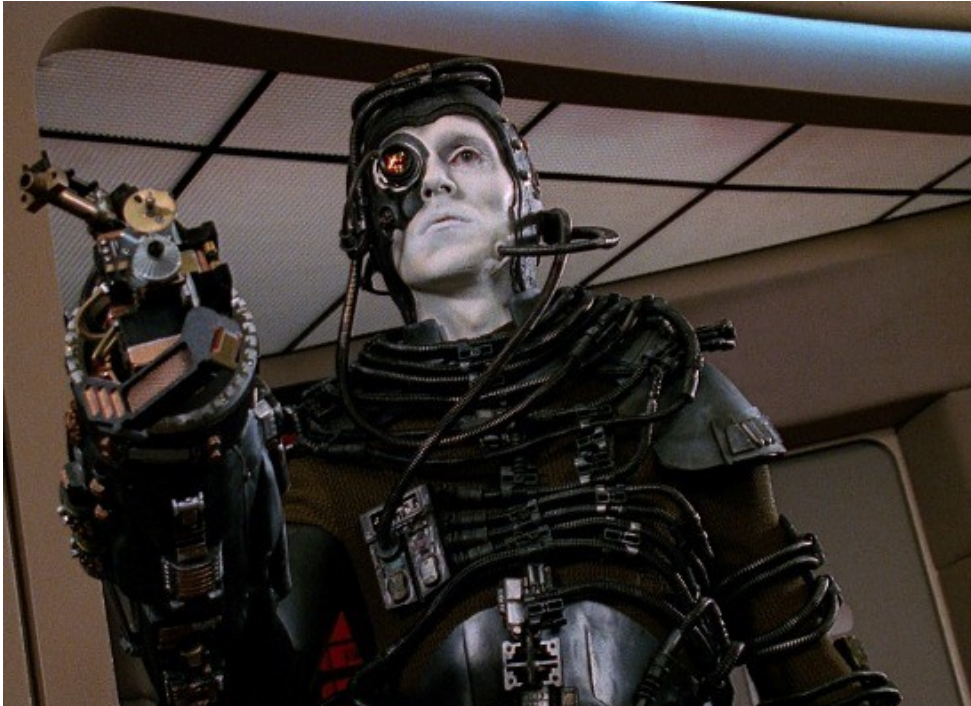
**Architekt?**

# Plan architekta



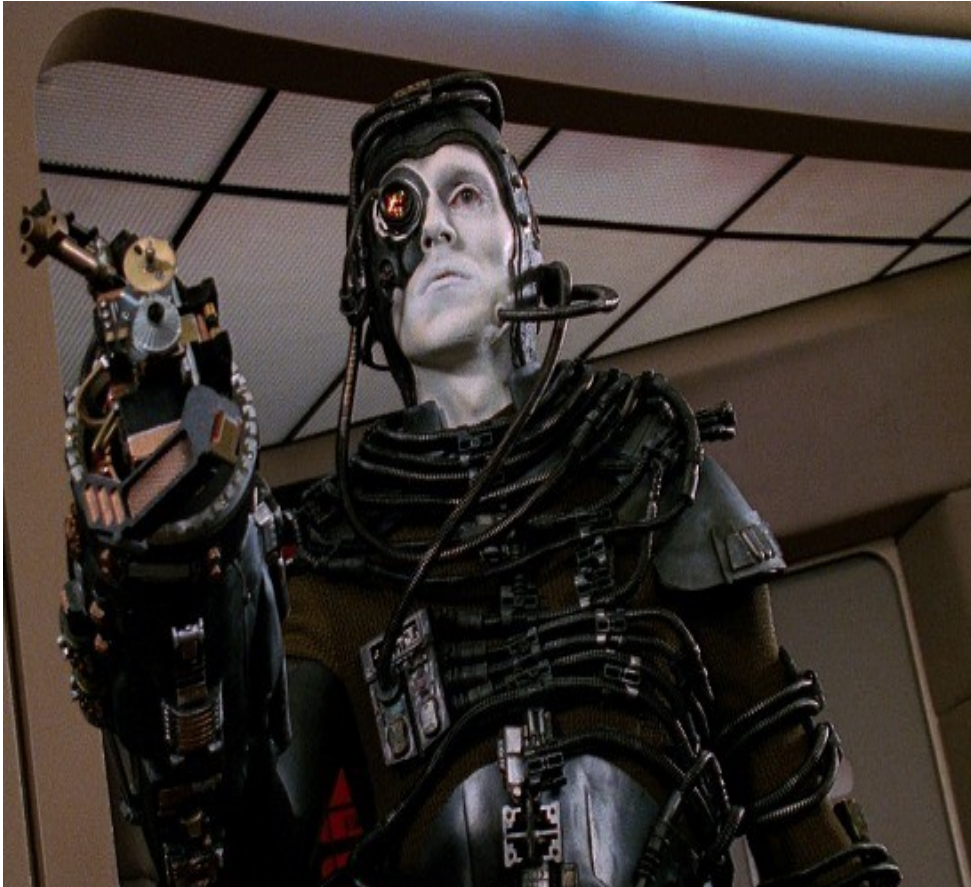
- **Architekt będzie architektorował**

# Plan architekta

