MICROSERVICES

A COMPLETE PICTURE

Oliver Sturm • @olivers • oliver@oliversturm.com

DevExpress®

Microsoft®
MVP
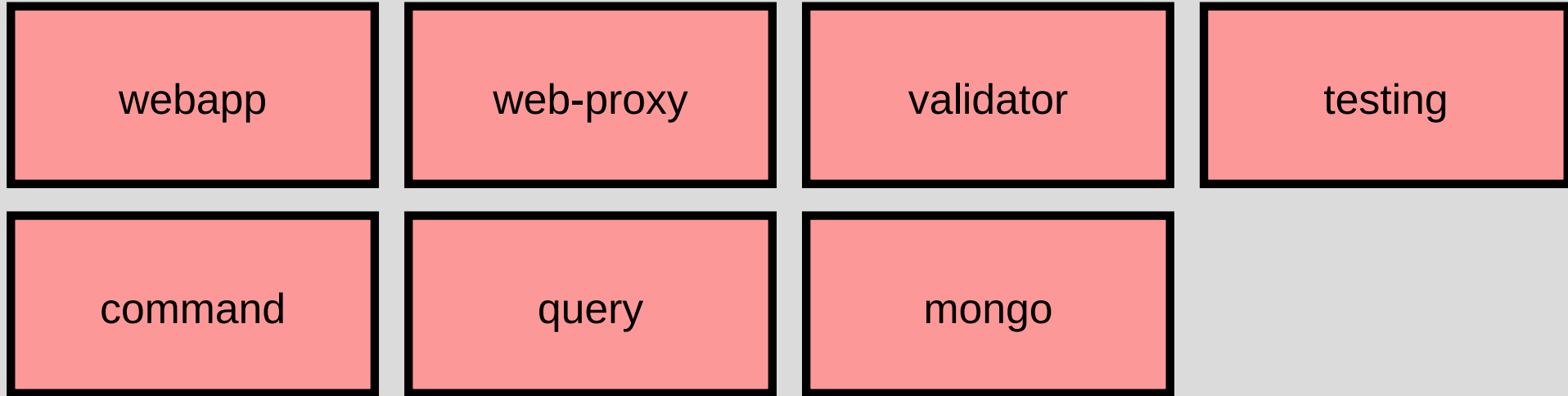Most Valuable
Professional

OLIVER STURM

- Training Director at DevExpress

- Consultant, trainer, author, software architect and developer for over 25 years

