

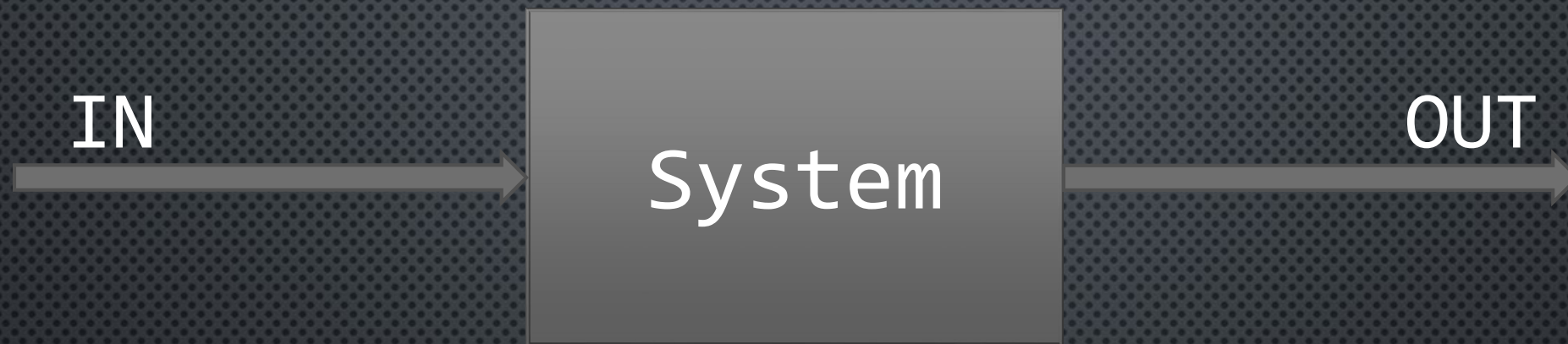
# SPRING APPS MONITORING

Ivan Macalák

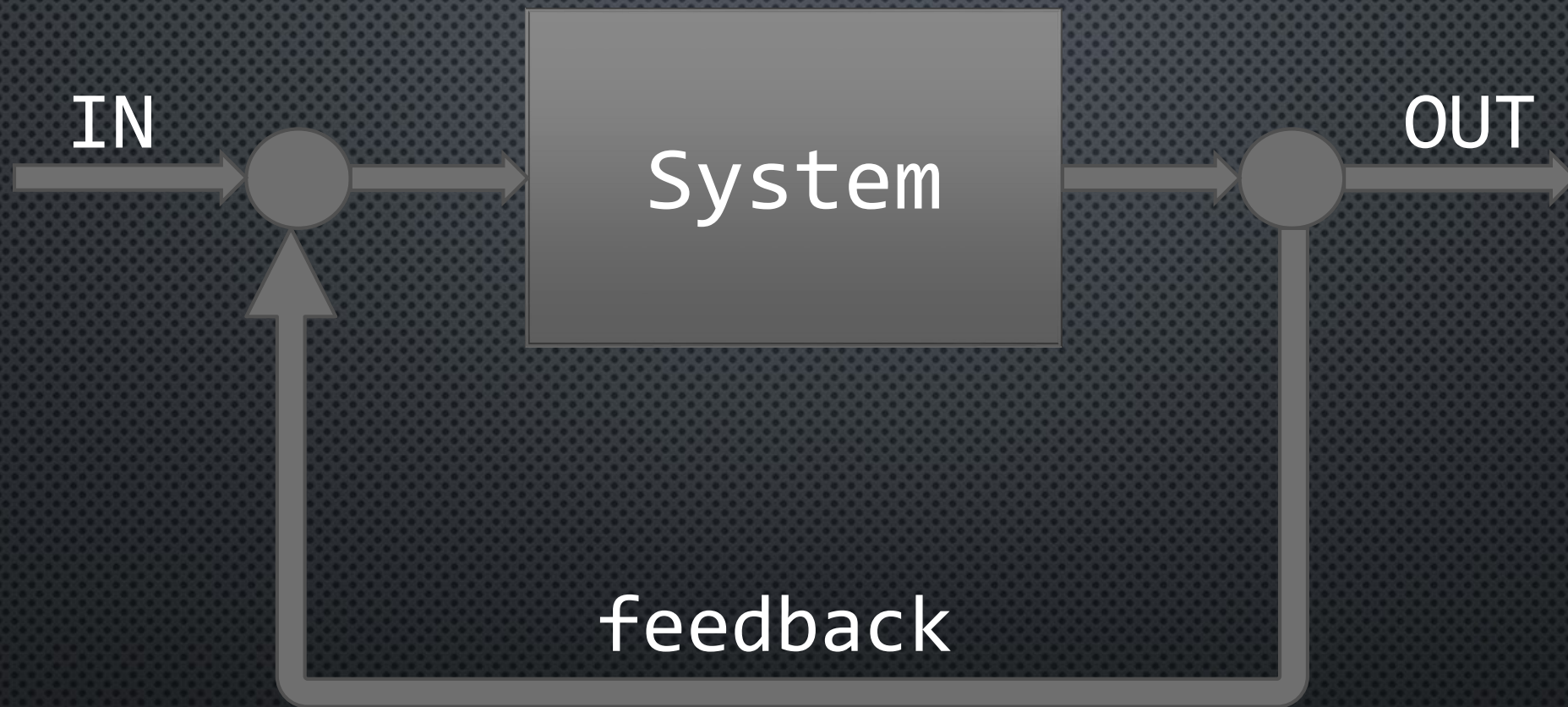
SW Architect/Java Lector/Mentor  
macalak@itexperts.sk