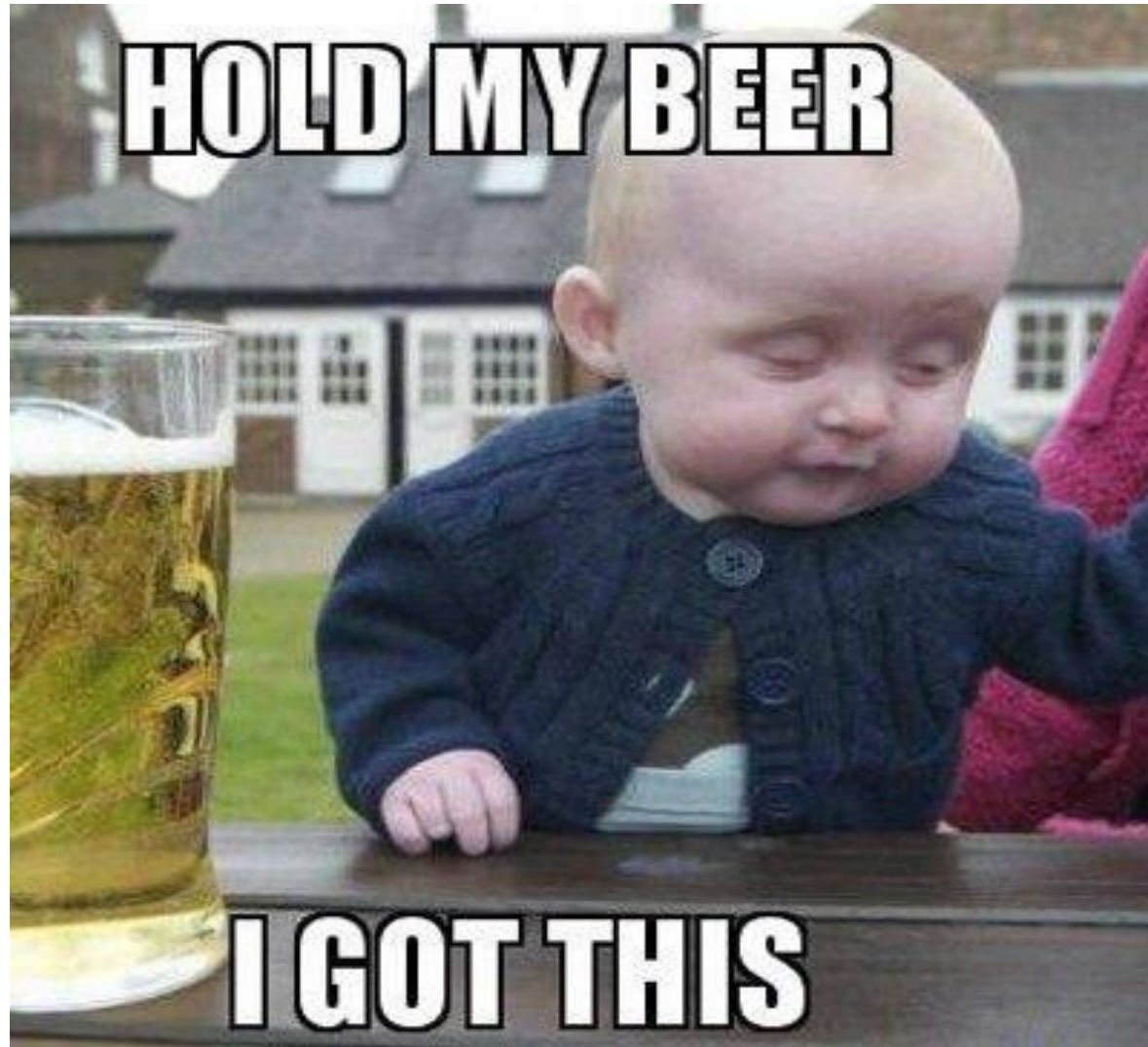


- **Developerzy będą kodzić.**
- **I tak biznes się będzie kręcił.**
- **Przy założeniu że biznes nie wie co zakładać.**
- **A Krzysiu do utrzymania projektu.**

