

Microservices potřebují Event Sourcing



Antonín Stoklásek

12/11/2018

Antonín Stoklásek

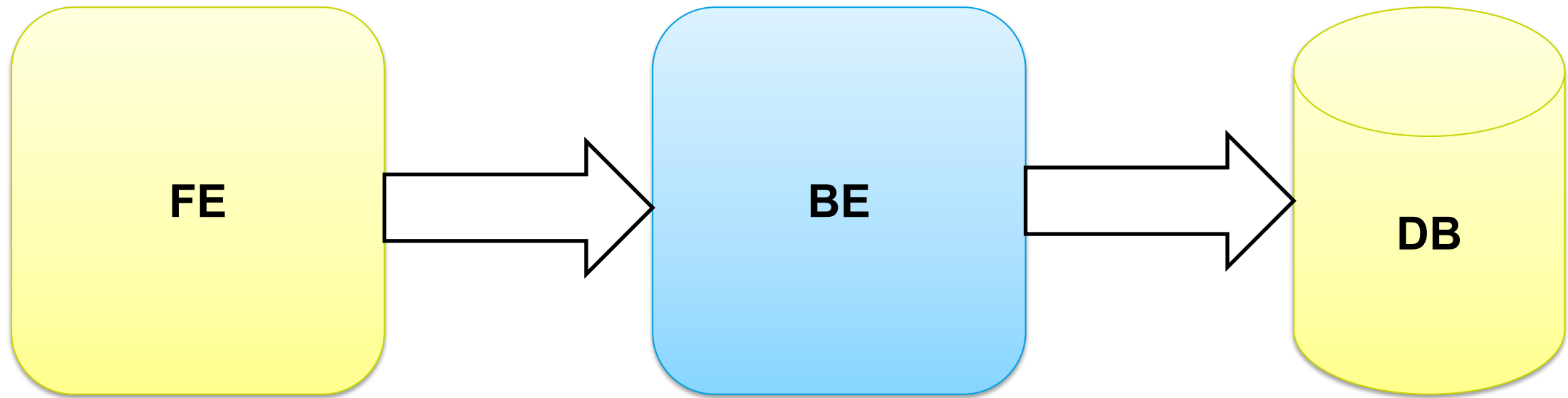
- Technical Team Lead
- Java, Docker, React ...
- Banking Software Company
- Twitter @tonda100
- GitHub tonda100
- <https://tonda100.github.io/>



Agenda

- Traditional Systems
- Microservices
- Message Driven Architecture
- Event Sourcing
- CQRS
- Demo

Traditional System



Monolith

- One technology
- One database
- One repository
- One EAR
- One Team Lead
- Bad!! Maybe??