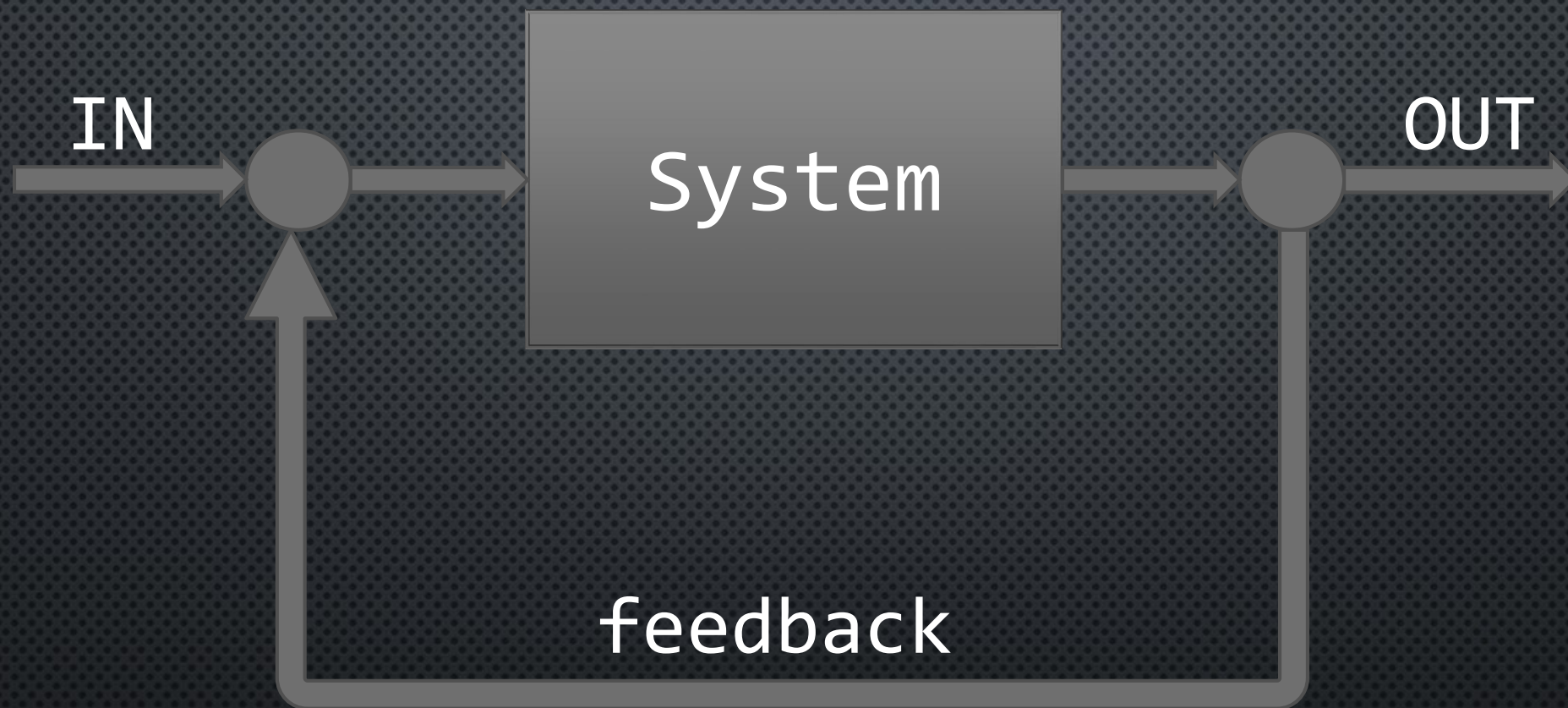
<https://sk.linkedin.com/in/ivanmacalak>