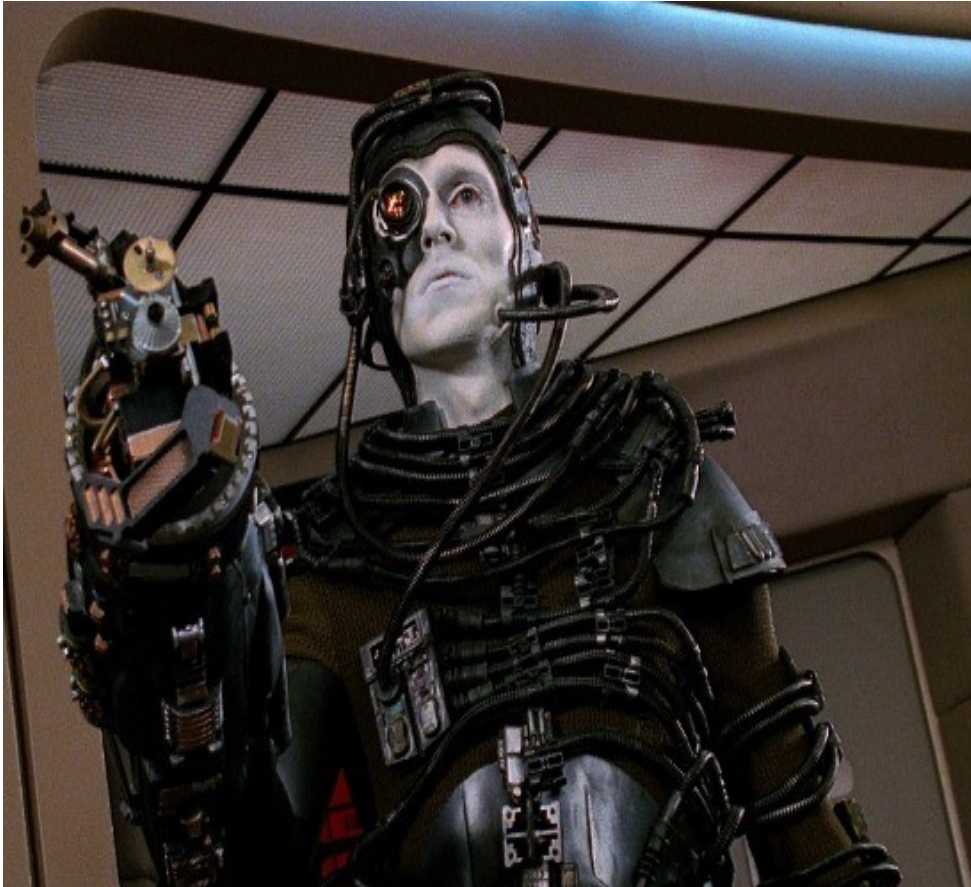
# Plan architekta



- Zenon z zespołem koduje **nową wersję** systemu.
- Za 3 miesiące **wielki deploy!**
- Będą **mikroserwisy!**



# Modus Operandi Zenona



- mikroserwisy za **wszelką cenę.**
- Tak nakazuje **HYPE!**
- **Hype driven developemnt.**

Dawaj Zenon te  
**mikroserwisy!**

Zagrajmy w

„Zenon mówi  
a Krzysiu pyta”

**Zenon mówi:**

Dzielimy monolit na mikroserwisy wg „ścieżki” w REST API.

np. /order to order.service

## Krzysiu pyta:

Nie lepiej poszukać bottlenecków w obecnej aplikacji i wydzielać tylko części, które muszą działać szybciej by sprostać wymaganiom?

A jak podzielić dane zależne między serwisami?

Może konteksty są zbyt mocno powiązane?

## Zenon mówi:

Mamy **Restowe API** to żeby nie tracić czasu będziemy go używać do komunikacji między serwisami.