Monolith Issues

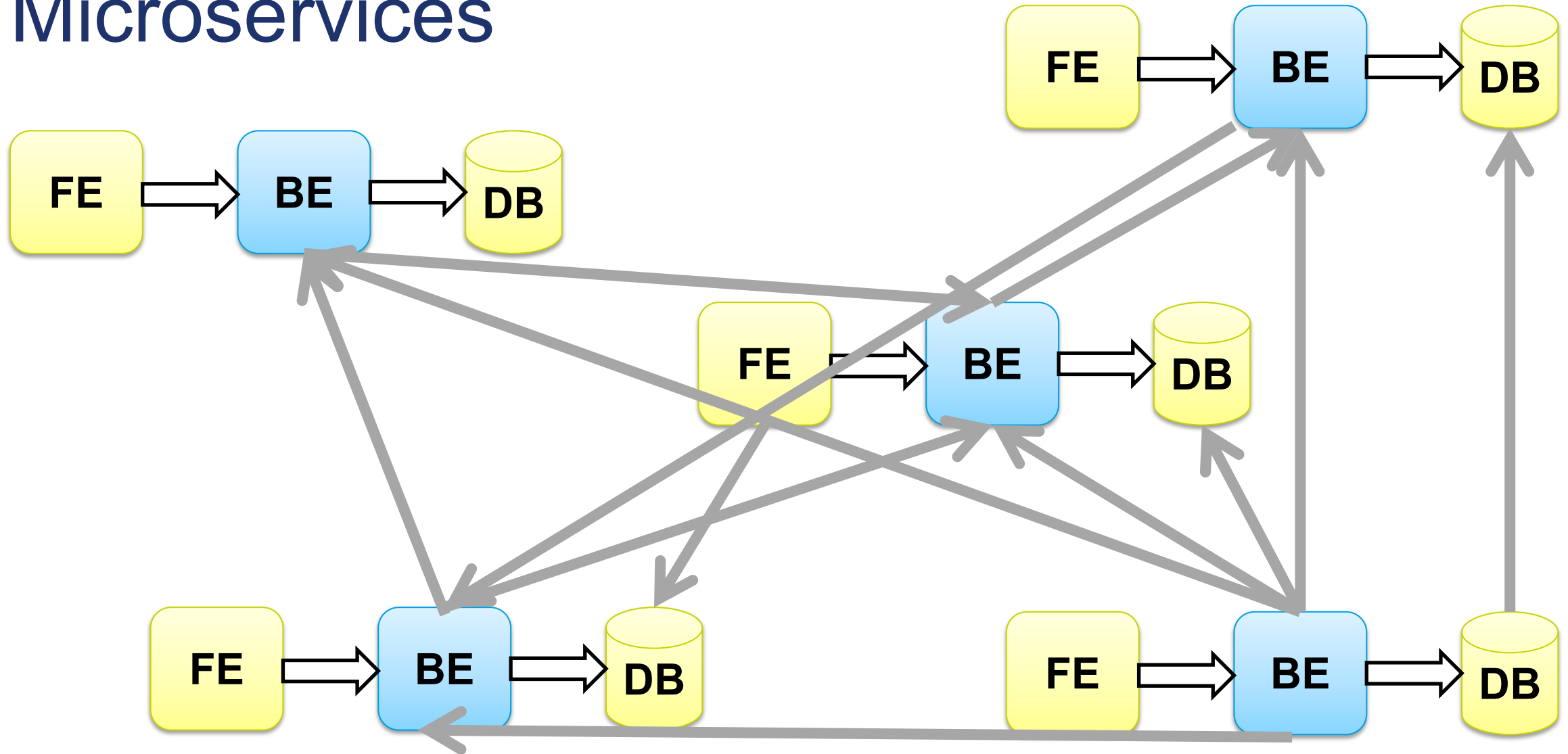
- Technology not suitable for everything
- Database not suitable for every use case
- Getting bigger
- Difficult to maintain
- Difficult to learn
- Nobody knows

Microservices will save us

Microservices

- Single purpose
- Small team (pizza rule)
- Suitable technology
- Suitable storage
- Easy to maintain
- Easy to learn

