Pietro Balestra 24.11.22

Complex system

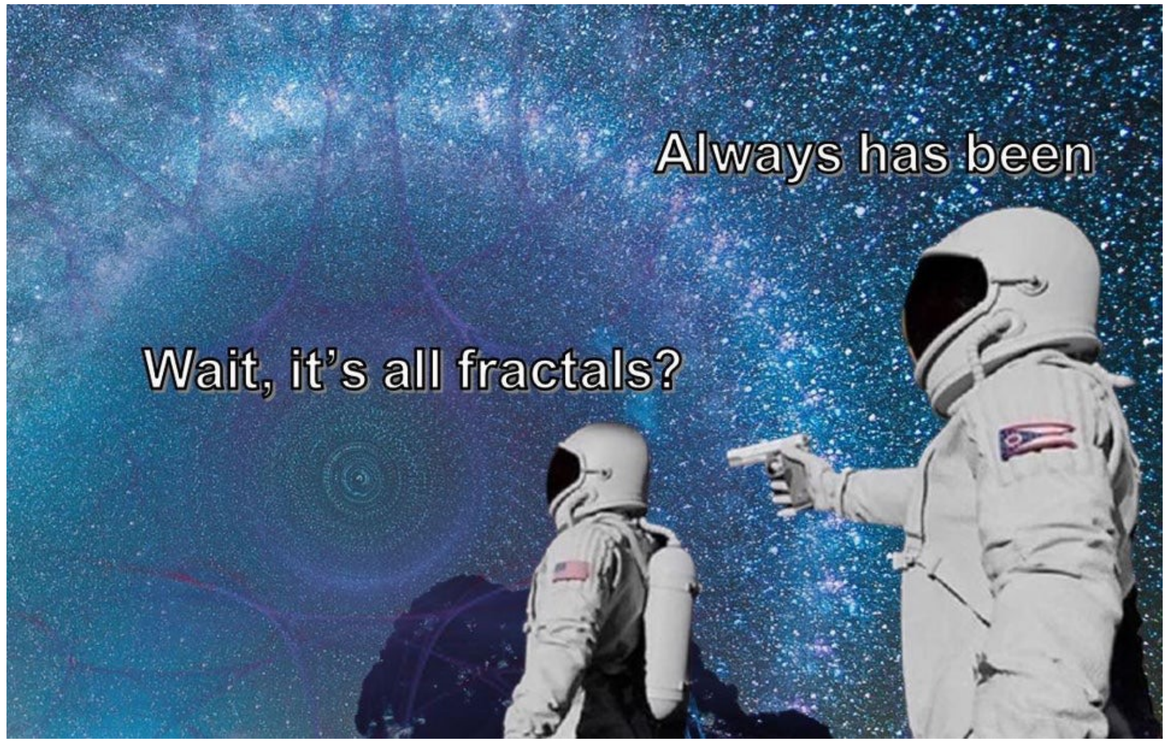
SOLID analogies

Complex system

- Entities and interactions

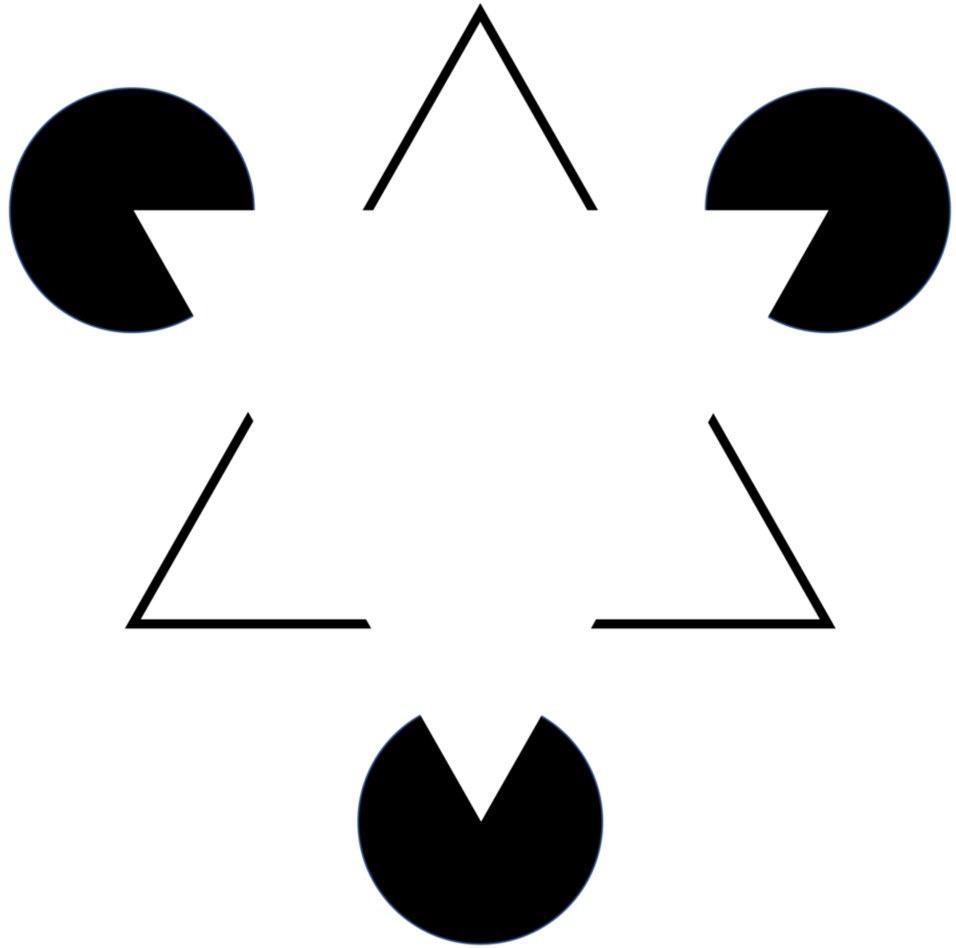
Complex system

- Entities and interactions
- Fractals



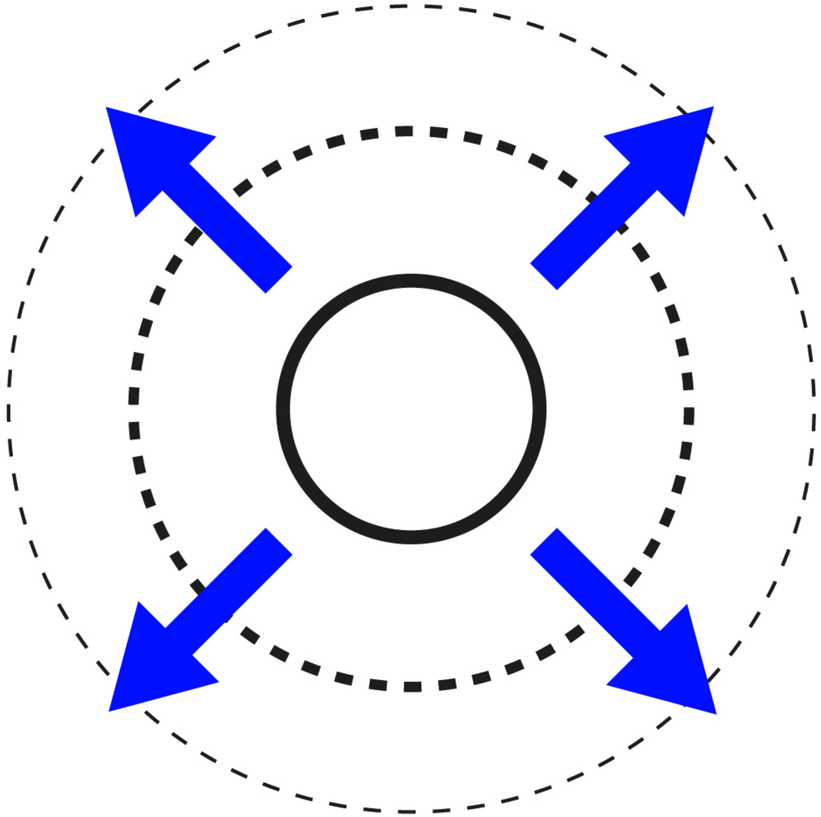
Complex system

- Entities and interactions
