Microservices

A Complete Picture

Oliver Sturm • @olivers • oliver@oliversturm.com

Oliver Sturm

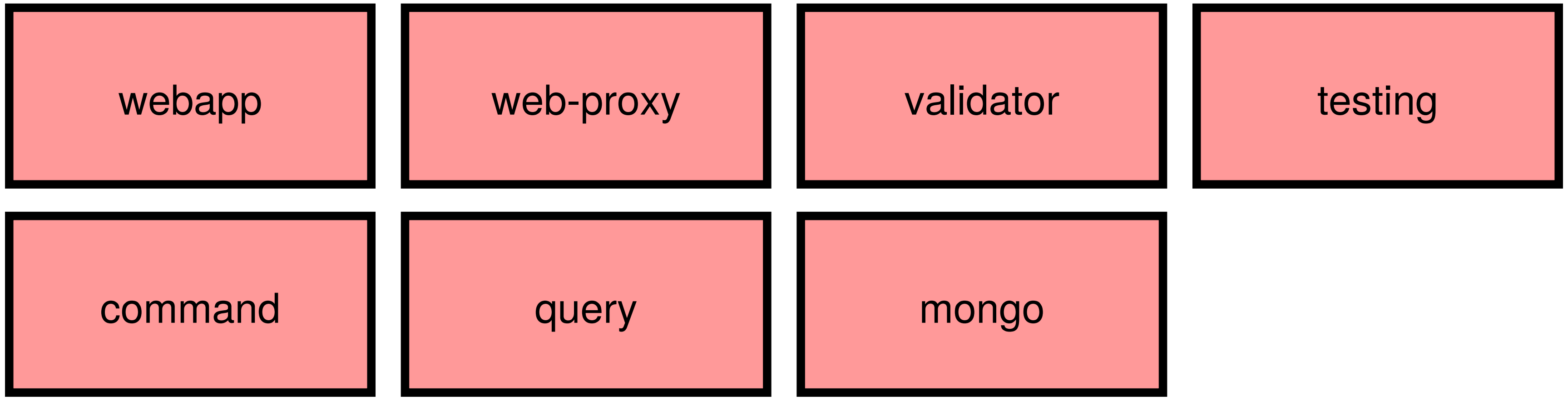
- Training Director at DevExpress
- Consultant, trainer, author, software architect and developer for over 25 years
- Microsoft C# MVP
- Contact: oliver@oliversturm.com