Microservices

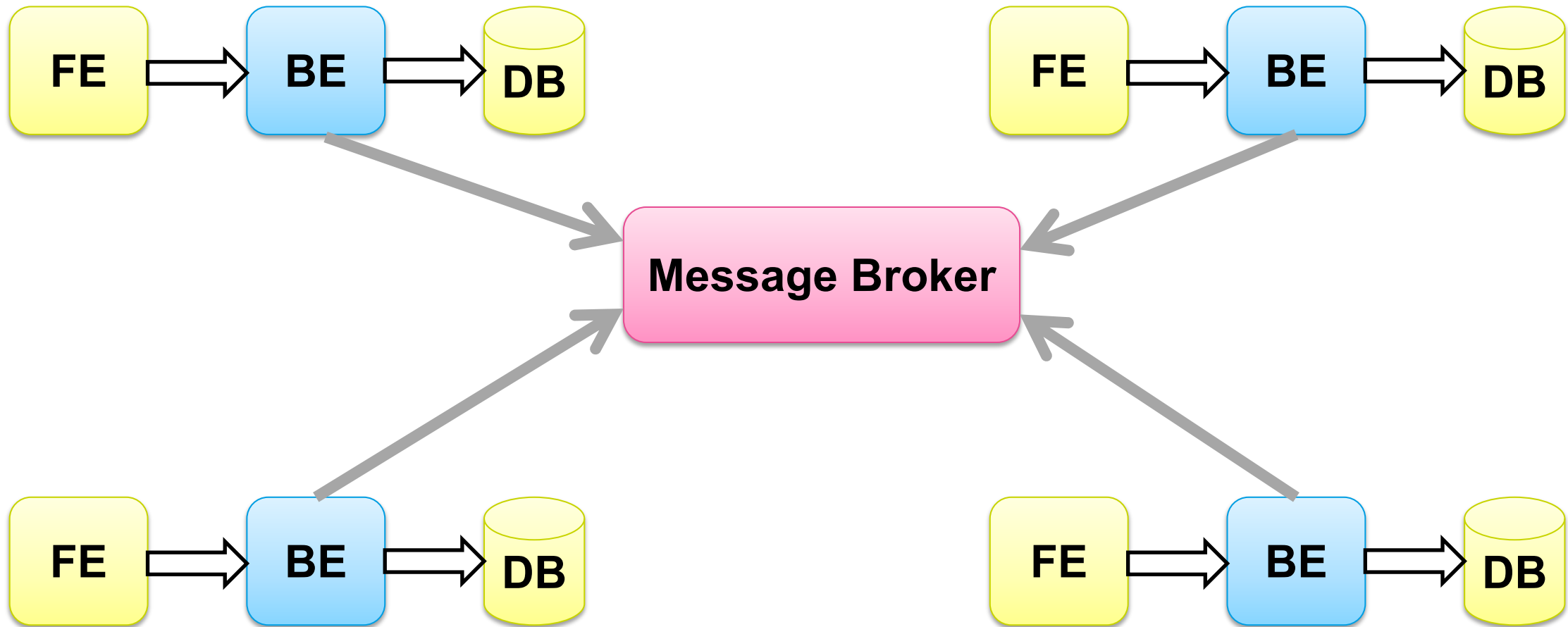


Distributed Monolith is NIGHTMARE

Distributed Monolith

- High coupling & Low cohesion
- Multiple teams
- Single service is useless without others
- One service fails => everything fails
- Synchronized communication
- Distributed transactions

Message Driven Architecture



What should be the message?

EVENT

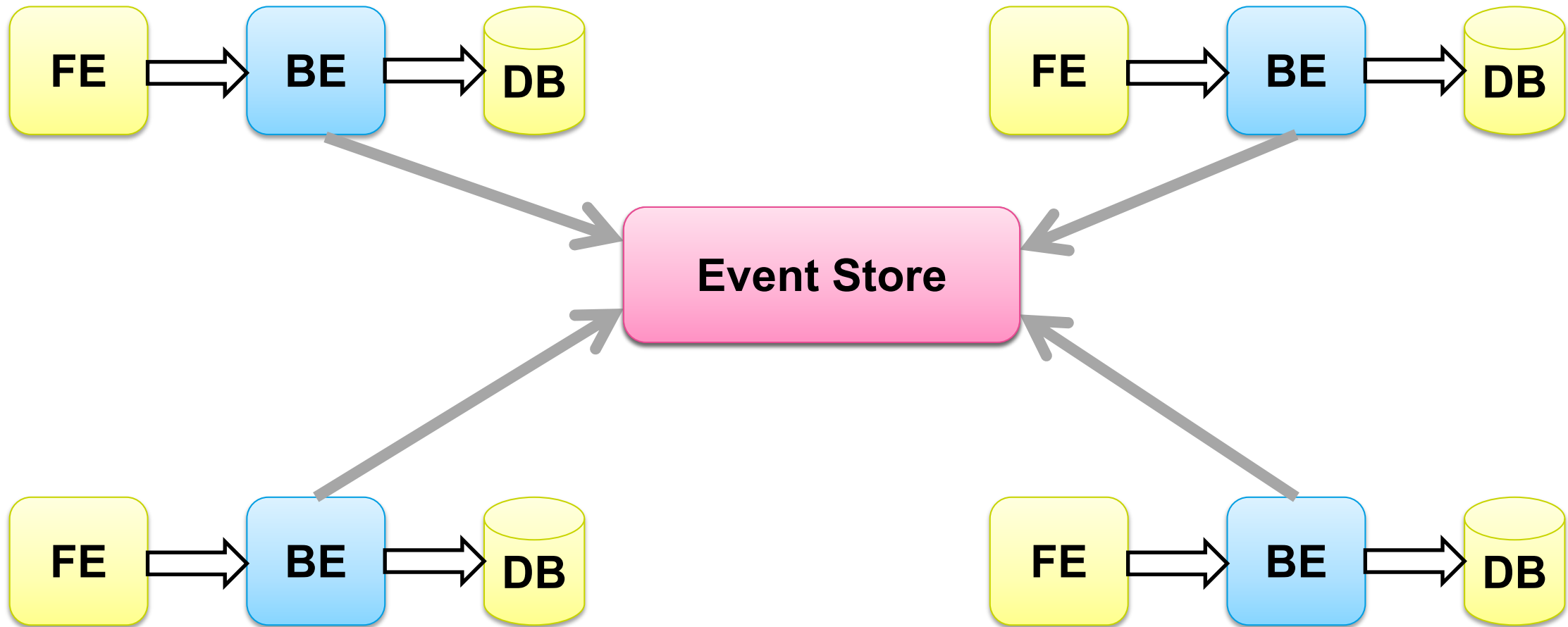
What belongs to Message Broker?