- Fractals
- Holistic



Complex system

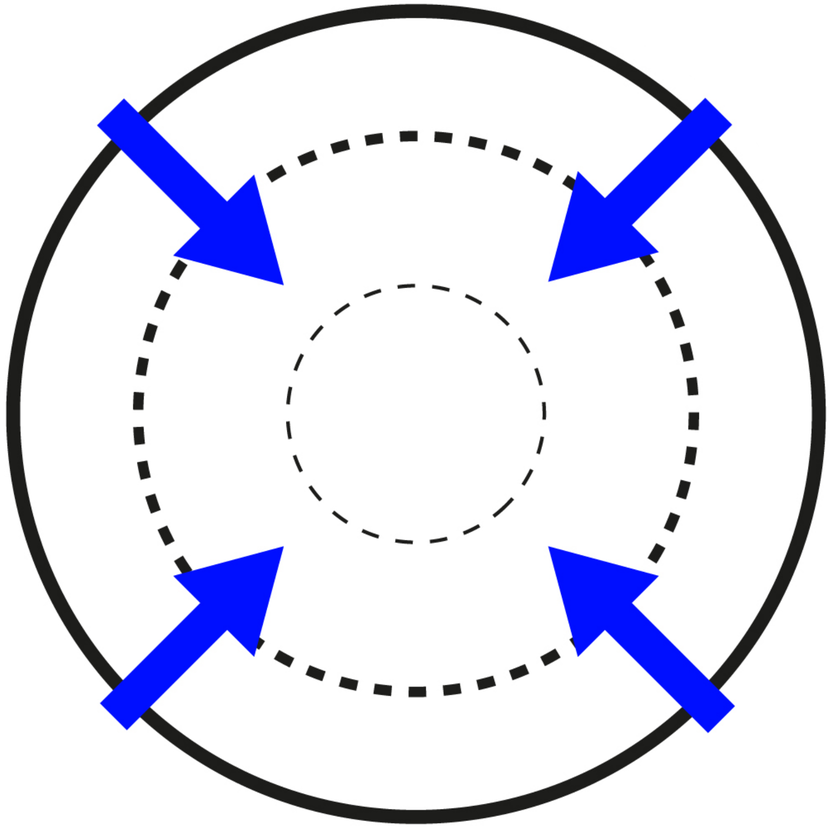
- Entities and interactions
- Fractals
- Holistic
- Classical approach



INSIDE-OUT

Complex system

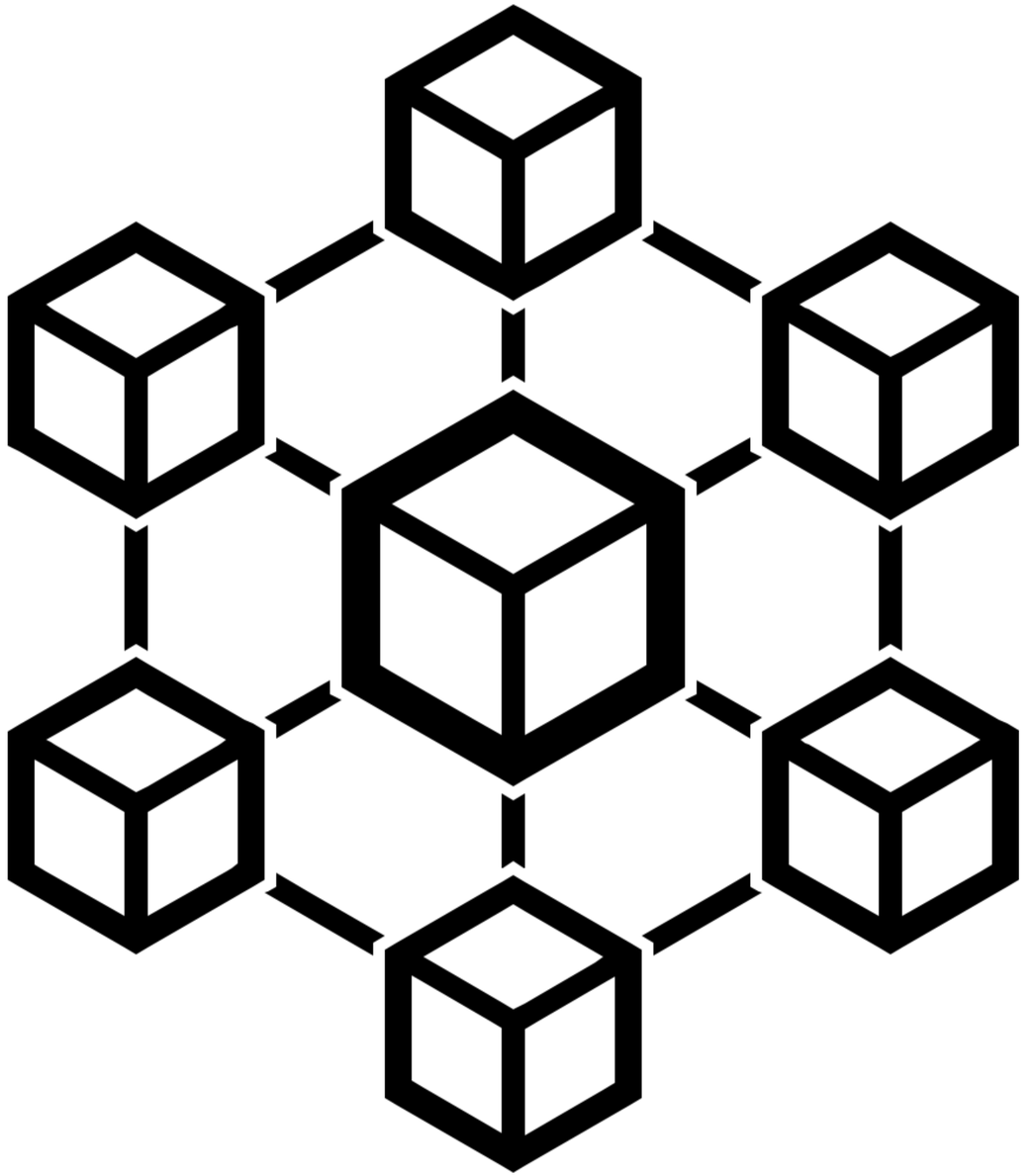
- Entities and interactions
- Fractals
- Holistic
- Classical approach
- Outside-In