<https://github.com/macalak/javadays2017>













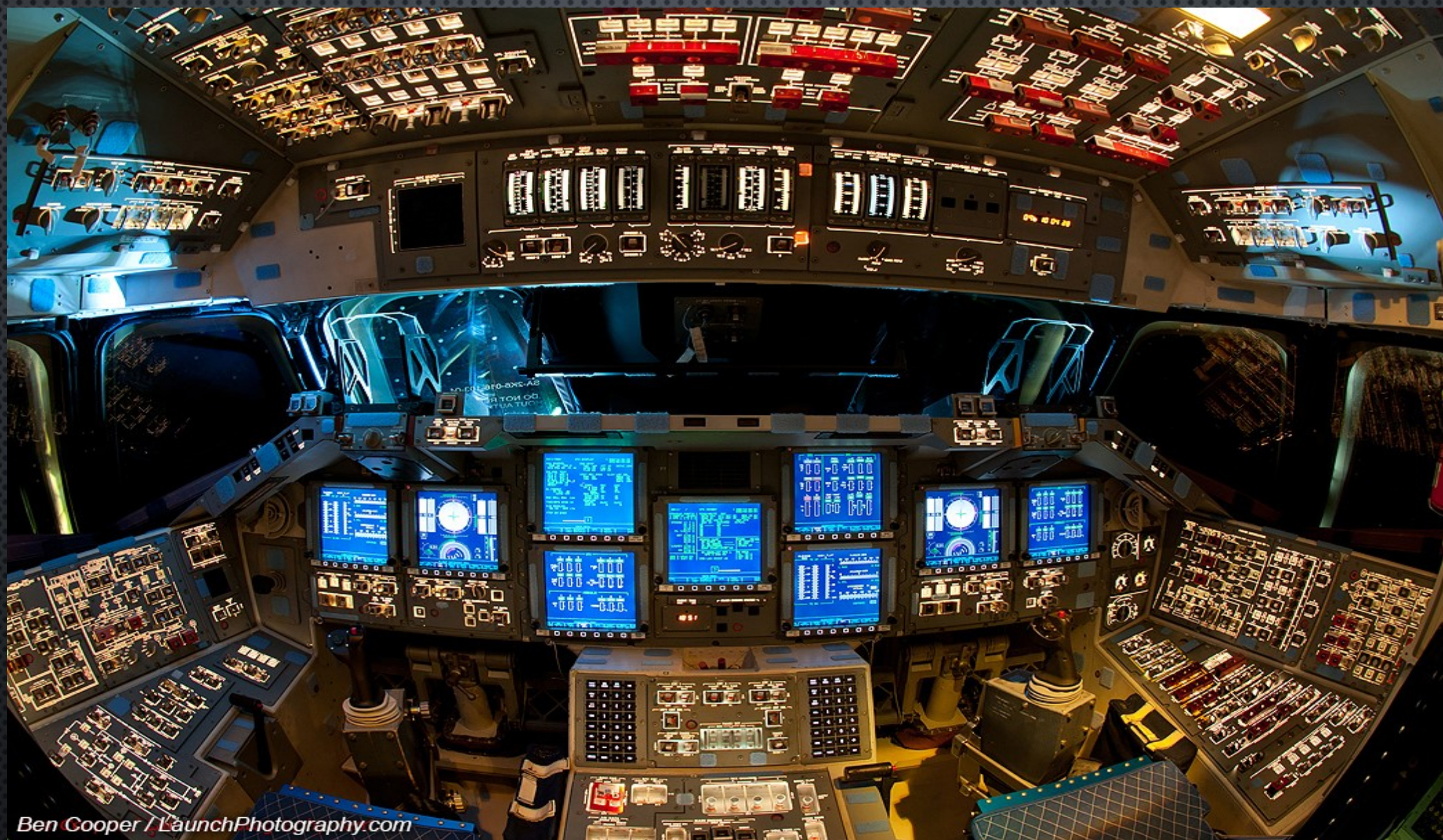




COPYRIGHT THOMAS MERKL

AIRLINERS.NET





Ben Cooper / LaunchPhotography.com



DEV



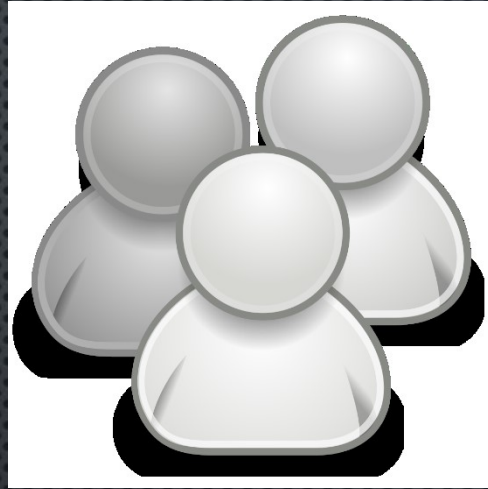
OPS



SEC



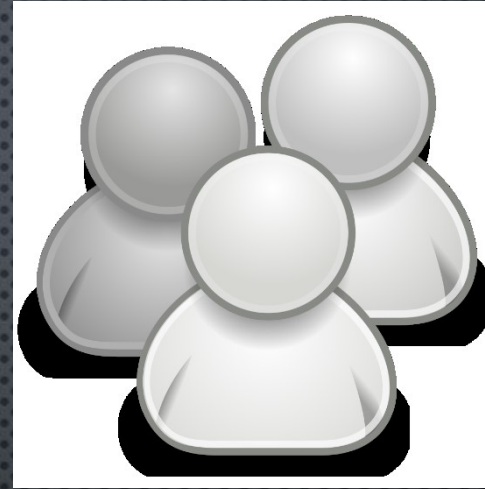
# DEV



- :> Errors
- :> Performance
- :> Requests traces
- :> Diagnostic



- :> Availability
- :> Entity changes
- :> Critical errors
- :> Performance
- :> SLA



OPS

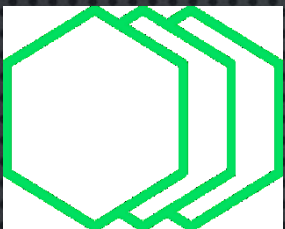


- :> Attacks
- :> Unsuccessful log-ins
- :> Accounts detail changes
- :> Unauthorized data access



SEC

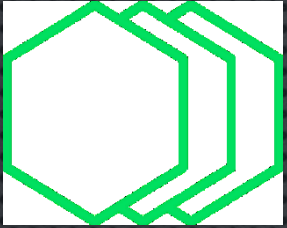




# Microservices







# Microservices monitoring

- :> Distributed loosely coupled services
- :> Asynchronous and event driven
- :> You should monitor health of service
- :> Services should provide useful metrics
- :> Services should log important messages





# Spring Boot

- :> Java based framework for building microservices
- :> Pre-configured dependencies
- :> Auto configuration
- :> Starters
- :> Single executable jar
- :> Embedded servers
- :> <http://start.spring.io> for quick start





# Spring Boot Actuator

:> Provides useful HTTP endpoints for application monitoring and interaction as:

- :> /health
- :> /metrics
- :> /info
- :> /loggers
- :> /refresh
- :> /restart





# Actuator Health

- :> Status of running application
- :> Autoconfigured health indicators
- :> Enable/disable sensitive info
- :> Write your own





# Actuator Metrics

- :> System metrics
- :> Data Source metrics
- :> Cache metrics
- :> Tomcat metrics
- :> Dropwizard metrics
- :> Micrometer metrics
- :> Counters
- :> Gauges
- :> Metrics exporters
- :> Write your own





# Actuator Auditing

:> Works with Spring Security

- :> authentication success
- :> authentication failures
- :> access denied
- :> write your own