- Microsoft C# MVP

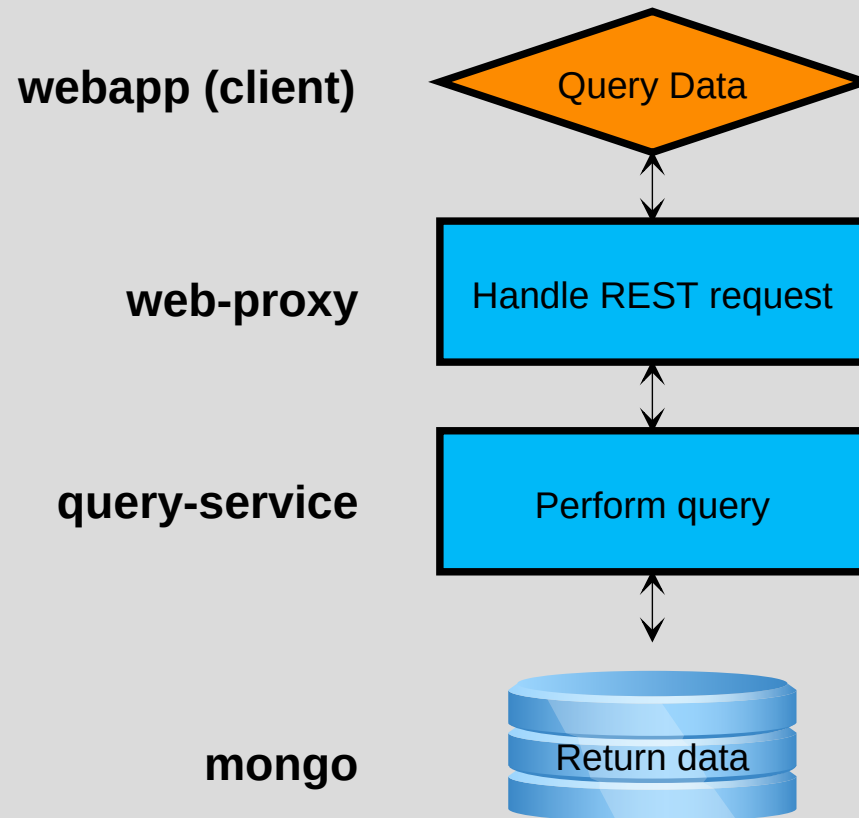- Contact: oliver@oliversturm.com

AGENDA

- Service structure
    - A look at a microservices architecture
- Communication
    - Considerations pro and con frameworks