OUTSIDE-IN

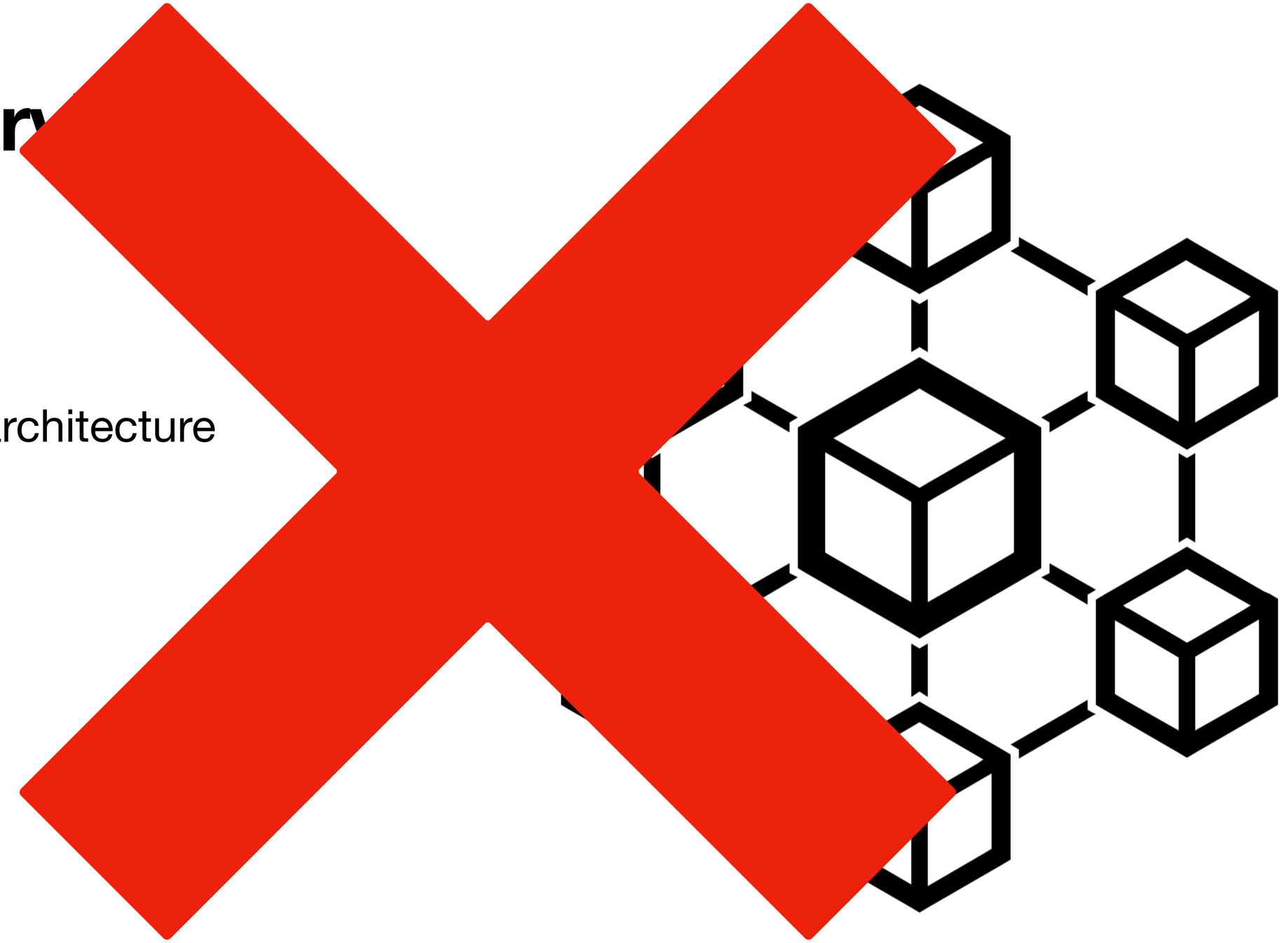
Microservices

- SOLID
- Hexagonal architecture
- Contracts
- Delivery
- Tests



Microserv

- SOLID
- Hexagonal architecture
- Contracts
- Delivery
- Tests

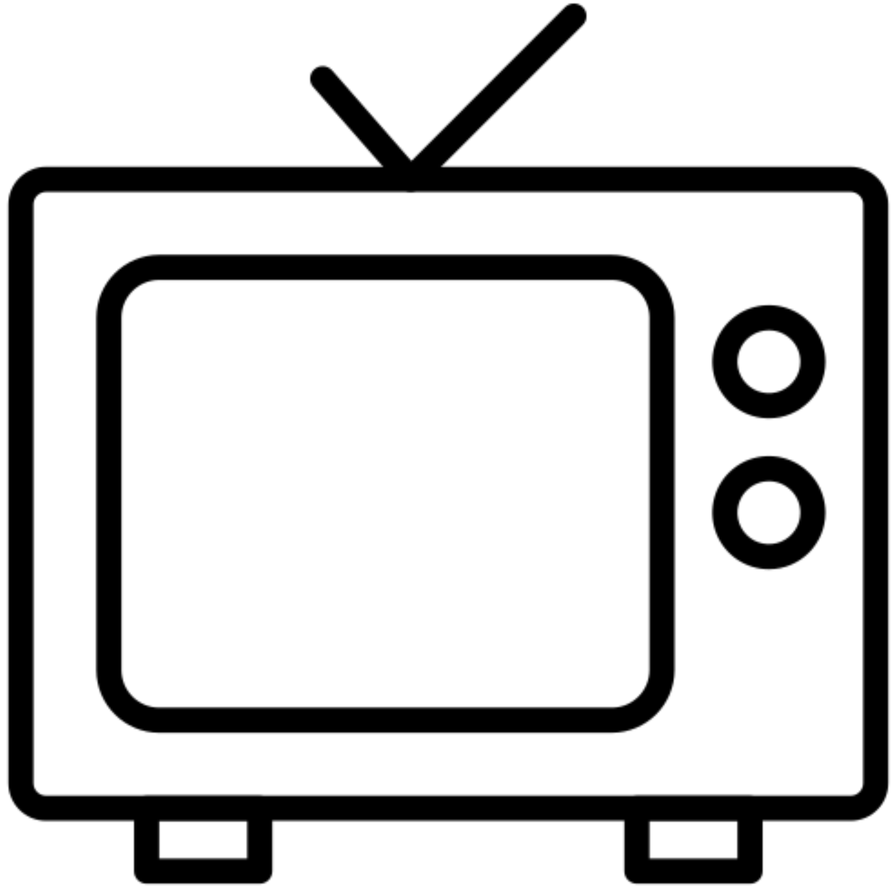


Microfrontends

- One UI to rule them all

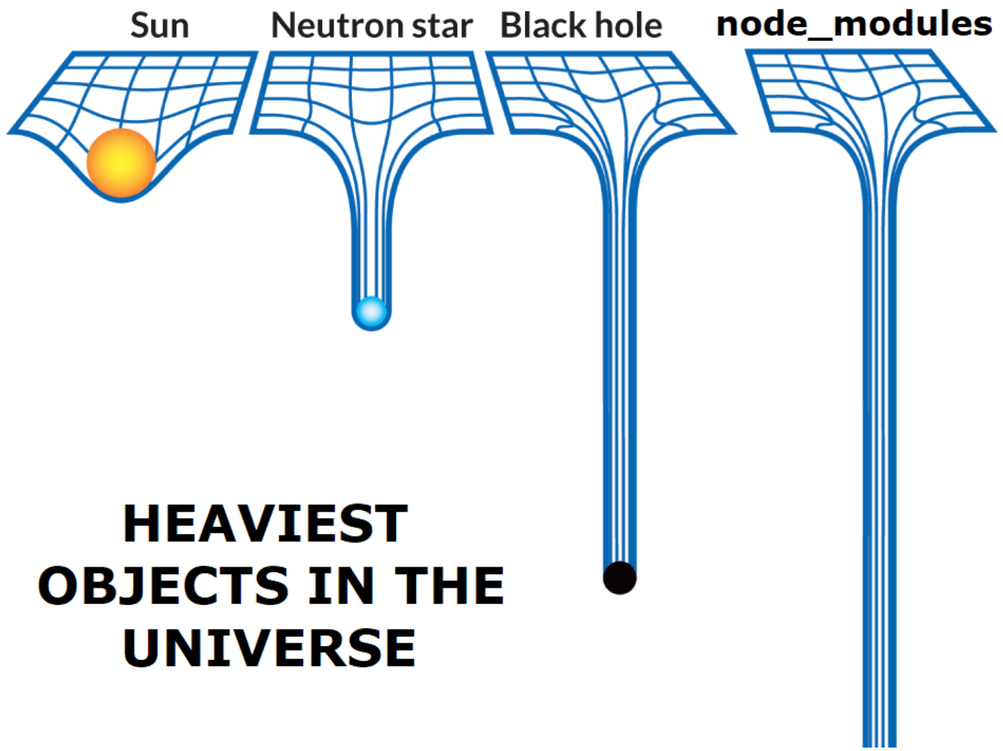
Microfrontends

- One UI to rule them all
- iFrame



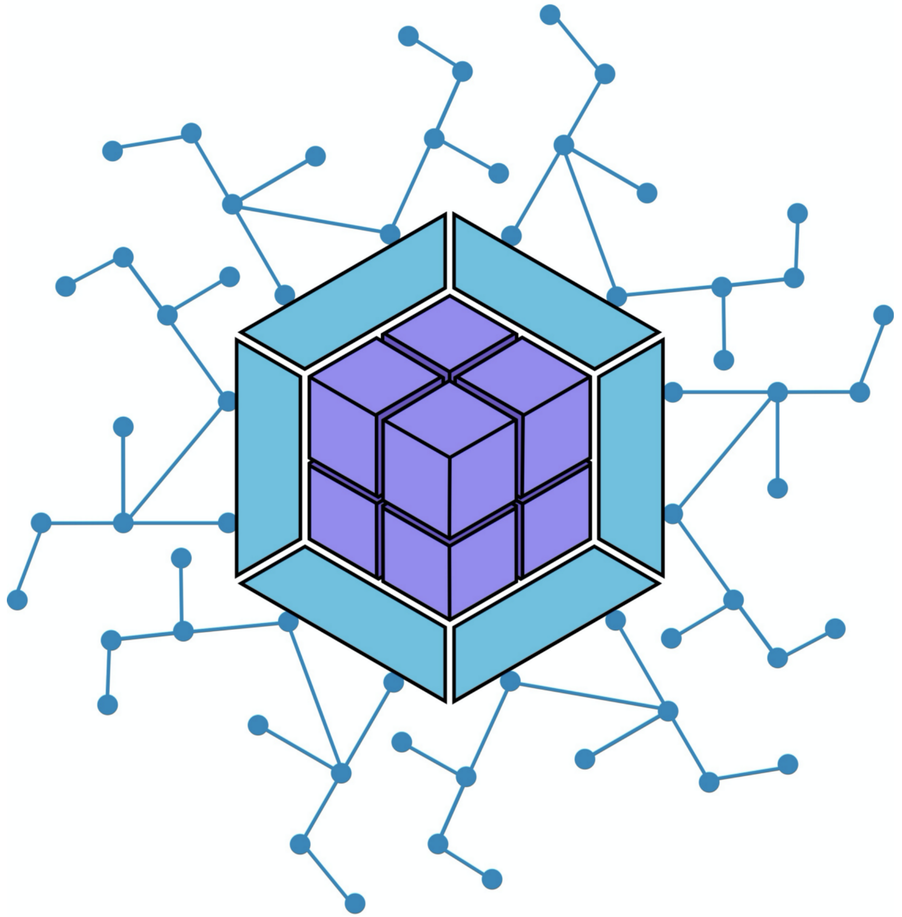
Microfrontends

- One UI to rule them all