Agenda

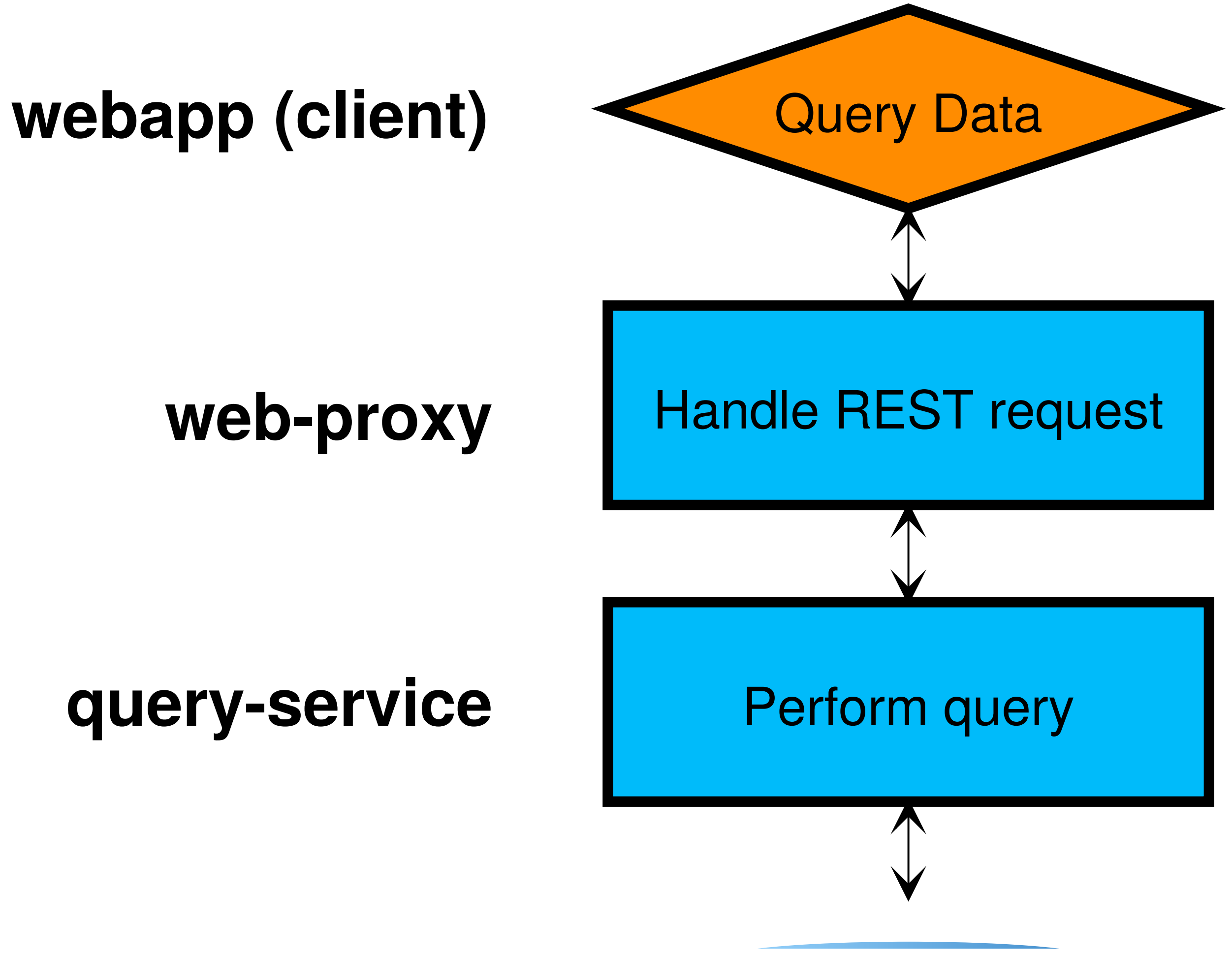
- Service structure
  - A look at a microservices architecture
- Communication
  - Considerations pro and con frameworks
  - Working with individual services
- Packaging/deployment
  - Developer concerns
  - Real-world deployment with AWS
- Developer stuff
  - Debugging

Service structure

My demo application system has at least seven services:

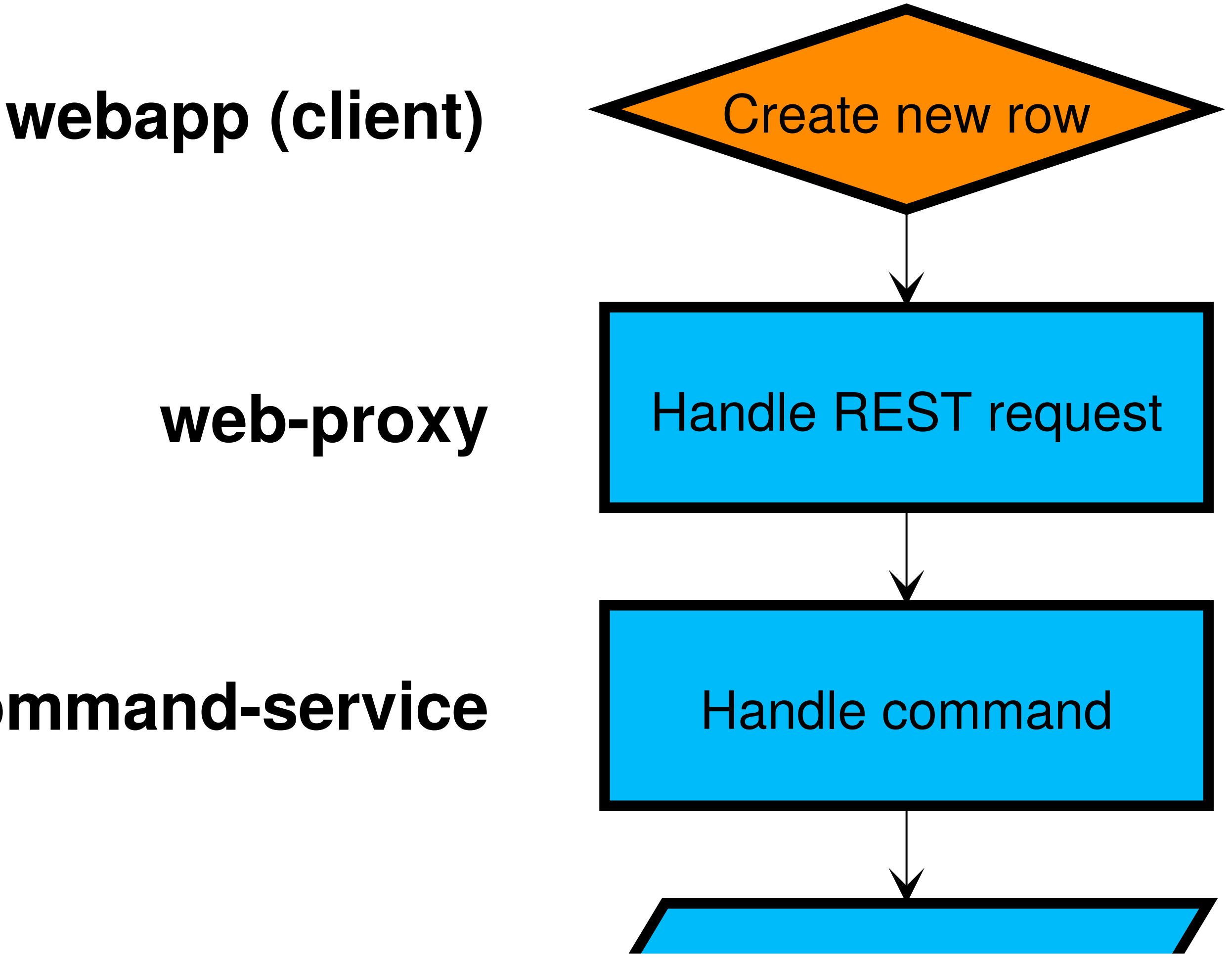


Querying data



mongo

Creating a new row

