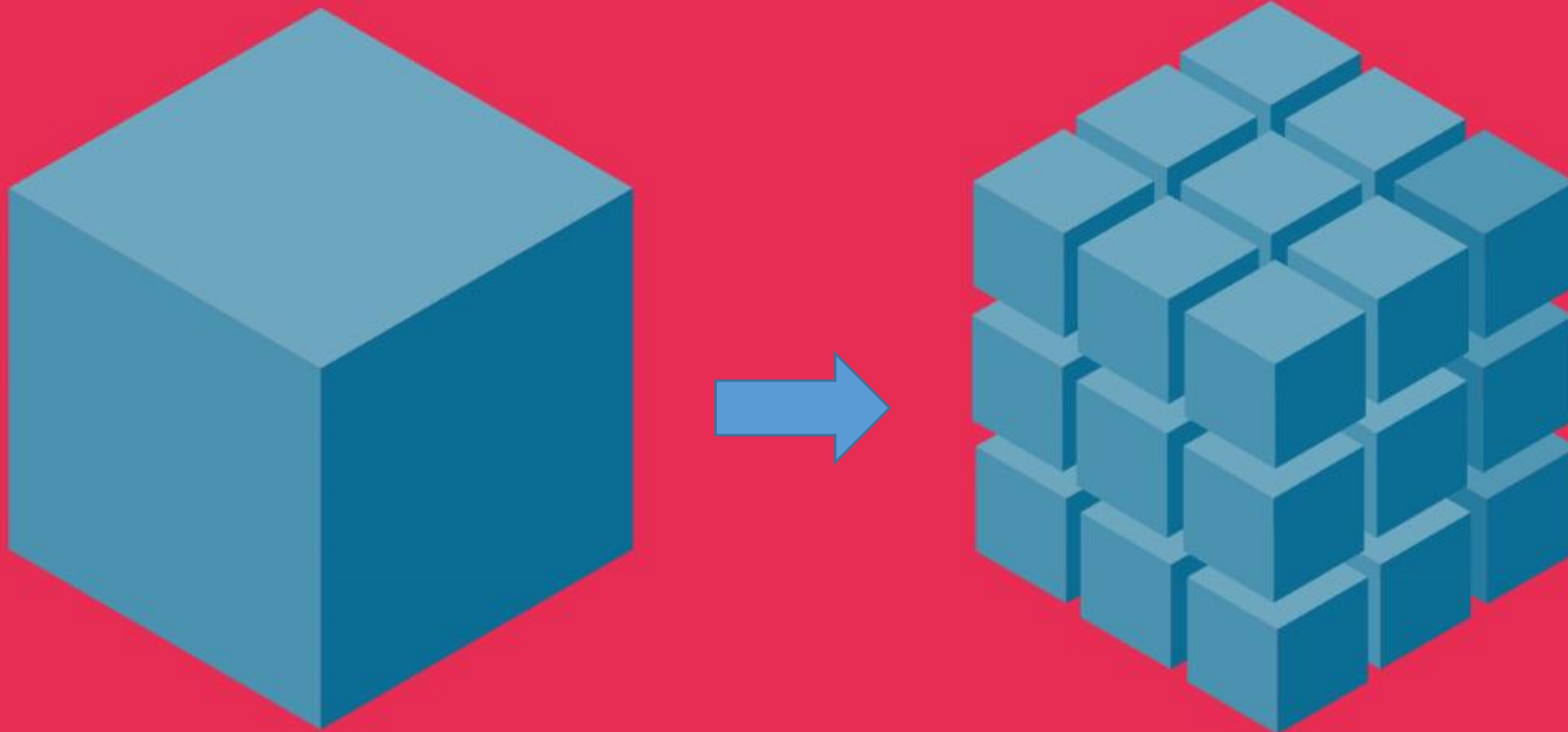


Reactive architecture

Rovinj, 10.10.2022
Saša Ivičević



Evolution in application architecture



reactive

adjective | re·ac·tive | \ rē- 'ak-tiv \

Definition of *reactive*

1 a: readily responsive to a stimulus



```
int b = 1
int c = 2
int a = b + c
b = 10
System.out.println(a) // 3
(not 12 because "=" is not
"reactive" operator)
```

```
int b = 1
int c = 2
int a $= b + c
b = 10
System.out.println(a) // 12
```