Krzysiu pyta:

Czy zdajesz sobie sprawę z tego że każde wywołanie po HTTP to przynajmniej dodatkowa 1 sekunda by uzyskać odpowiedź?

## Zenon mówi:

Baza danych. Może zostać jedna. Po co tracić czas na migracje bazy dla każdego serwisu.

Krzysiu pyta:

A co z migracjami bazy danych? Czy one nie spowodują dead-locków przy deployu nowych wersji mikroserwisów?

Krzysiu pyta:

Czy używanie wewnętrznych Id w serwisach spoza kontekstu nie narusza zasady eventual consistency?

**Krzysiu pyta:**

A co z bottleneckami?  
Co z slow queries?

**Zenon mówi:**  
Dodamy indexy.

**Krzysiu pyta:**  
Co z slow insertami?

**Zenon mówi:**  
Usuniemy indexy.

Tyle  
pytań.  
A tak mało  
odpowiedzi.

3 **Miesiące** minęły.

Dzień sądu.

**Deploy!**

**Niespodziewanie**  
pojawiły się błędy

Tylko Zenon **nie pomyślał** jak umożliwić **debug** problemu.

A **Błąd** tak jak w **PHP**: zgłasza **klient**,  
a nie

**kompilator.**

**CEO** i **Inwestor** zaczynają groźnie **marszczyć** brwi.

Projekt fizycznie **traci kasę**.

A zespołowi zaczyna doskwierać brak zastosowania

# Observability patterns

## Krzysiu pyta:

Gdzie są metryki? - Nie wiemy co i ile się wykonuje.

Gdzie zapisujemy traces? - Nie wiemy co i skąd  
przychodzi.

Czy mamy agregowane logi? - Nie widzimy kiedy system  
działa źle.

Gdzie jest retncja błędów? - Wyjątki nie są rejestrowane.

**Debug jest niemożliwy.**

Developerzy muszą zgadywać.

**Krzysiu pyta:**

**Czy dodaliśmy correlation id do każdego requestu, by  
móc prześledzić drogę między wywołaniami  
mikroserwisów?**

Jakoś się **udało**. Po wielu **nieprzespanych**  
**nocach**. Niezliczonej liczbie kaw.

**Ustabilizowali**  
system.

Życie toczy się dalej...

Pojawiają się nowe problemy

**Ruch** jest coraz **większy**.

**Dobrobyt.** Nowi klienci.  
Inwestycja zaczyna się spłacać.

Jednak system mógłby obsłużyć **więcej** klientów.

Ale **brak metryk** powoduje,  
że **Zenon** miota się jak **dziecko we mgle**.

Bo system ma **bottlenecki**.

Tylko nikt nie jest w stanie **określić gdzie**.

Nerwowe decyzje Zenona zdradzają, że traci grunt pod nogami.

Zespół się wypala.  
Nikt nie może pójść na urlop.

A biznes chce  
**nowych funkcjonalności.**

Tylko kto ma je zrobić. Skoro cały zespół  
**podpiera produkcje patykami.**

# Krzysiu oznajmia:

To nie są **mikroserwisy**, tylko **dystrybuowany monolit**.

Do inwestora docierają **niepokojące sygnały**.

Traci on kompletnie zaufanie do **Zenona**.

Zenon **znika** w odmętach LinkedIn.



# Morał:

**Nie potrzebujesz** architektury mikroservisowej,  
aż do momentu kiedy **świadomie** nie zaczniesz jej  
**potrzebować**.

# Wnioski

Nie masz czasu zrobić tego

**dobrze?**

To czy masz czas żeby zrobić to

**dwa razy?**

Skalowanie biznesu i mikroserwisy to  
**nie są synonimy**

Mikroserwisy zwiększają **złożoność** projektu

**wielomianowo.**

Wraz ze **złożonością**  
wzrasta **koszt** wdrażania  
nowych funkcjonalności w projekcie.