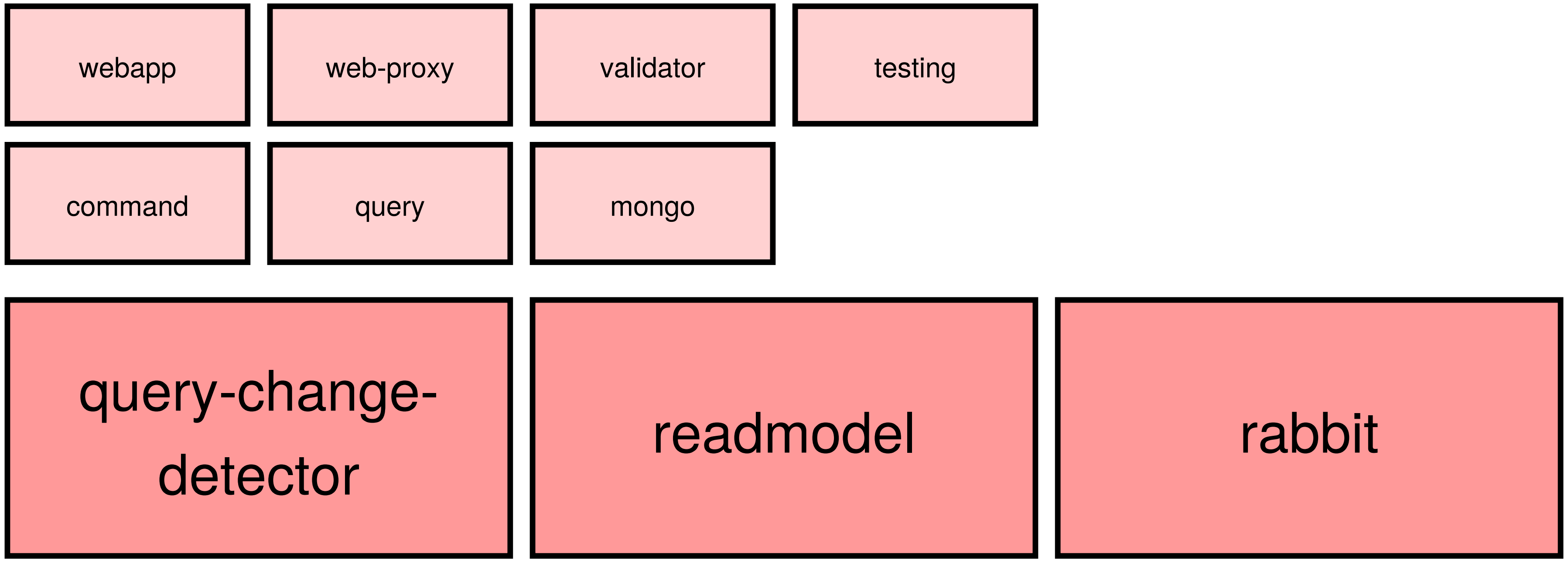


mongo

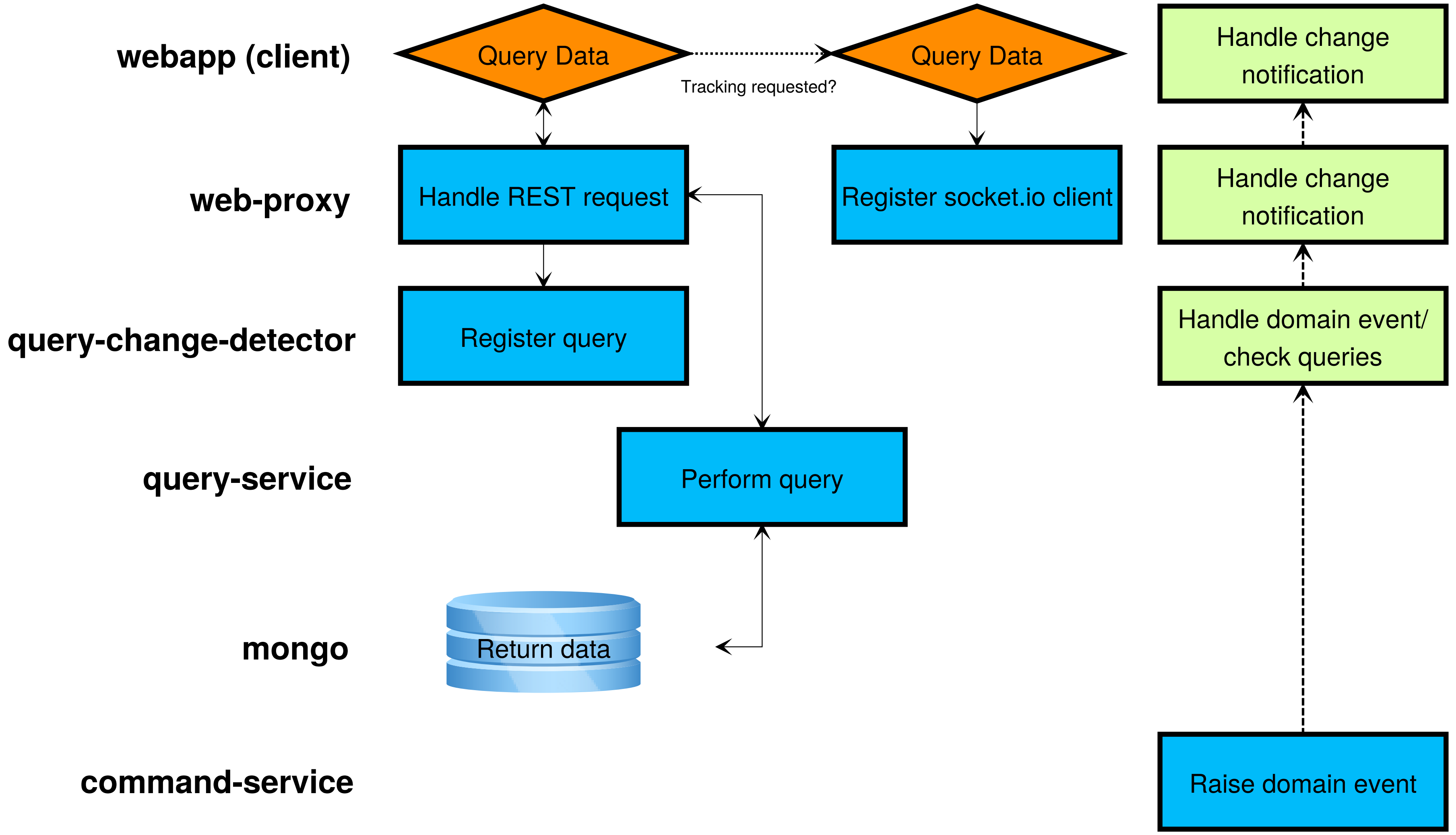
New Row