\$= special "reactive" operator

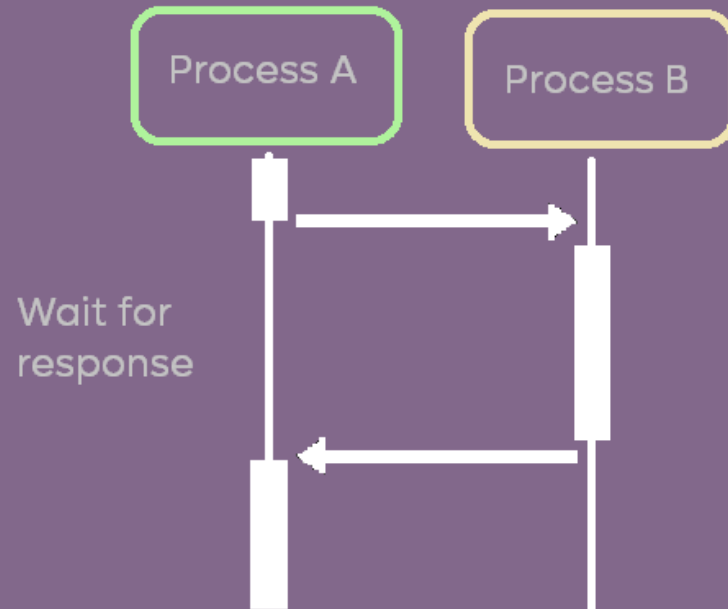
$$C = A + B$$

f_x	=SUM(A1, B1)		
	A	B	C
1	1	2	3
2			
3			

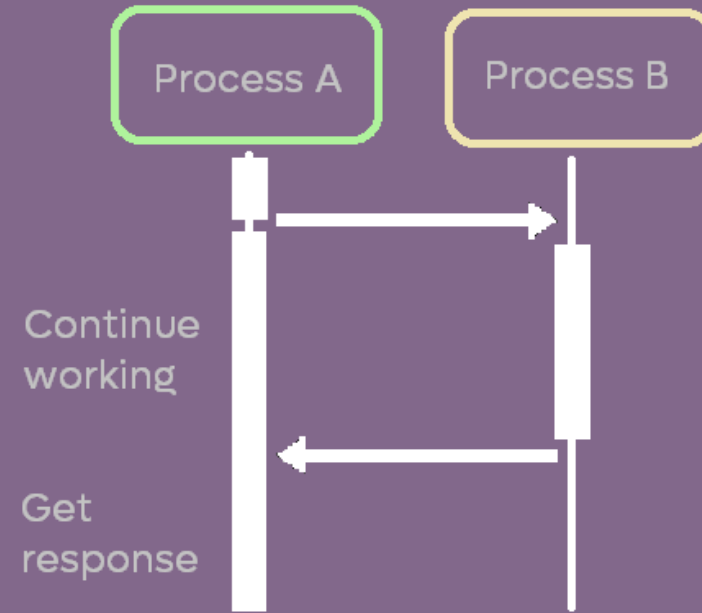


Sync. vs Async

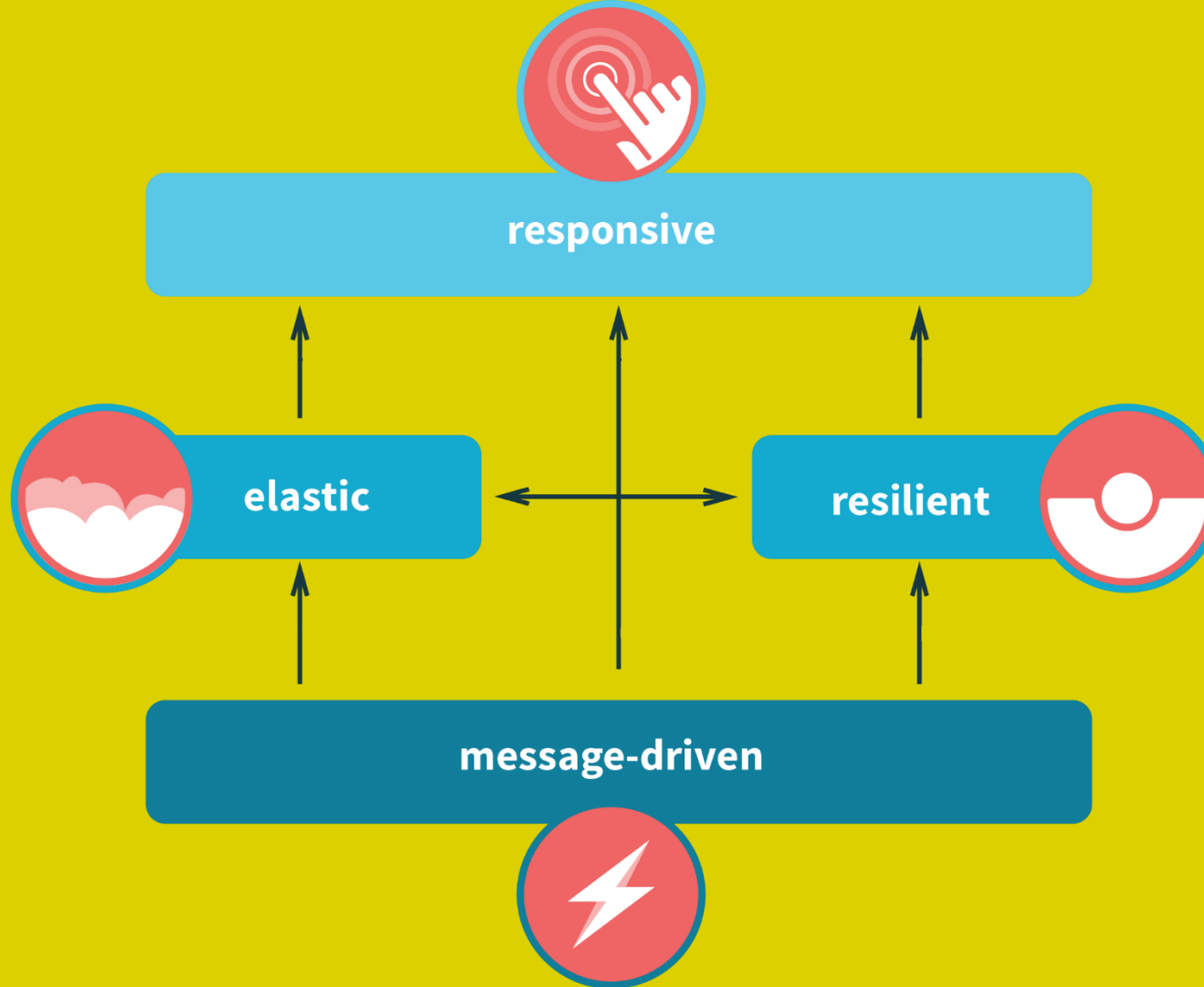
Synchronous



Asynchronous



Reactive manifesto



Responsiveness

3 seconds

That's how long it takes for your visitors to get impatient and leave your website

60%

Leave the site if the page takes more than 3 seconds to load



80%

Of these do not come back to visit the website



50%

Almost half of them communicates to his friends the negative experience



0.0000001 seconds after
Instagram goes down



Resilience

- Replication
- Isolation
- Delegation
- Containment



Elasticity

Elasticity provides responsiveness, despite increase (or decrease) in load



Message driven

- Responsiveness, Resilience, Elasticity are supported by Message Driven Architecture.
- Messages are asynchronous and non-blocking.
- It provides loose coupling, isolation and location transparency.
- Resources are consumed only while active.



(RESILIENCE + ELASTICITY) x MESSAGE DRIVEN = RESPONSIVE





Advantages

- Response time
- Resource optimization
- Scaling and growth
- Paralelysm
- Stability



Disadvantages

- Not for simple use cases
- Debugging
- Complexity



Reactive applications

Created with
reactive
components

Developed
using reactive
programming

Organized
within an
reactive
architecture

Following
reactive
manifesto



THANK YOU