Wydzielaj mikroserysy wg  
**domeny biznesowej.**

**Zdefiniuj wąskie gardło,**  
wydziel **jeden** mikroserwis. Zobacz czy się sprawdzi.

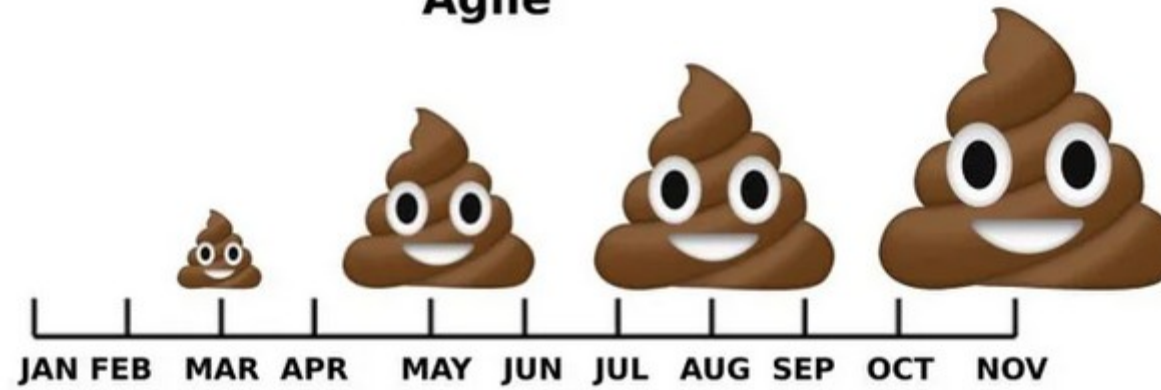
# Waterfallowy

deploy to nigdy nie jest dobry pomysł.

