Event-driven architecture for a 12-factor app

How the event-driven architecture fits into the “Backing services” and “Stateless processes” 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

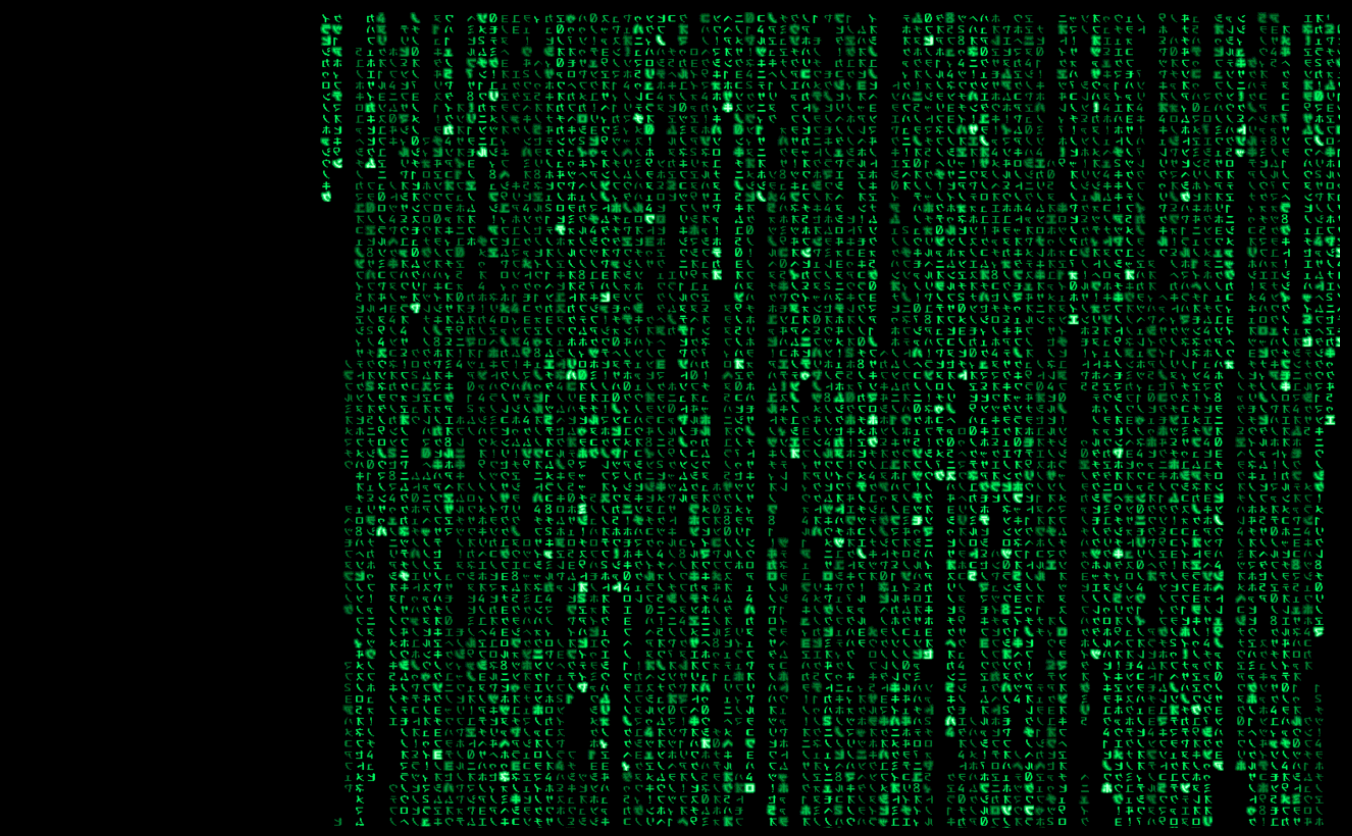
Run admin/management tasks as one-off processes

The event-driven architecture

Event-driven arch meets 12-factor app

Demo!

Wake up, Neo…



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

Questions?

Feel free to ask!