Microservices, Micropains, Microgains

Bartosz Sokół

@bartsokol

bart-sokol.info

THERE WILL BE MEMES.
A LOT OF MEMES.
imgflip.com

What is this „**microservice**” thing?

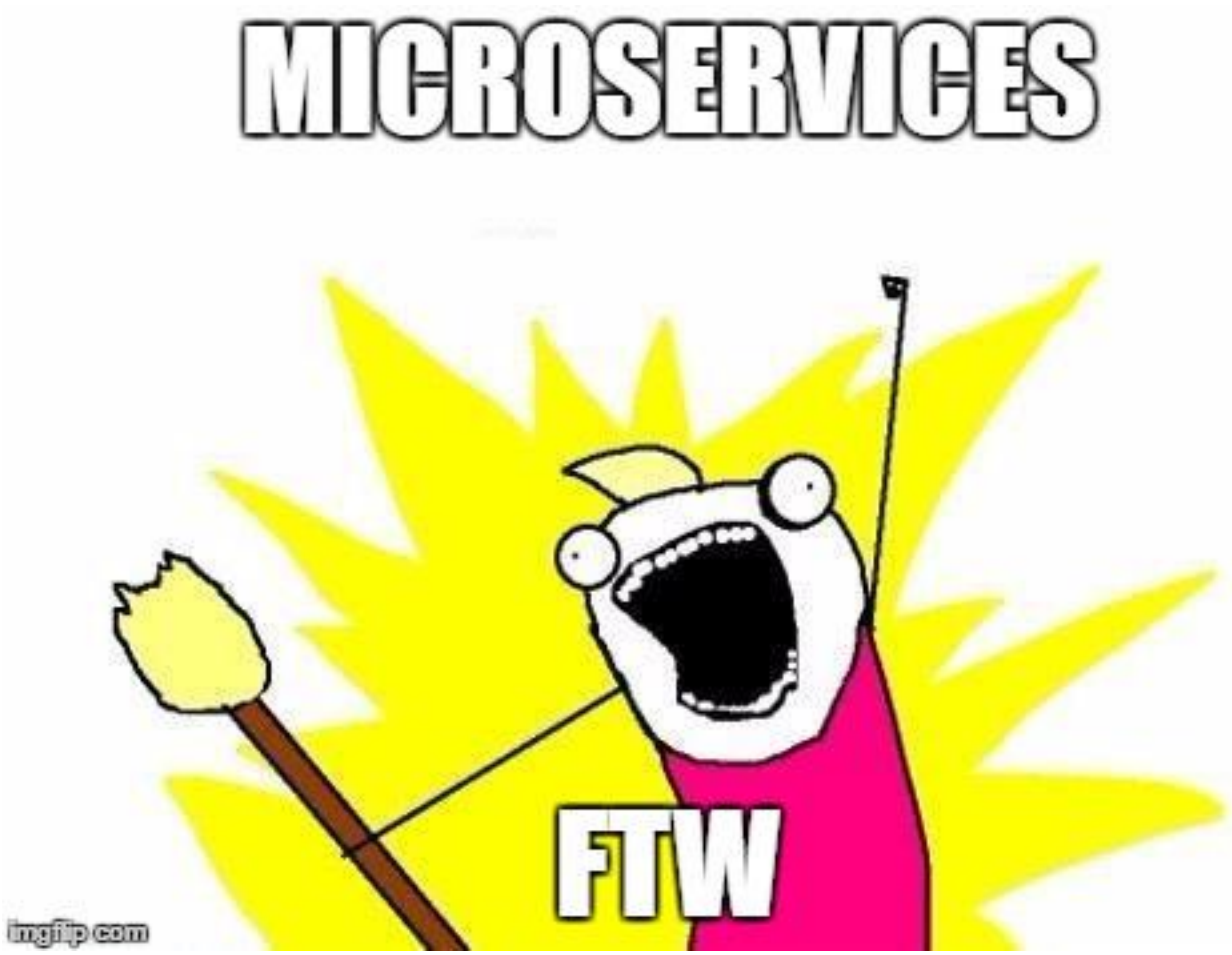
I DON'T ALWAYS CREATE SERVICES
BUT WHEN I DO, I MAKE SURE THEY'RE MICRO
imgflip.com

Microservice

- Micro- *
- -Service **

* It’s small. Like really small. Few hundred LOC? One sprint to develop?