## Waterfall

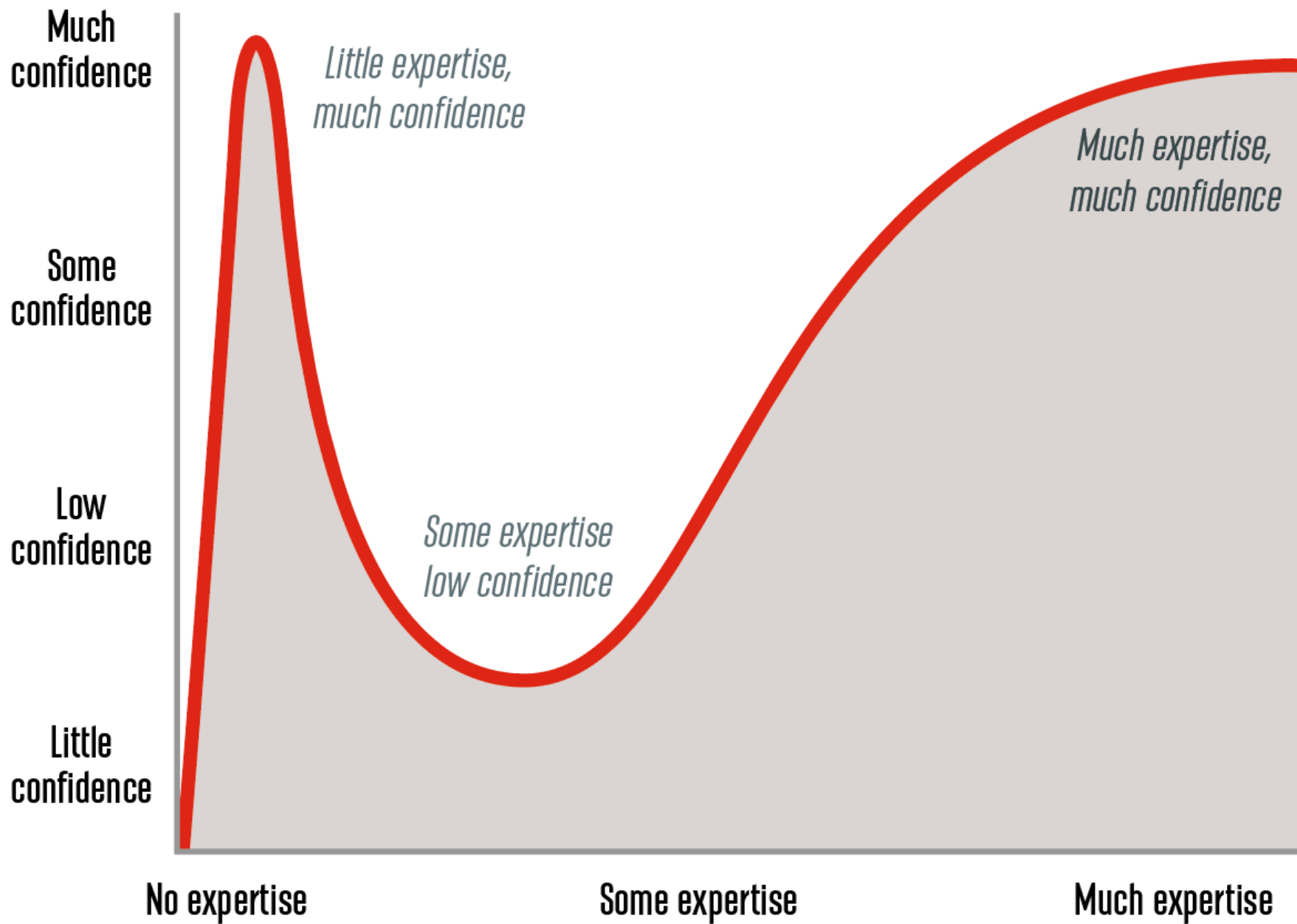


## Agile



A **Zenon** padł ofiarą fenomenu

**Dunninga-Krugera.**



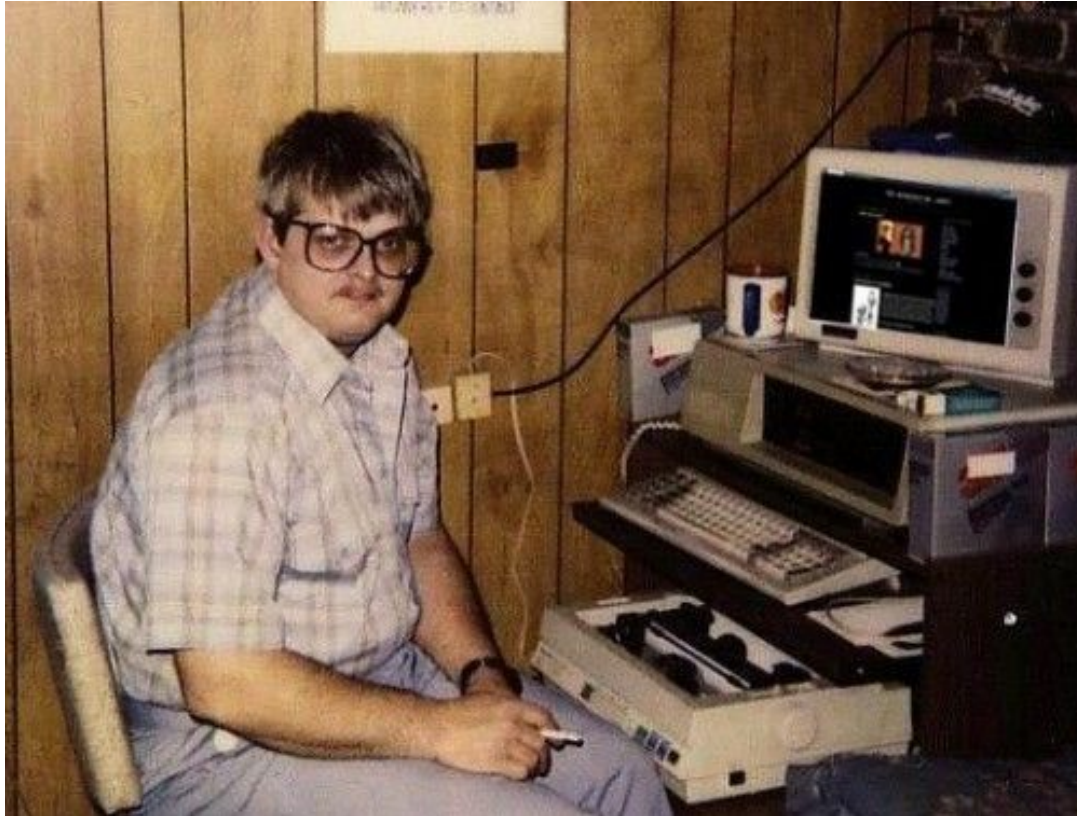
# Do obejżenia później:

# Prezentacje o architekturze mikroservisowej

<https://www.youtube.com/watch?v=kb-m2fasdDY>  
<https://www.youtube.com/watch?v=hnpzNAPiC0E>  
<https://www.youtube.com/watch?v=4FJufP8zwgg>  
<https://www.youtube.com/watch?v=PzEox3szeRc>  
<https://www.youtube.com/watch?v=E8-e-3fRHBw>  
<https://www.youtube.com/watch?v=NP189MPfR7Q>  
<https://www.youtube.com/watch?v=GBTdnfD6s5Q>  
<https://www.youtube.com/watch?v=PFQnNFe27kU>  
<https://www.youtube.com/watch?v=dQw4w9WgXcQ>

A co u Krzysia?