- iFrame
- npm module



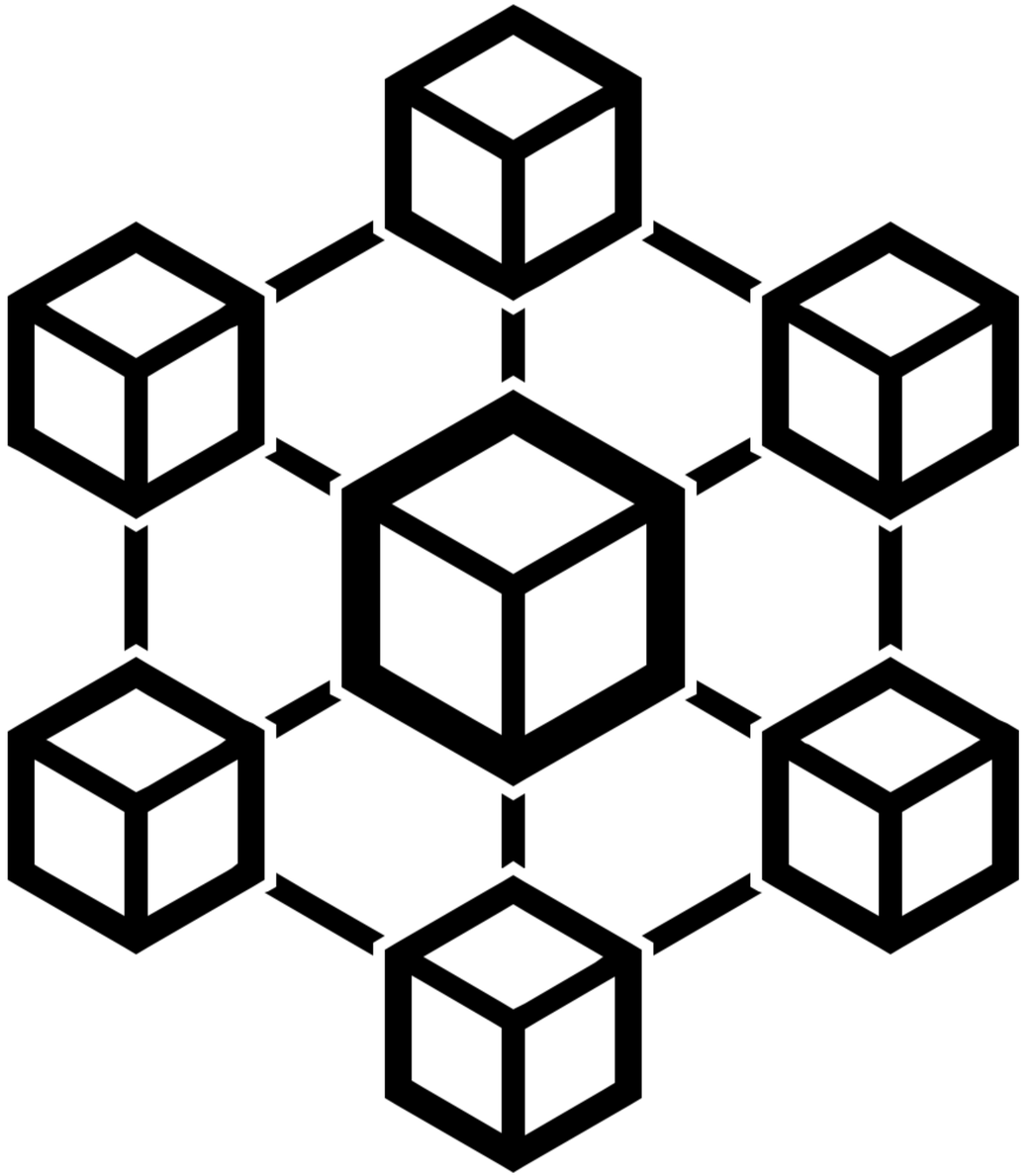
Microfrontends

- One UI to rule them all
- iFrame
- npm module
- Module federation



Microservices

- SOLID
- Hexagonal architecture
- Contracts
- Delivery
- Tests



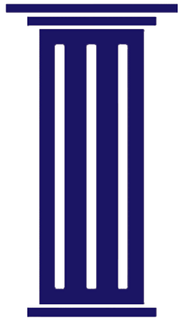
SOLID

Microservices

SOLID

Microservices

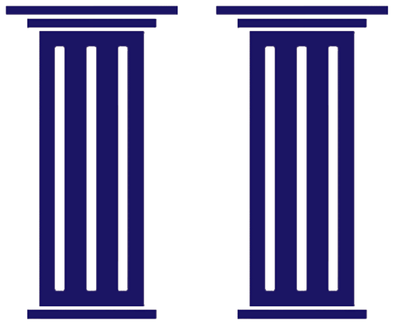
- **S**ingle-responsibility principle



SOLID

Microservices

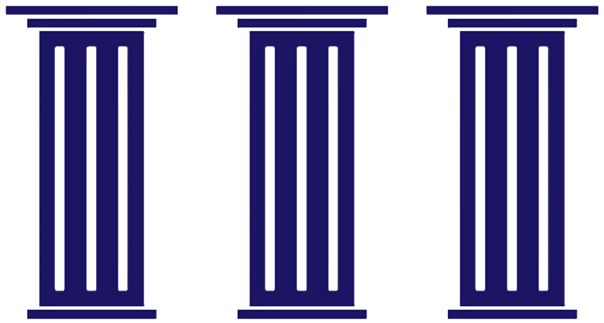