Service structure

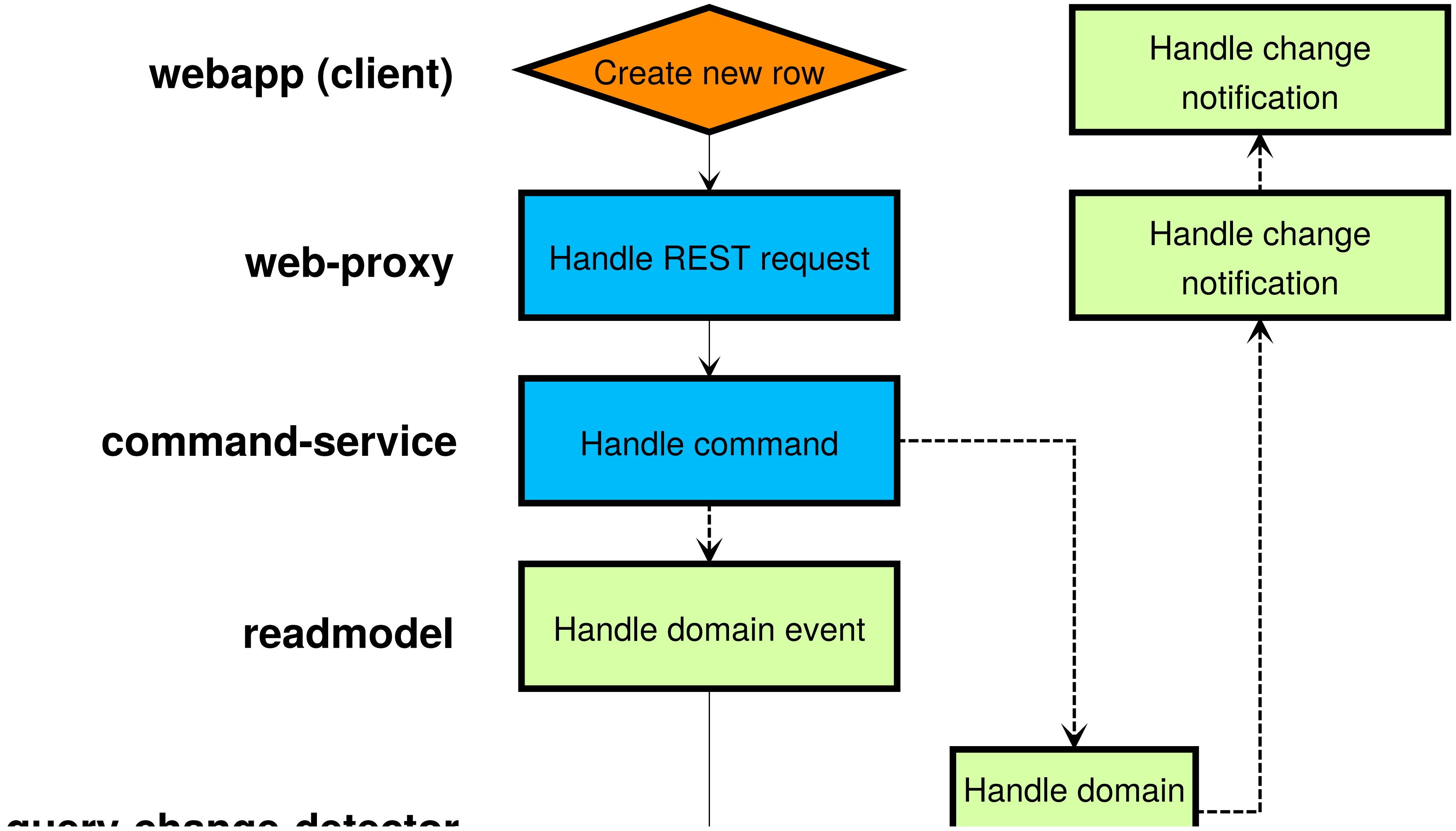
More advanced architecture has more services:

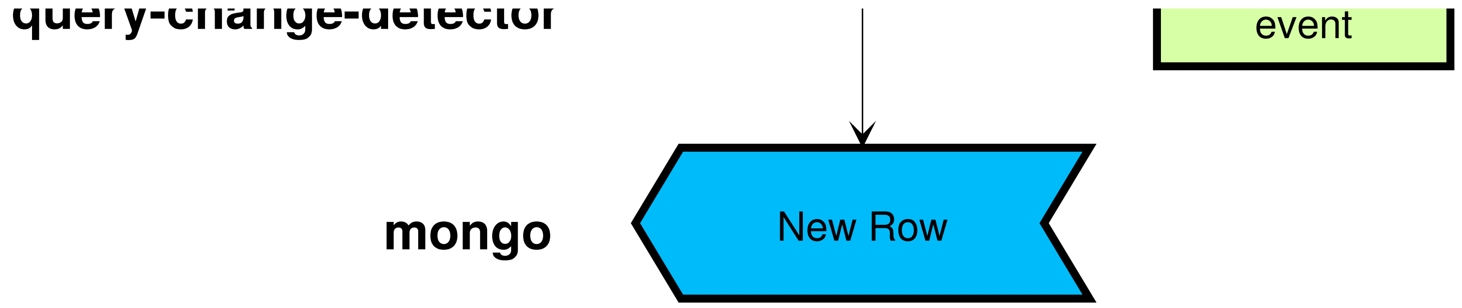


Querying data with CQRS/ES


webapp (client)
web-proxy
query-change-detector
query-service
mongo
command-service
Query Data
Tracking requested?
Query Data
Handle change notification
Handle REST request
Register socket.io client
Handle change notification
Register query
Handle domain event/ check queries
Perform query
Return data
Raise domain event