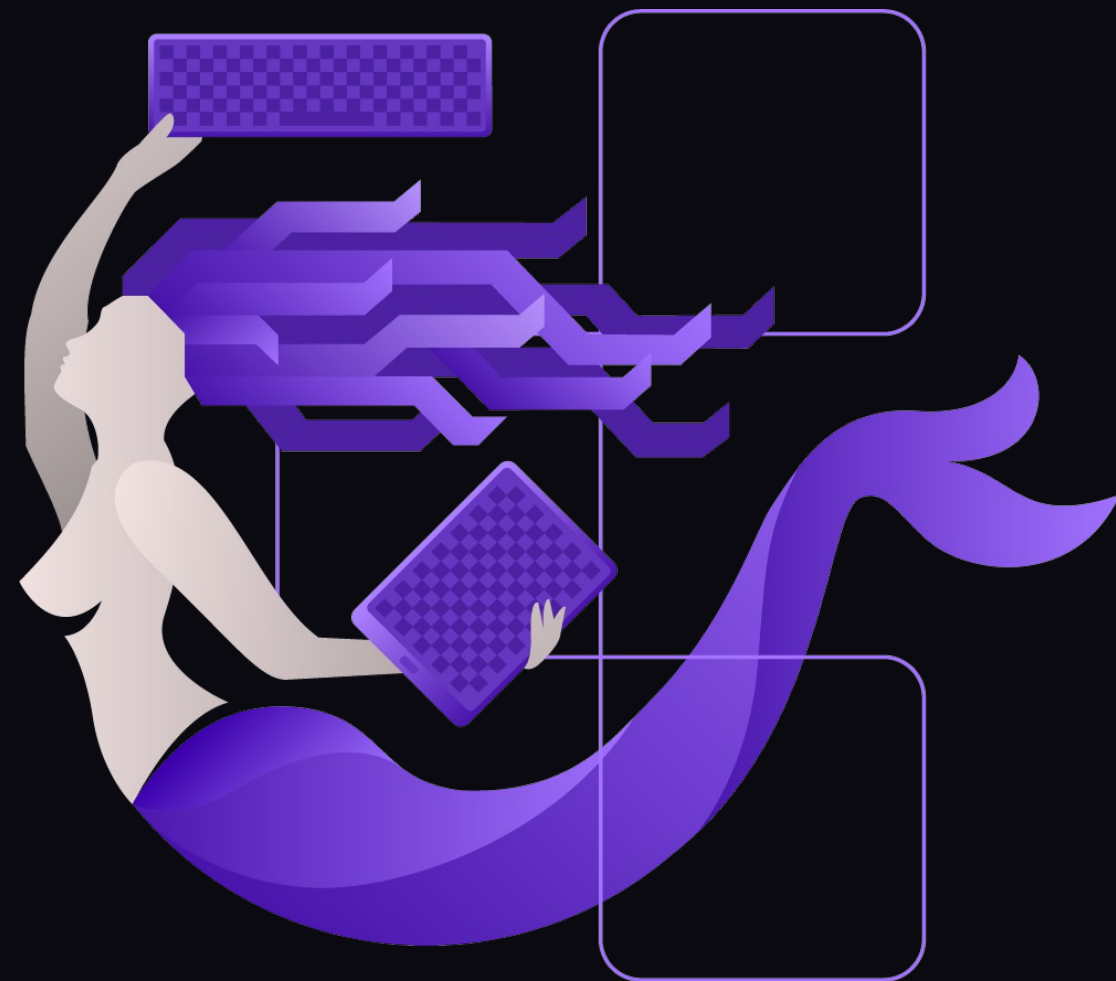
# Krzysiu



- Krzysiu znów jest Lead developerem
- Tylko jemu zależało na dobru projektu
- Ma pracy na kolejne lata by posprzątać po Zenonie

## Thank you for watching!

Remember to rate the presentation and leave your questions in the section below.



[www.WarszawskieDniInformatyki.pl](http://www.WarszawskieDniInformatyki.pl)



31 March - 1 April 2023



Politechnika Warszawska + online

MAIN ORGANIZER: **AcademicPartners**  
FUNDACJA

ORGANIZING COMMITTEE: dozens of organizations from the IT / data science sector (full list on the event website)