- **S**ingle-responsibility principle
- **D**ependency inversion principle



SOLID

Microservices

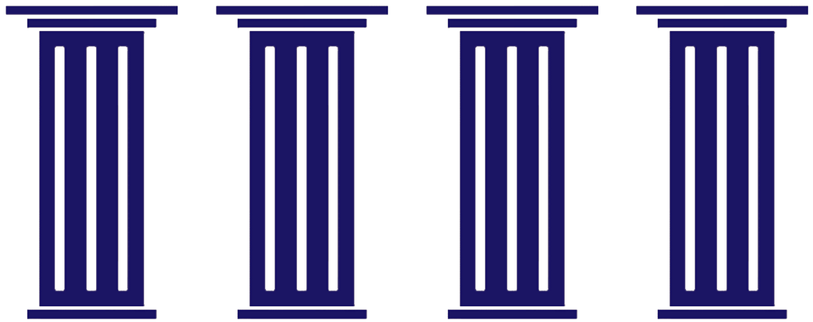
- **S**ingle-responsibility principle
- **D**ependency inversion principle
- **L**iskov substitution principle



SOLID

Microservices

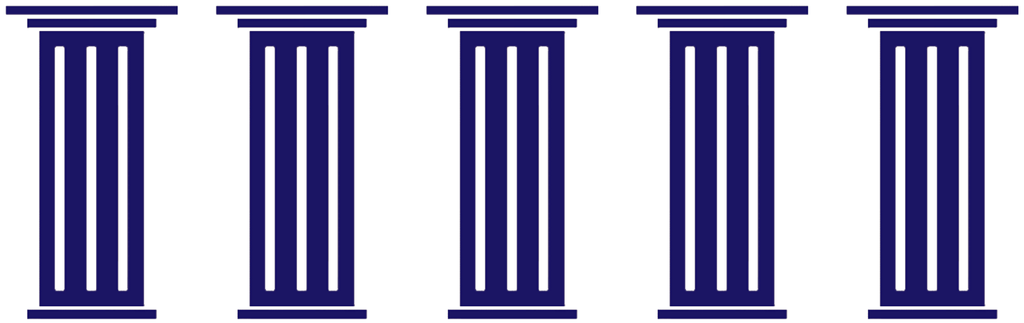
- **S**ingle-responsibility principle
- **D**ependency inversion principle
- **L**iskov substitution principle
- **I**nterface segregation principle