Creating a new row with CQRS/ES

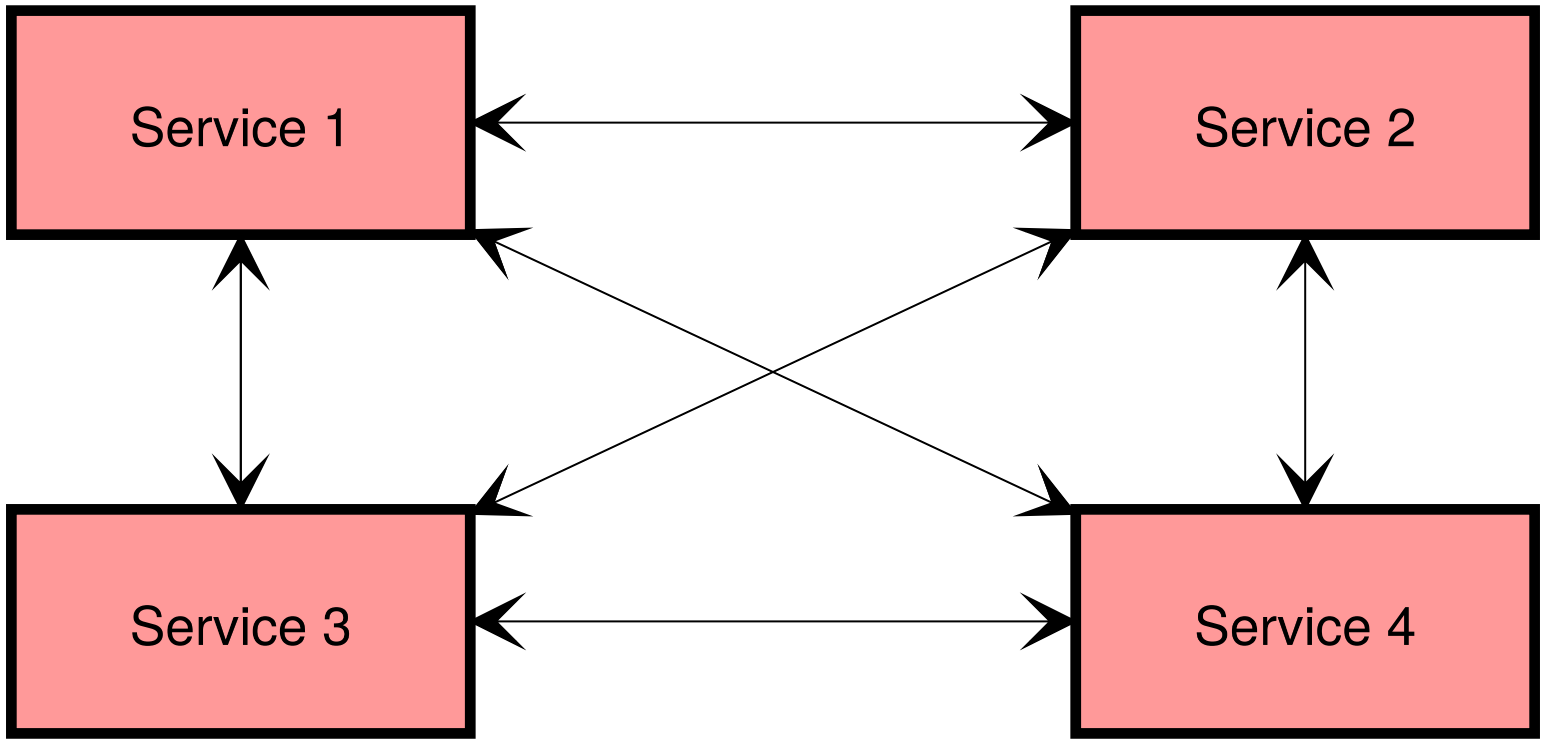



query-change-detector
event
mongo
New Row

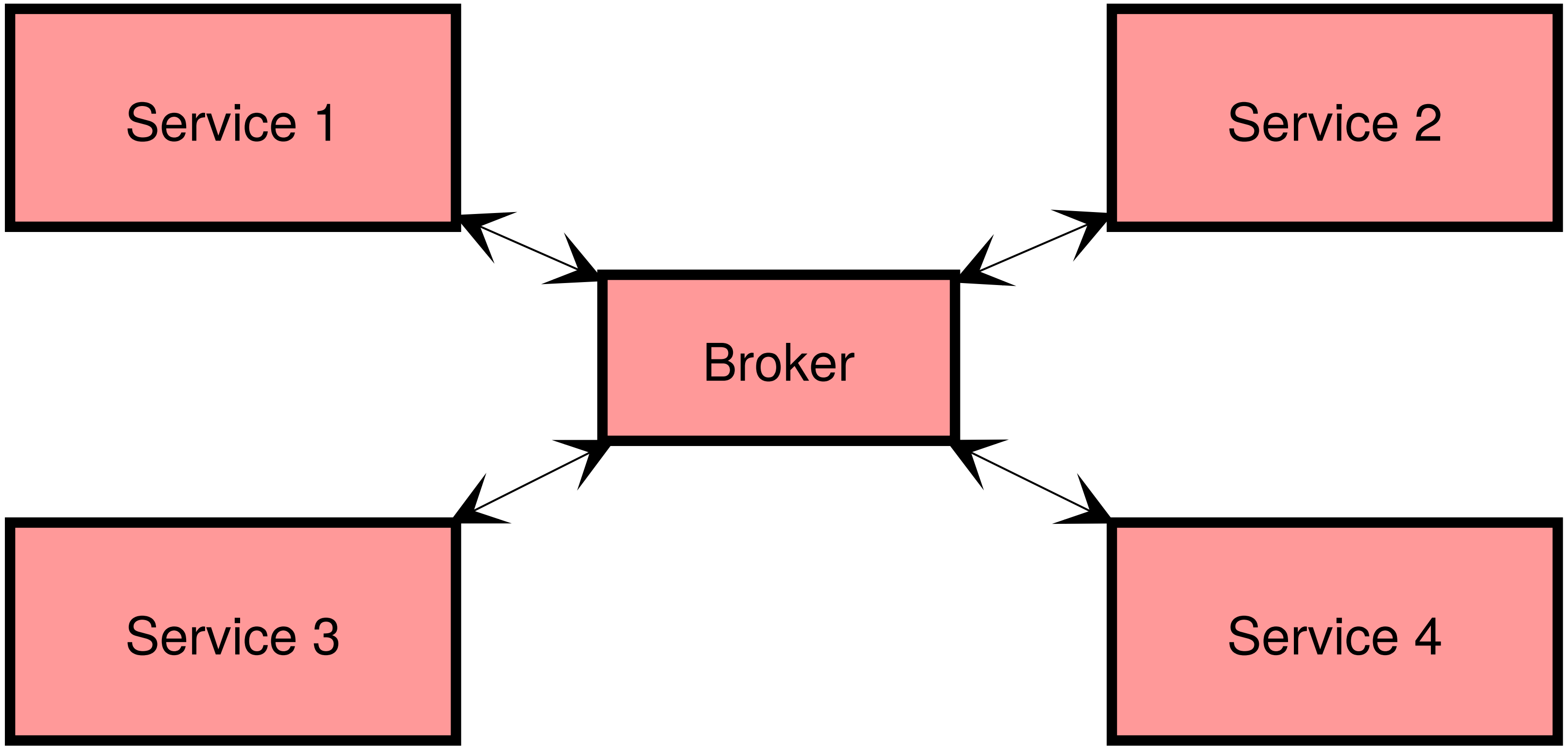
Communication

- 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 a web service. 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 easier to test
- Services can be debugged as individual entities
- Services **are** individual entities – best regards from functional programming!

Sources

- This presentation:
    - https://oliversturm.github.io/microservices-complete-picture
    - Deprettified content in pdf format: https://oliversturm.github.io/microservices-complete-picture/slidecontent.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@oliversturm.com