    - Working with individual services
- Packaging/deployment
    - Developer concerns
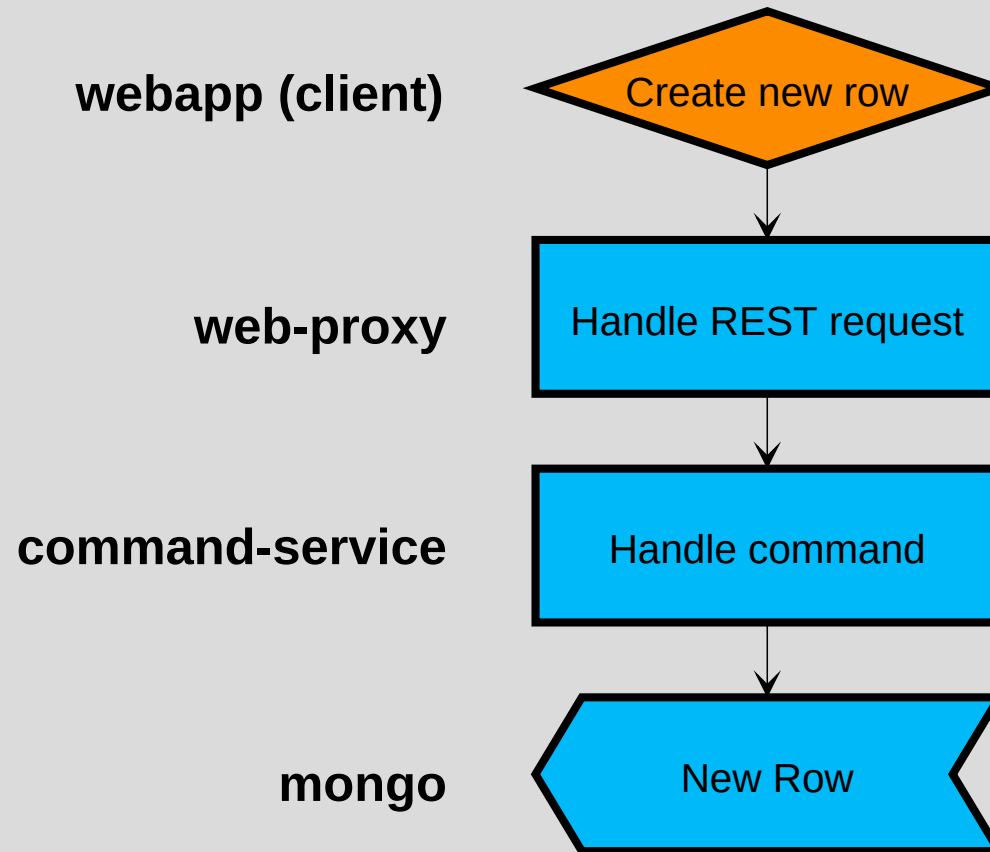   - Real-world deployment with AWS
- Developer stuff
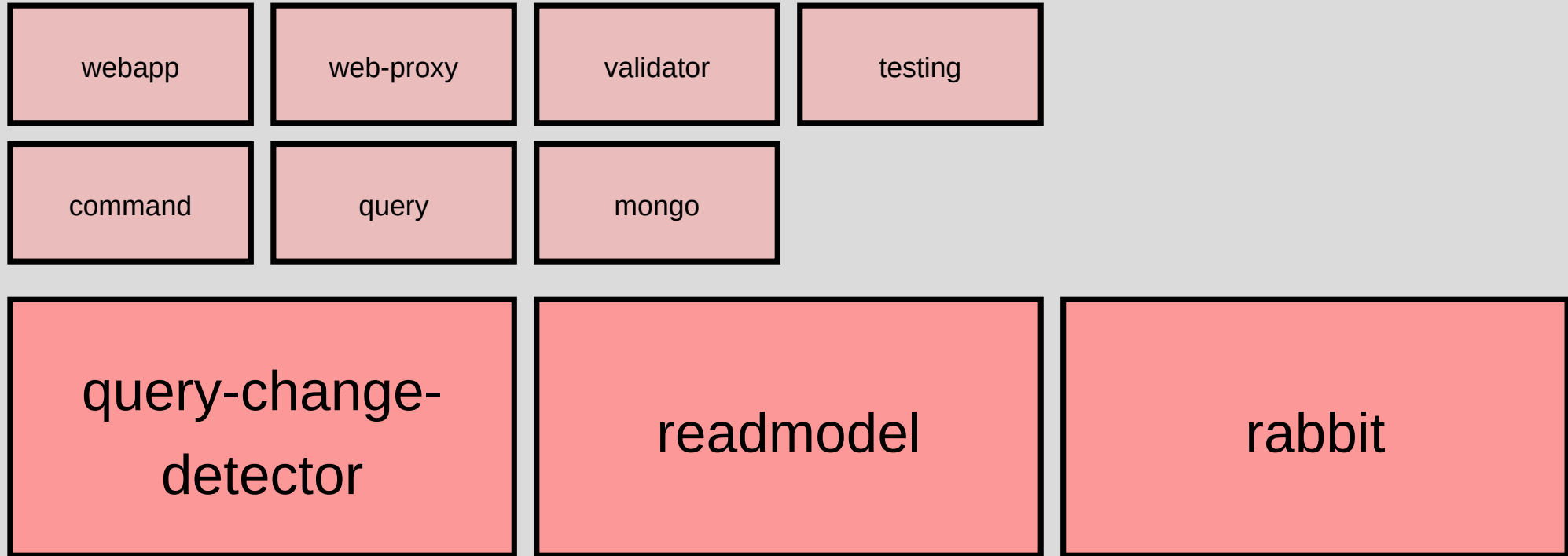   - Debugging