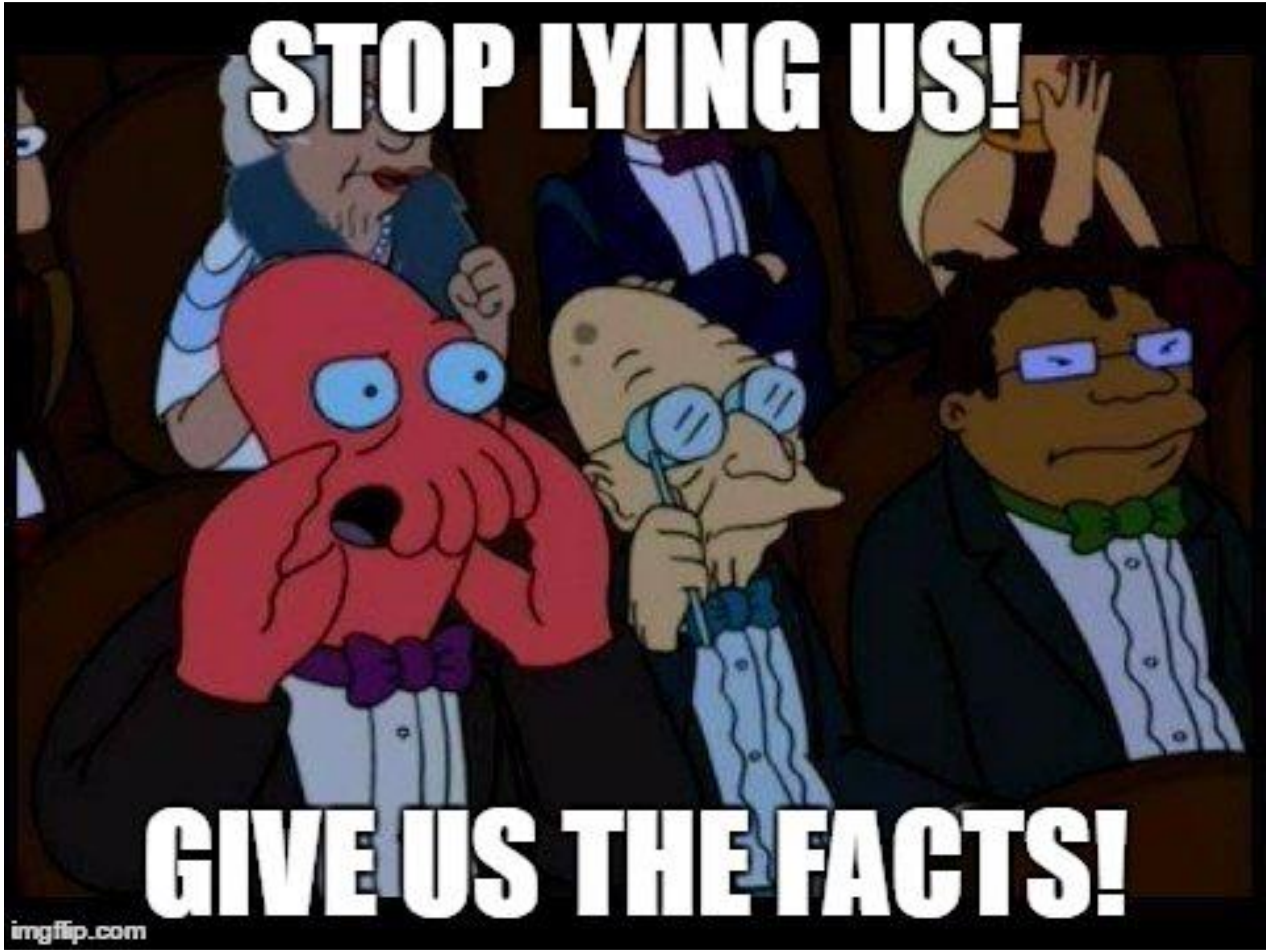
** Serves someone. Does something. Usually one thing. Not much more.


MICROSERVICES
FTW
imgflip.com

Pure facts *

- There are 1,729 gazillion microservices running right now
- Each microservice will bring you $376k of income each year
- They’ll speed up your app by 1385%
- Developer who created microservice will get new job in less than 5 minutes
- You’ll get 178 new followers if you tweet about #microservices right now

* DISCLAIMER: There are no reliable sources for those facts. Some of them may be imaginated. Actually all of them. Don’t trust them unless you’re sure what you’re doing.