SOLID

Microservices

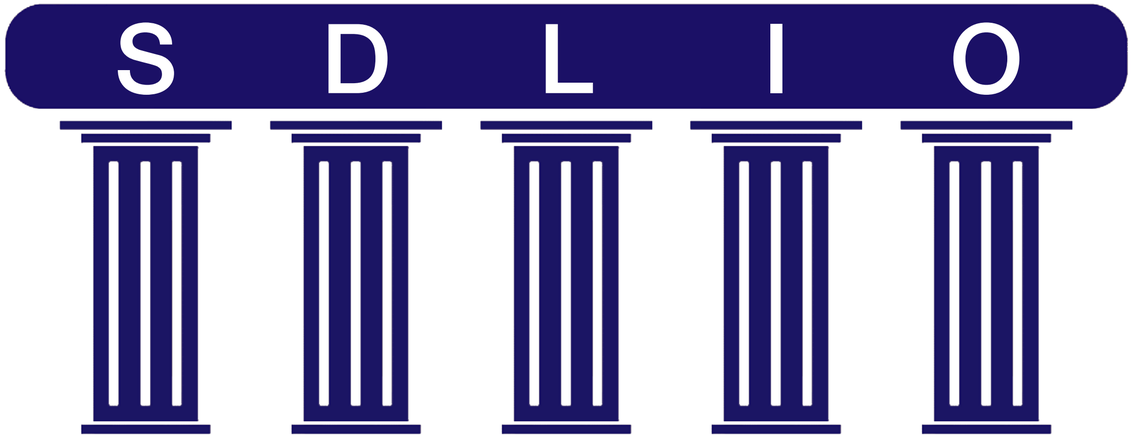
- **S**ingle-responsibility principle
- **D**ependency inversion principle
- **L**iskov substitution principle
- **I**nterface segregation principle
- **O**pen-closed principle



SOLID

Microservices

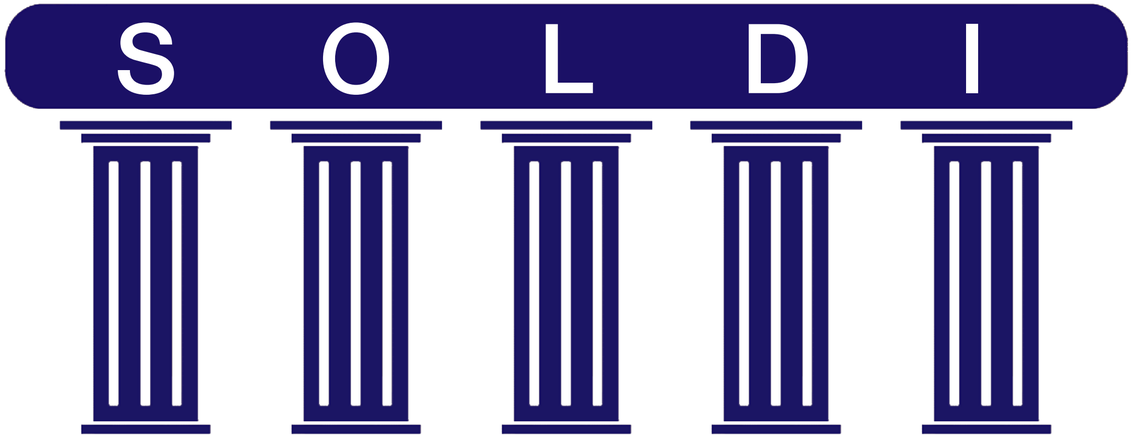
- **S**ingle-responsibility principle
- **D**ependency inversion principle
- **L**iskov substitution principle
- **I**nterface segregation principle
- **O**pen-closed principle



SOLID

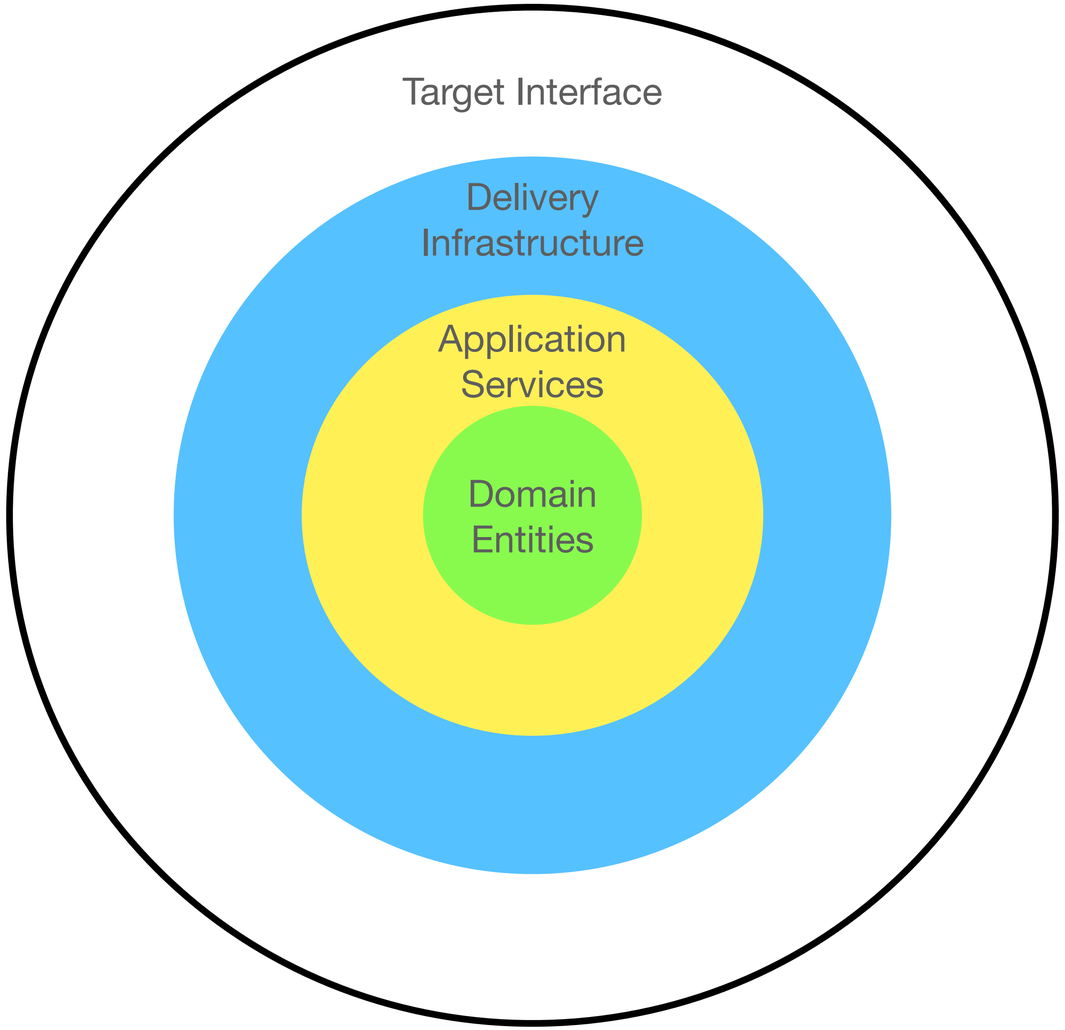
Microservices

- **S**ingle-responsibility principle
- **D**ependency inversion principle
- **L**iskov substitution principle
- **I**nterface segregation principle
- **O**pen-closed principle



