

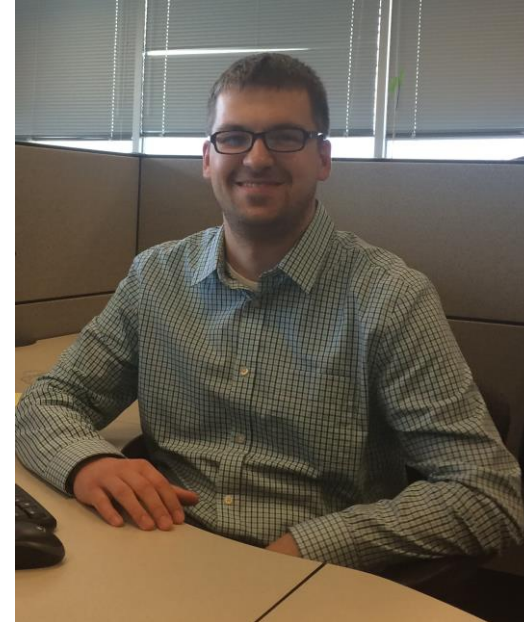
Event-driven architecture for a 12-factor app

How the event-driven architecture fits into the “Backing services”,
“Processes” and “Disposability” factors

Dmitrii Sosedov

NICE TO MEET YOU!

- ▶ Husband
- ▶ Father
- ▶ Software engineer
- ▶ Bookworm



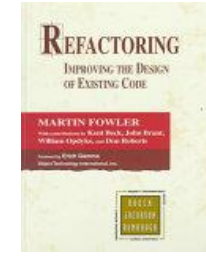
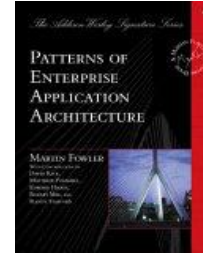
LinkedIn: <https://www.linkedin.com/in/dsosedov>

GitHub: <https://github.com/dsosedov>

Web site: <http://dmitrii.sosedov.org>

The Twelve-Factor App

Source: <https://12factor.net>



▶ I. Codebase

One codebase tracked in revision control, many deploys

▶ II. Dependencies

Explicitly declare and isolate dependencies

▶ III. Config

Store config in the environment

▶ IV. Backing services

Treat backing services as attached resources

▶ V. Build, release, run

Strictly separate build and run stages

▶ VI. Processes

Execute the app as one or more stateless processes

▶ VII. Port binding

Export services via port binding

▶ VIII. Concurrency

Scale out via the process model

▶ IX. Disposability

Maximize robustness with fast startup and graceful shutdown

▶ X. Dev/prod parity

Keep development, staging, and production as similar as possible

▶ XI. Logs

Treat logs as event streams

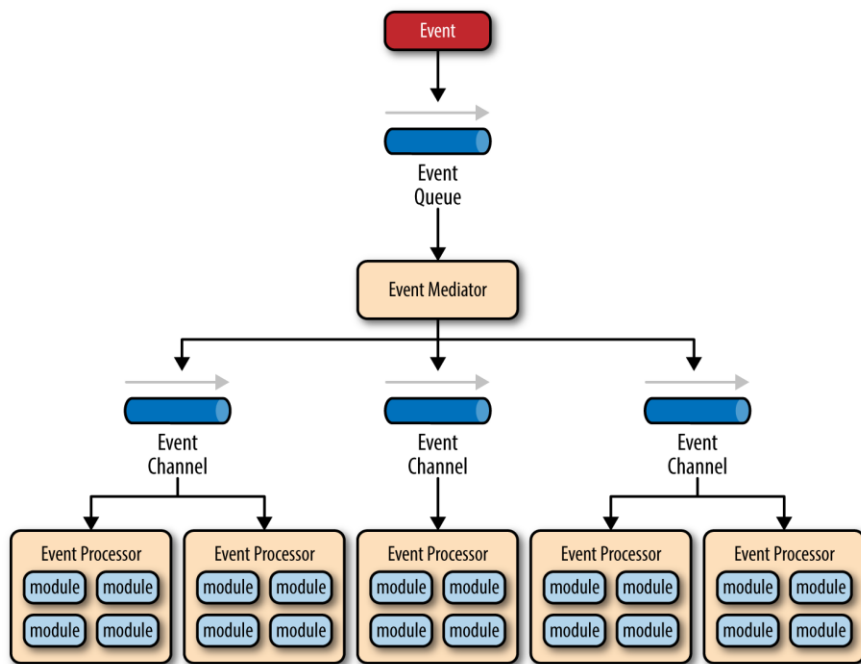
▶ XII. Admin processes

Run admin/management tasks as one-off processes

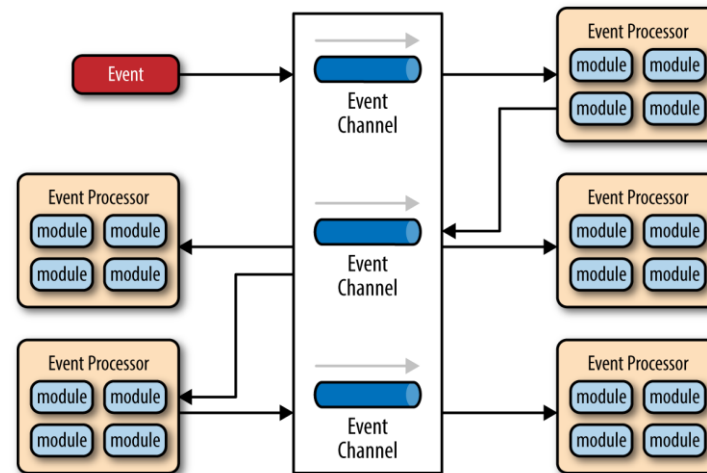
The event-driven architecture

The event-driven architecture is made up of highly decoupled, single-purpose event processing components that asynchronously receive and process events.

Mediator Topology



Broker Topology



Demo!

WAKE UP, NEO...

Event-driven arch meets 12-factor app

By following the distributed asynchronous architecture pattern, we turn a legacy set of services into a highly scalable and robust SaaS solution.

Sync app

- ▶ Tight coupling
- ▶ Some processes have multiple reasons to change
- ▶ Resources cannot be detached
- ▶ High chances for data to get out of sync
- ▶ Stopping a resource must be planned

Async app

- ▶ Loose coupling
- ▶ Each process has one and only one reason to change
- ▶ Resources can be detached
- ▶ Low chances for data to get out of sync
- ▶ Resource can stop at any given point



The deck and the sample app source code are available at <https://github.com/dsosedov/event-driven-arch-demo>

Questions?

Feel free to ask!