STOP LYING US!
GIVE US THE FACTS!
imgflip.com

What microservices should be

- Small components providing one functionality
- Communicating with other services over inter-process protocols (in particular over network)
- Independently deployable
- Fault-tolerant
- Scalable to any numer of instances

SO MANY
SERVICES
HISTORY.COM
imgflip.com

What microservices can be

- Solution for scaling out
- Domain separation provoker
- Continous improvement trigger
- Chance to try different technology

YOU CANNOT JUST
ADD MORE RAM
imgflip.com

What microservices are not

- Cheap and easy thing
- Performance booster
- Solution for any problem
- Easy to create, test and maintain

HE THOUGHT THEY'LL BE
FASTER
imgflip.com

Microservices architecture must-haves

- Solid deployment pipeline (continous delivery) and good DevOps skills (either across the team or via dedicated roles)
- Good monitoring tools (error and performance tracing)
- Good understanding of architecture across the team
- Solid development workflow, including good communication, code reviews, QA proces
- Understanding of network related issues, including availability, performance and security

AND WHAT SHOULD I DO NOW WITH ALL THOSE SERVICES?
DEPLOY
MONITOR
TEST
imgflip.com

Microservices dos

- Prepare your deployment flow: source control, review tools, continous integration, deployment tools, environment monitoring
- Make sure you know what’s going on on your environments
- Make sure you can trace and test interactions between services and swap then on the fly
- Get familiar with different communication protocols to choose the right one for the job
- Version your APIs
- Be cautious about breaking changes in the API and behaviour

ALL THE SERVICES
ARE GREEN NOW
imgflip.com

Microservices don’ts

- Don’t start project with microservices – in most cases they will be only unnecessary cost
- Don’t share database across services
- Don’t share data contract libraries
- Don’t use binary (or similar) serializer unless they support compatibility modes (missing or extra fields)
- Don’t go to big (monolith) or too small (nanoservices)

I SHARED CONTRACTS ONCE
IT WAS AWFUL
imgflip.com

Talking with others

- There are multiple ways in which microservices can communicate
  - HTTP
  - REST
  - Message Queues
  - Databases
  - Files…
- Services can be synchronous or asynchronous
  - Sync ones respond directly to requests
  - Async ones do some action (in response to request or message) when possible and can send response when task is finished (but don’t have to)

HE WANTED TO USE
"REST"
imgflip.com

Common problems with microservices

- Dependent service is down
- Network is down
- Service can be shut down on restarted at any time
- Contract has changed
- Malformed data was received
- State is out of sync
- Do we need to think about security?