Seven Services of the Demo Application

webapp

web-proxy

validator

testing

command

query

mongo

QUERYING DATA

webapp (client) — Query Data

web-proxy — Handle REST request

query-service — Perform query

mongo — Return data

CREATING A NEW ROW

webapp (client) Create new row

web-proxy Handle REST request

command-service Handle command

mongo New Row

ADDITIONAL SERVICES FOR ADVANCED ARCHITECTURE

webapp	web-proxy	validator	testing

command	query	mongo

query-change-detector	readmodel	rabbit

QUERYING DATA WITH CQRS/ES
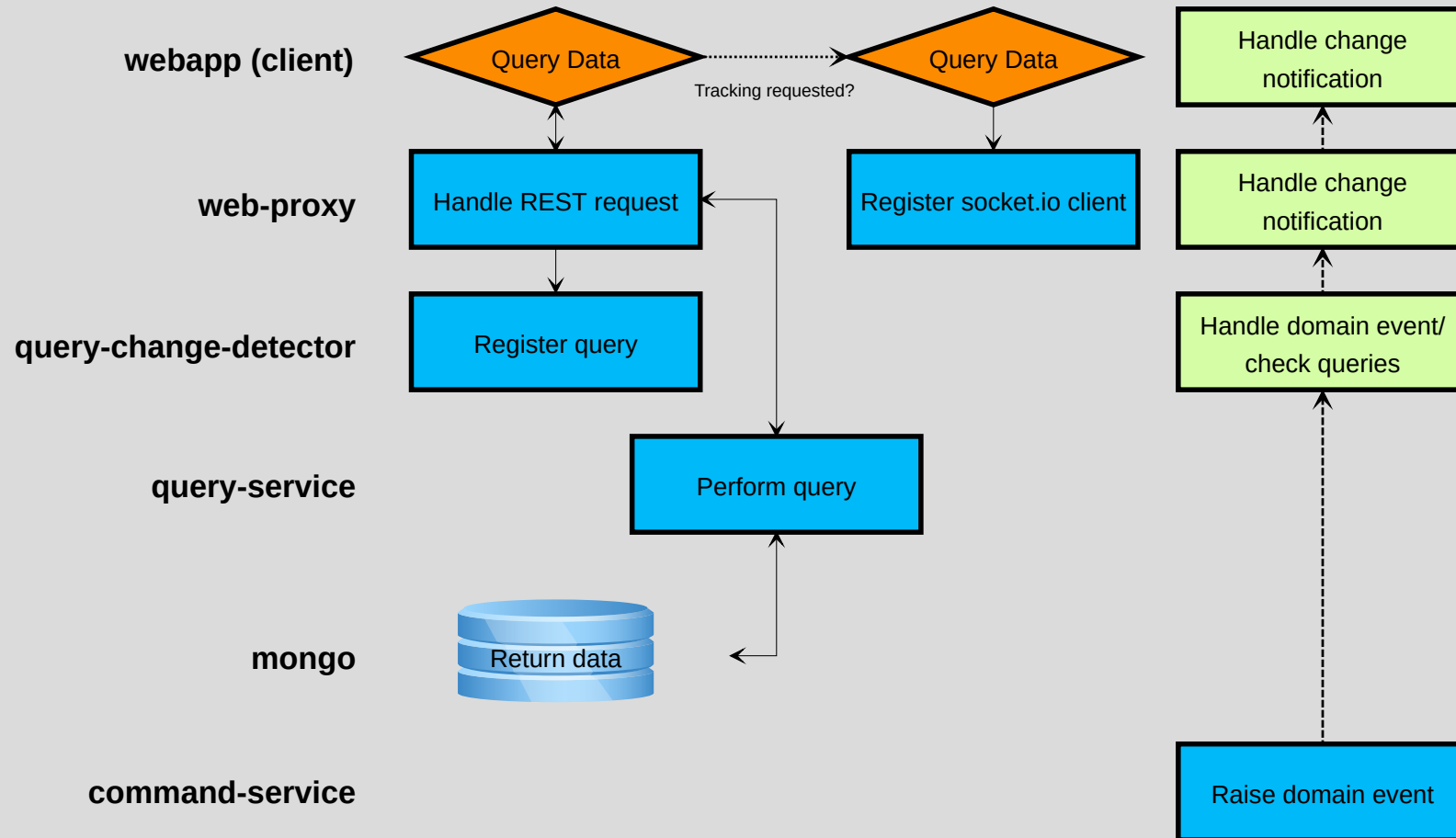
CREATING A NEW ROW WITH CQRS/ES

webapp (client) Create new row

web-proxy Handle REST request

command-service Handle command

readmodel Handle domain event

query-change-detector Handle domain event

mongo New Row

Handle change notification

Handle change notification
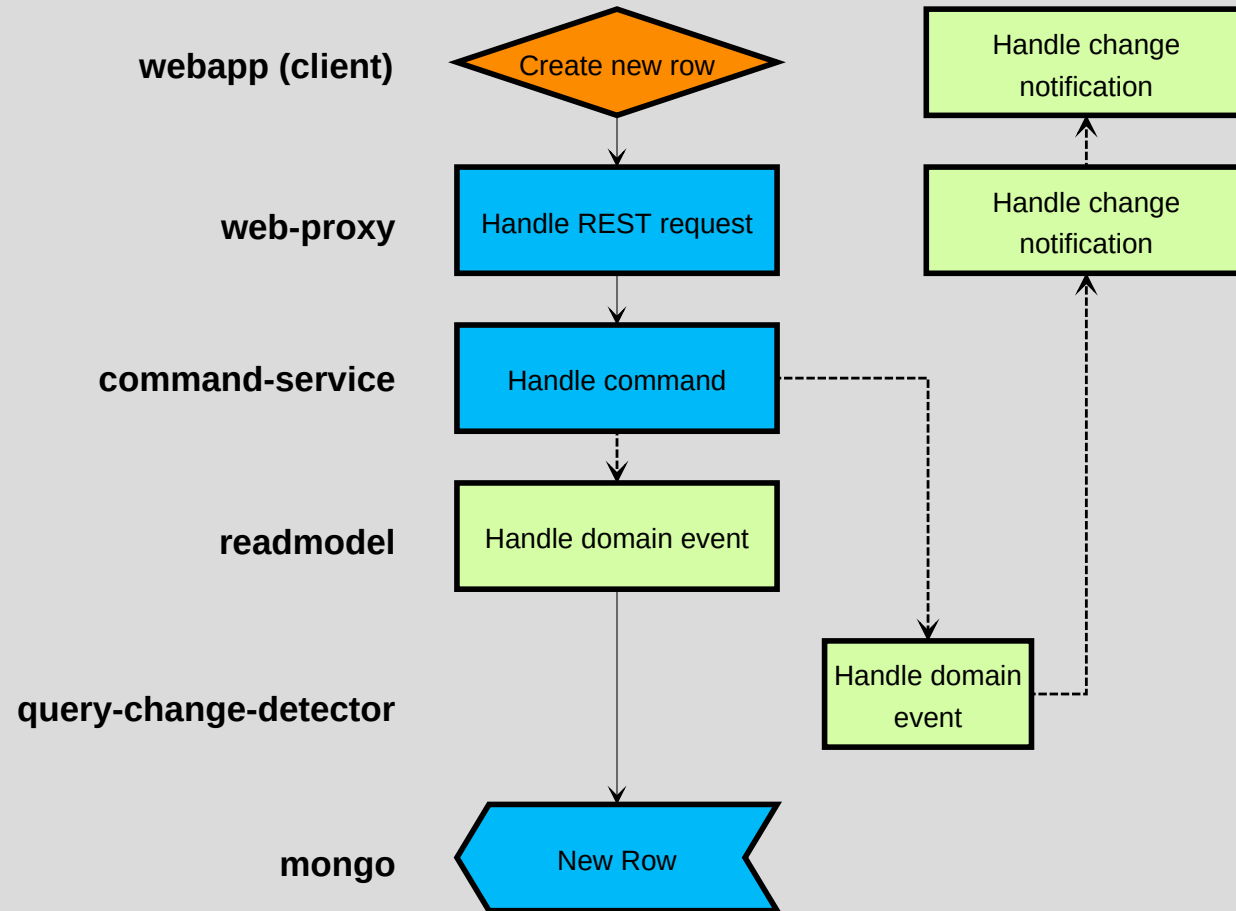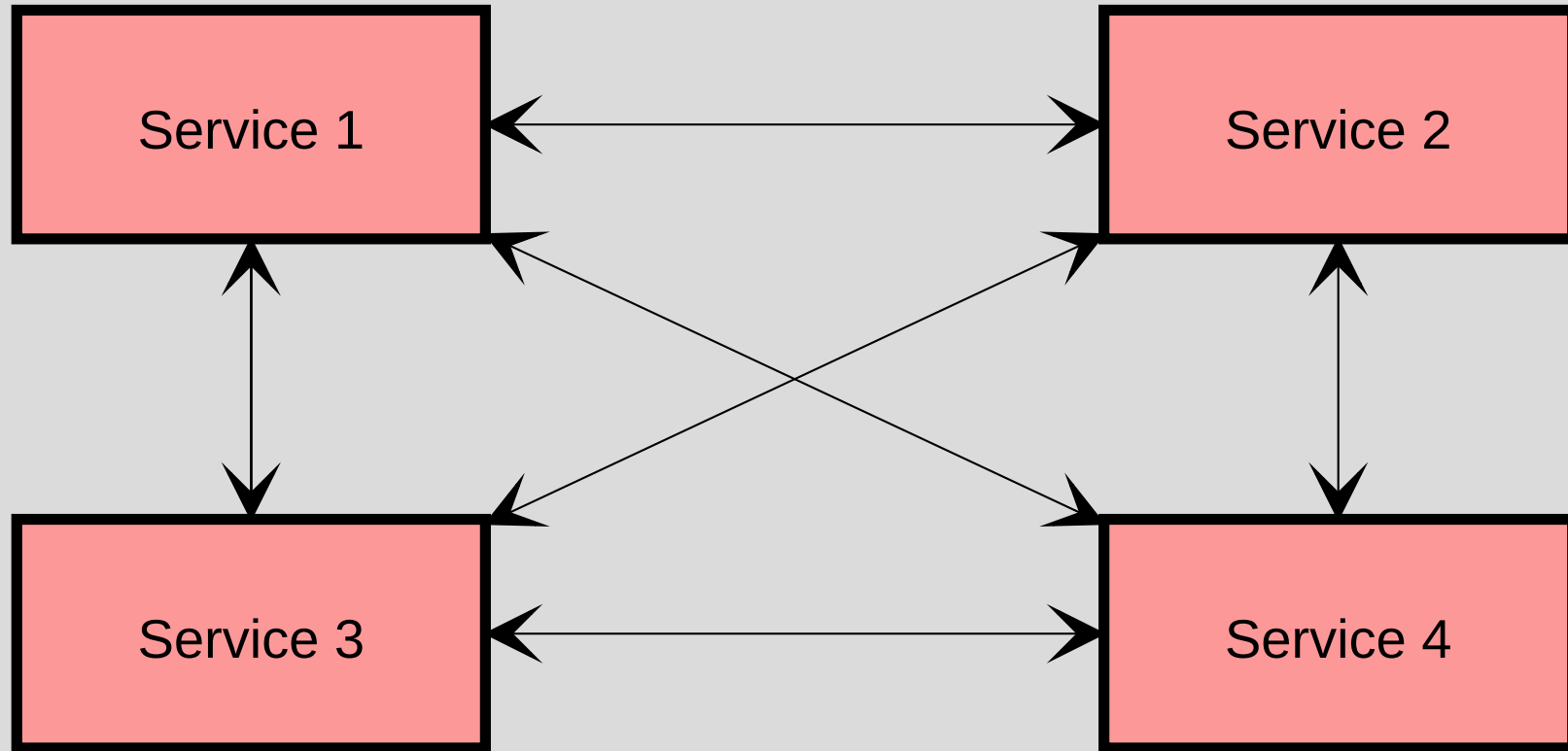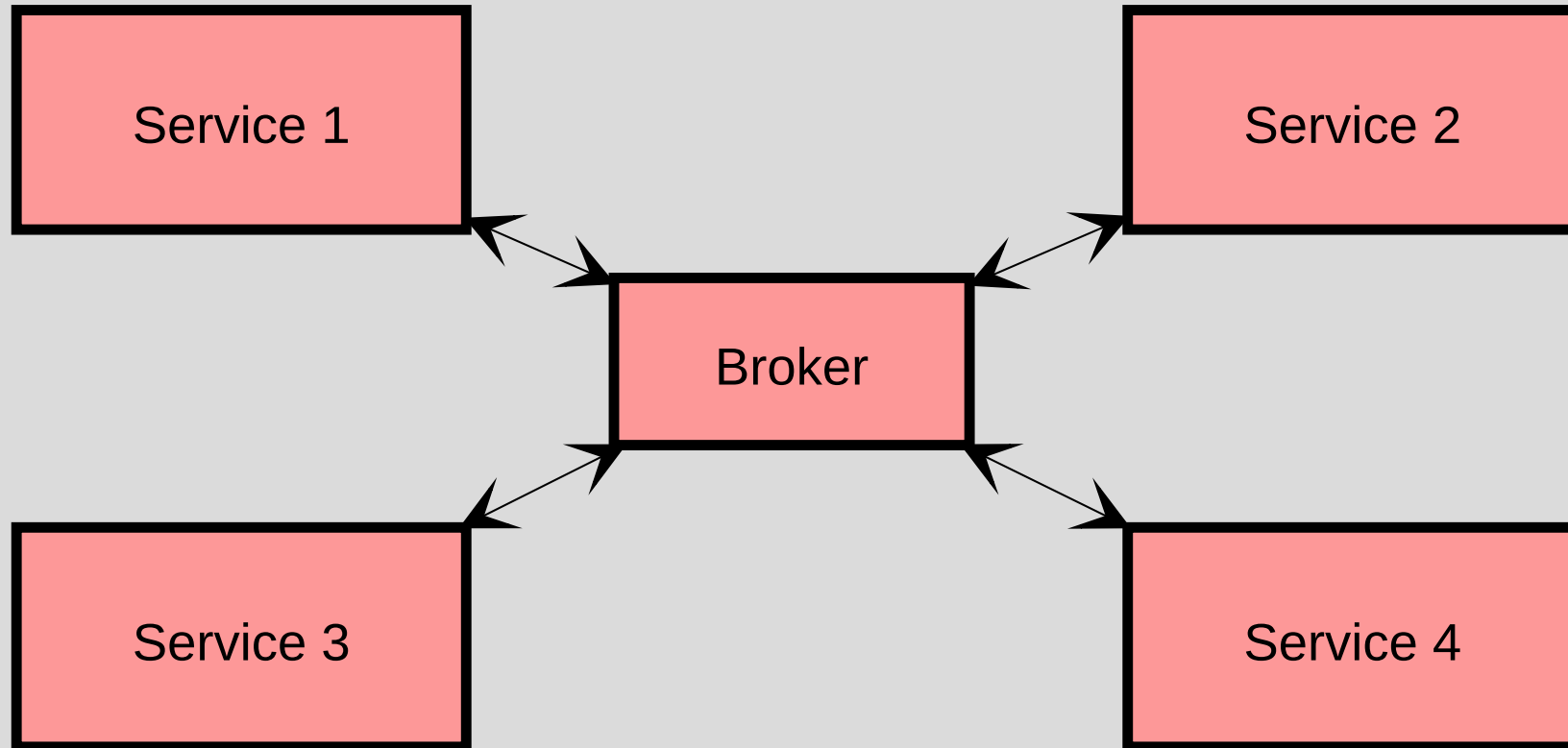
COMMUNICATION

- Structural question: *who talks to who*?

- Implementation question: *how does the talking work*?

DIRECT COMMUNICATION

Using a Broker

HOW DOES THE TALKING WORK?

- Each service could be an *open web service* with its own external interface. REST? Proprietary? Your choice.

- Each service could be implemented to *talk to the broker exclusively*

- *Libraries* exist that implement communication

Packaging/deployment

- Running lots of services manually isn't much fun
  - Consider *automation*

- Services may need *individual runtime environments*

- *Container systems* to the rescue!

DEBUGGING

- *Granularity* of services makes it easy to test
- Services can be debugged as *individual autonomous entities*
  - Best regards from functional programming!

Sources

- This presentation:

    - https://oliversturm.github.io/microservices-complete-picture

    - PDF download:
      https://oliversturm.github.io/microservices-complete-picture/slides.pdf

- Demo code:

    - https://github.com/oliversturm/cqrs-grid-demo (check *master* and *event-sourcing* branches)

Thank You

Please feel free to contact me about the content anytime.

Oliver Sturm • @olivers • oliver@oliversturm.com