Everything

What about new Microservices?

Events stored forever

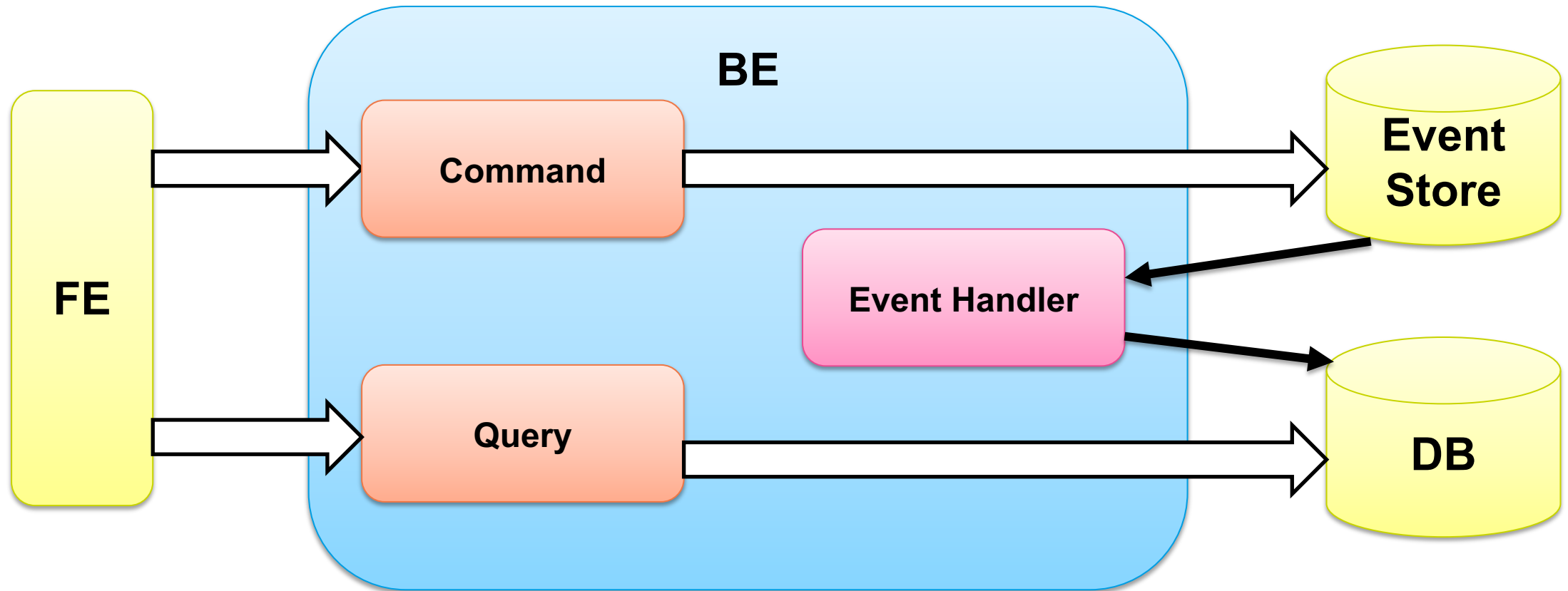
Event Sourcing



Event Store

- Source of truth
- Immutable
- Topics
- Publishing events
- Subscribing
- Guaranteed event order

CQRS



Event Sourcing

- Event matters => state is irrelevant
- Audit log out of the box
- Eventual Consistency
- Performance
- Real world abstraction
- Not loosing information

Event Sourcing – dark side

- Disk size
- Lack of experience
- Immutable
- Eventual consistency
- Event upgrading

DEMO

Summary

- Event matters => state is irrelevant
- No information is lost
- Eventual consistency is not bad
- Performance – no locks
- Suitable read models
- Never do distributed transactions

Děkuji za pozornost.

www.JavaDays.cz

www.gopas.cz