WAITING UNTIL
OTHER SERVICE
WILL BE UP
imgflip.com

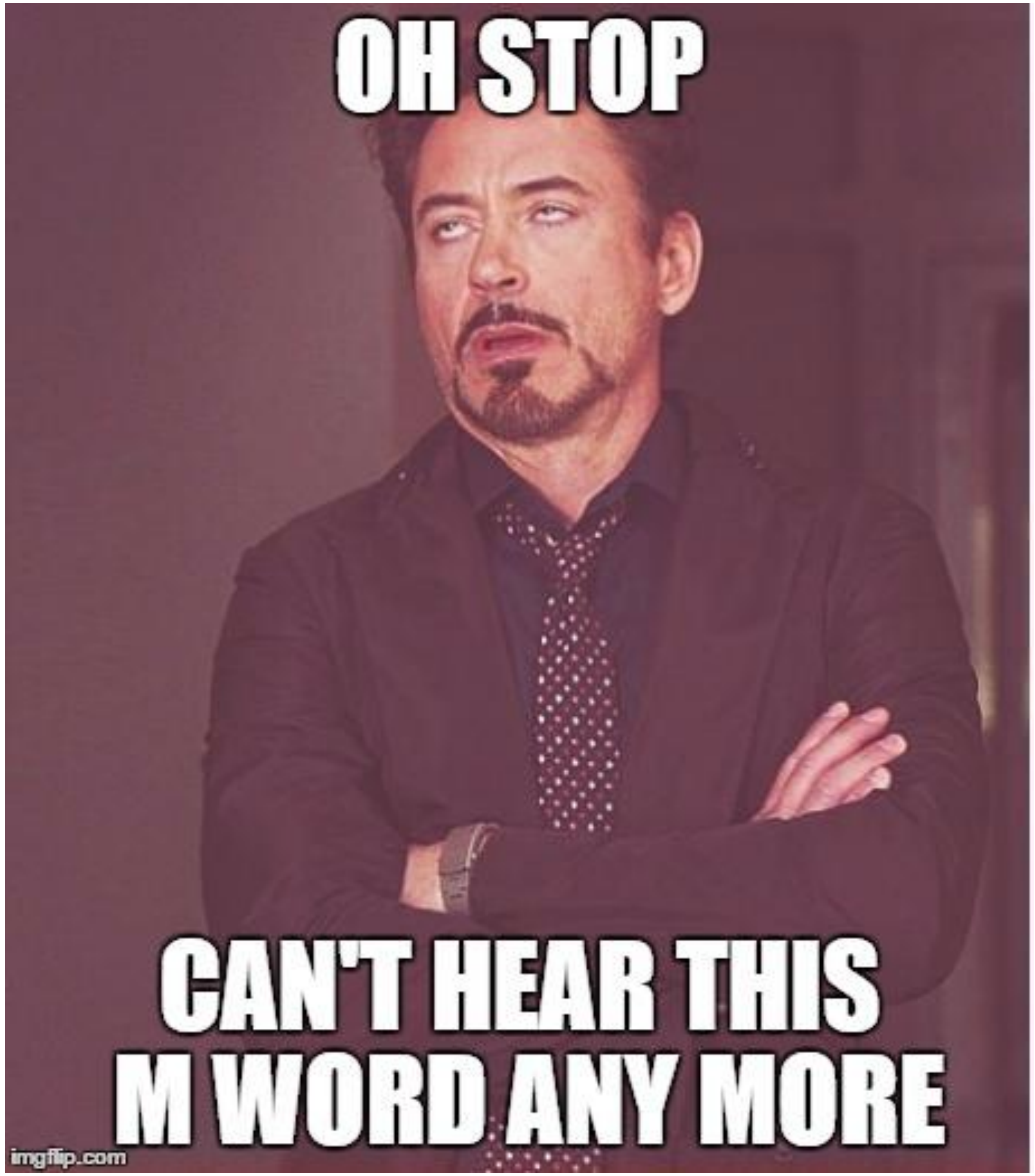
Is it safe?

- Anything exposed on network is vulnerable
- Private networks can help, but only a bit
- Use encryption wherever possible
  - When using HTTP-based communication, use HTTPS/HTTP2
- You can use tokenized requests and responses (e.g. using OAuth) to authenticate callers and authorize access to resources

THEY TOLD ME WE WERE
SAFE
imgflip.com

Not so obvious benefits of microservices

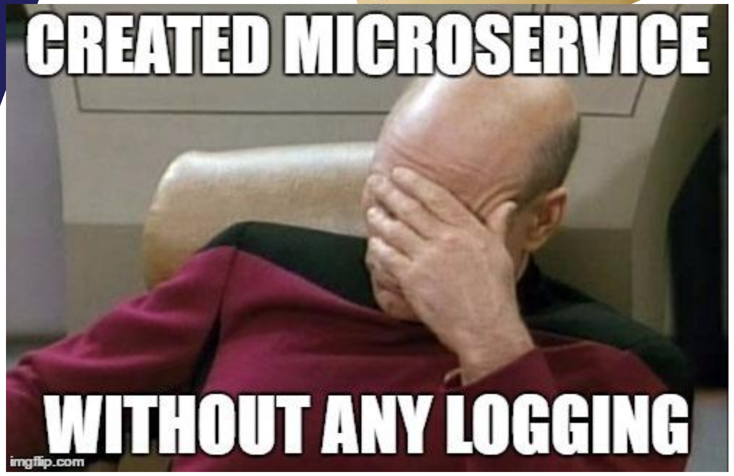
- It’s easier to iteratively improve your code – learn from previous services, try to improve with each new service
- It’s easier to try out new technologies
  - Cross-platform communication and data standards are virtually a must
  - You still have to know how to deploy, monitor and test new tech
- Each completed service is a finished project
  - Each release is reason to celebrate
  - Team happiness level increases
  - People are more keen to stay in the company


OH STOP
CAN'T HEAR THIS
M WORD ANY MORE
imgflip.com

So what are you going to do tomorrow?

I'LL CREATE MY OWN SERVICE
WITH BLACKJACK AND HOOKERS
imgflip.com

Q&A



@bartsokol

bart-sokol.info

Thank you!

Bartosz Sokół

@bartsokol

bart-sokol.info