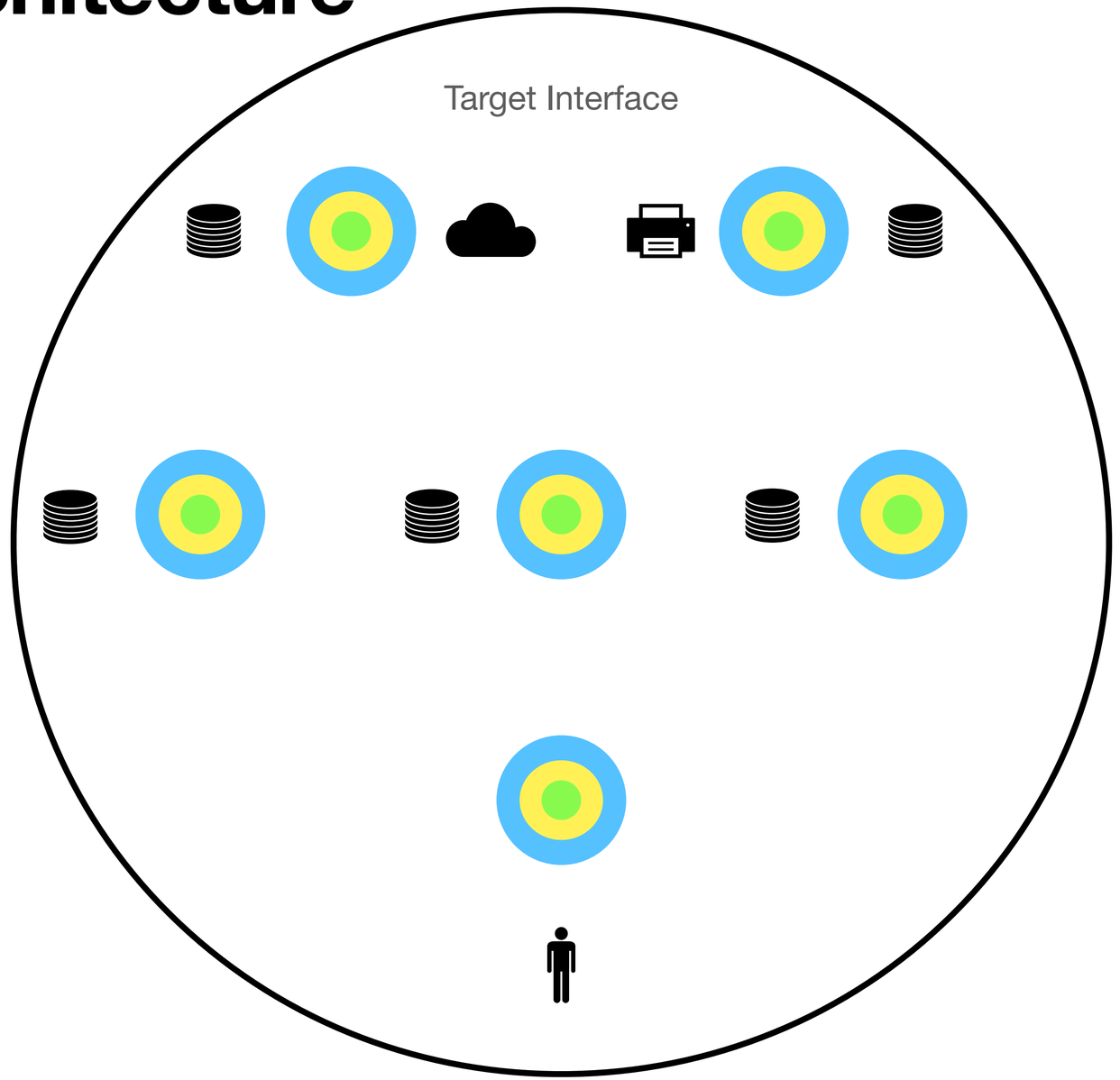
Hexagonal architecture

Service



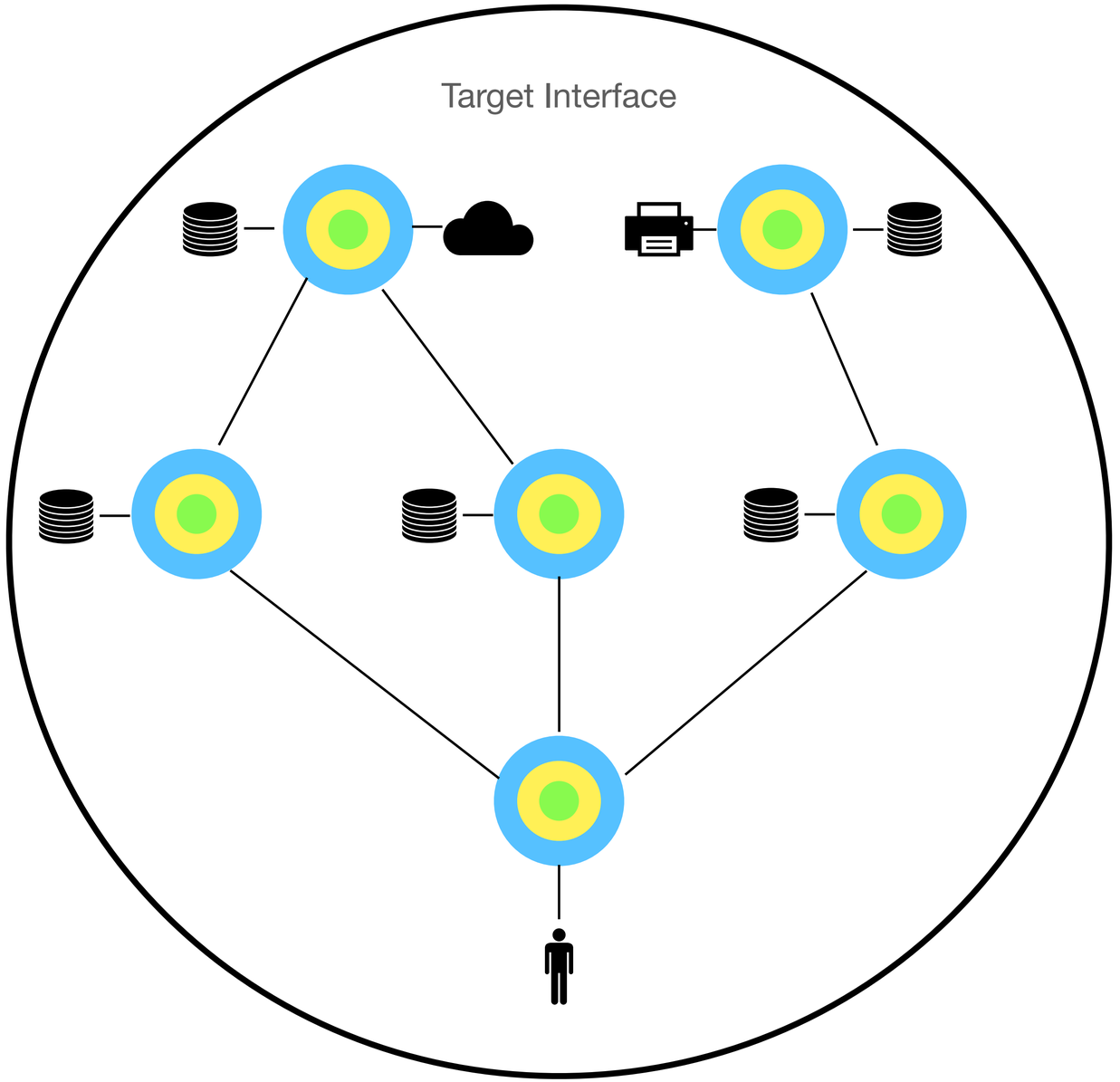
Hexagonal architecture

Microservices



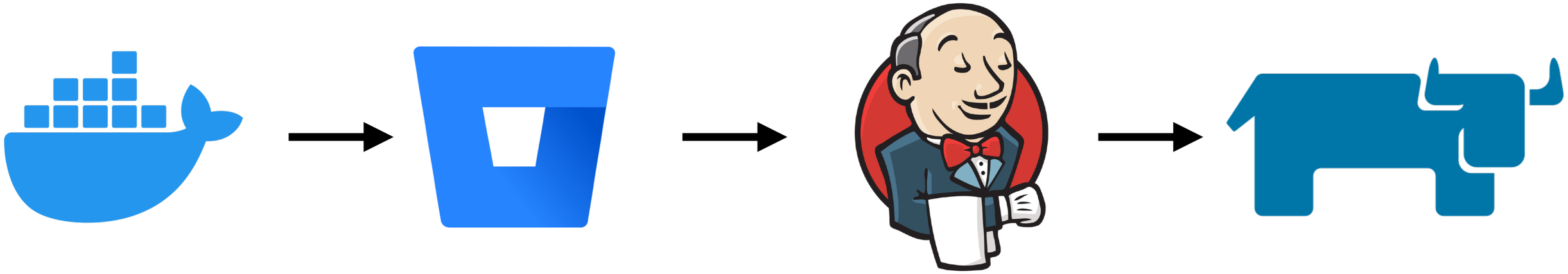
Contracts

Microservices



Delivery

Microservices

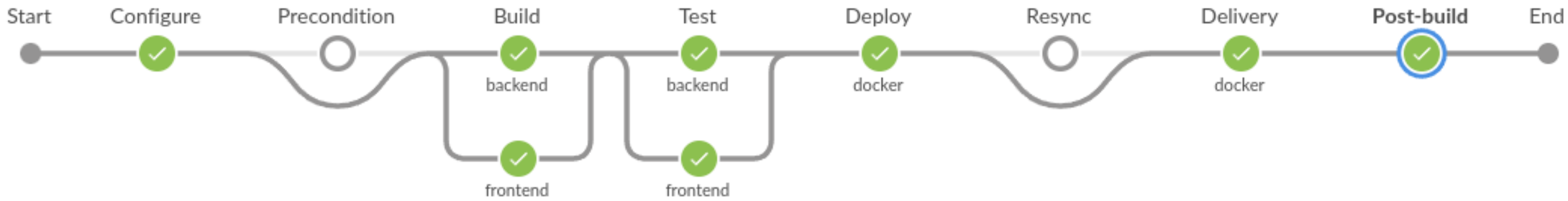


Delivery

Microservices

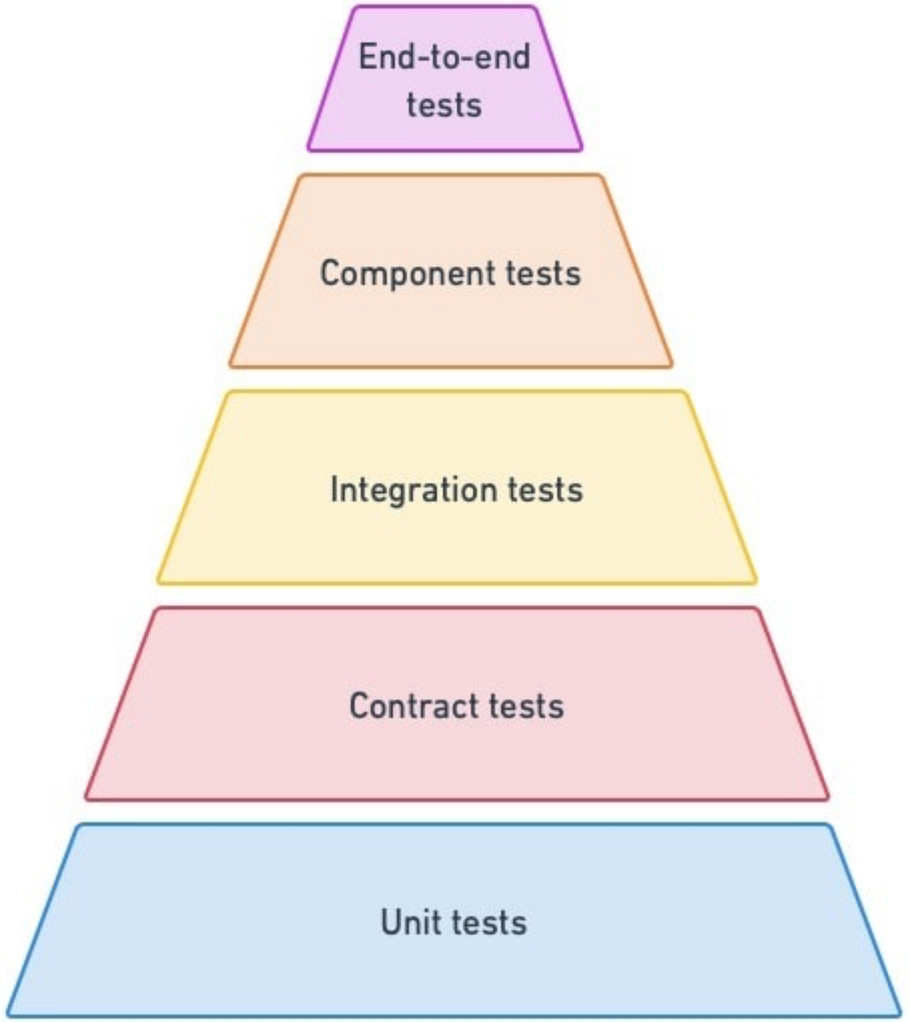
Branch: master | 4m 18s | Changes by Fornoni Pierluigi

Commit: `fe4f95f` | 2 hours ago | Branch event



Tests

Microservices



References

- https://semaphoreci.com/blog/test-microservices
- https://www.educative.io/blog/solid-principles-oop-c-sharp
- https://imgflip.com/memegenerator

Question?

Thank you for your attention

- pietro@balestra.dev
- github.com/p1e7r0