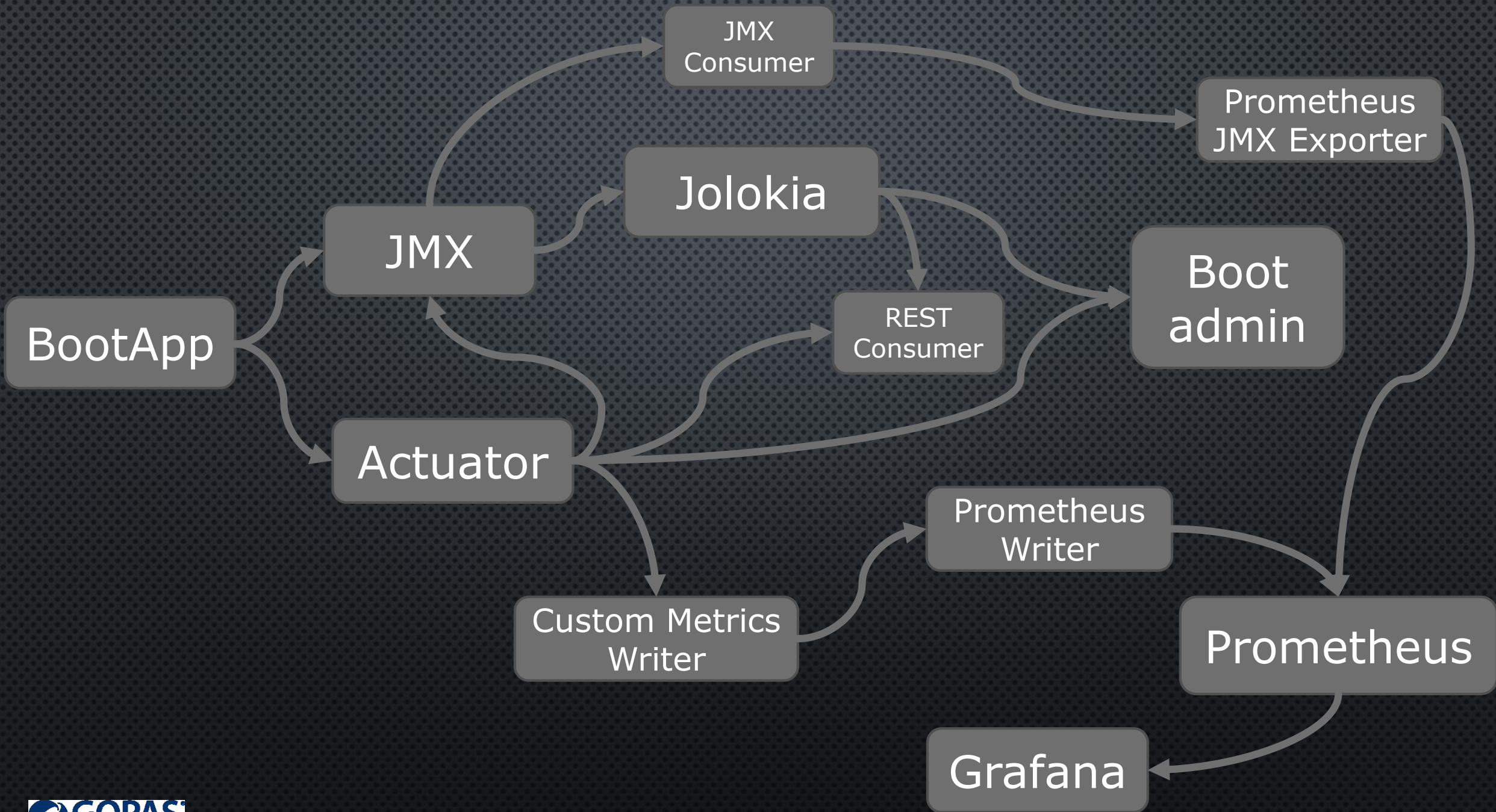




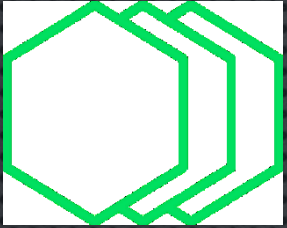
# Spring Boot Admin

- :> Server for management and monitoring Spring Boot applications
- :> Supports Netflix Eureka discovering
- :> Uses Spring Boot Actuator endpoints





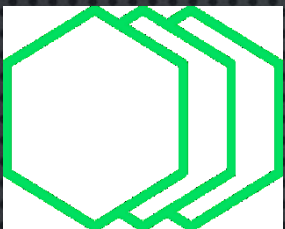




# Microservices logging

- :> Distributed loosely coupled services
- :> Asynchronous and event driven
- :> You should have centralized logs solution
- :> Services should provide useful logs
- :> Logs analyze
  - :> search
  - :> visualization
  - :> alerts

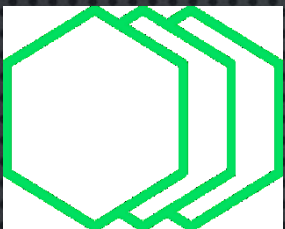




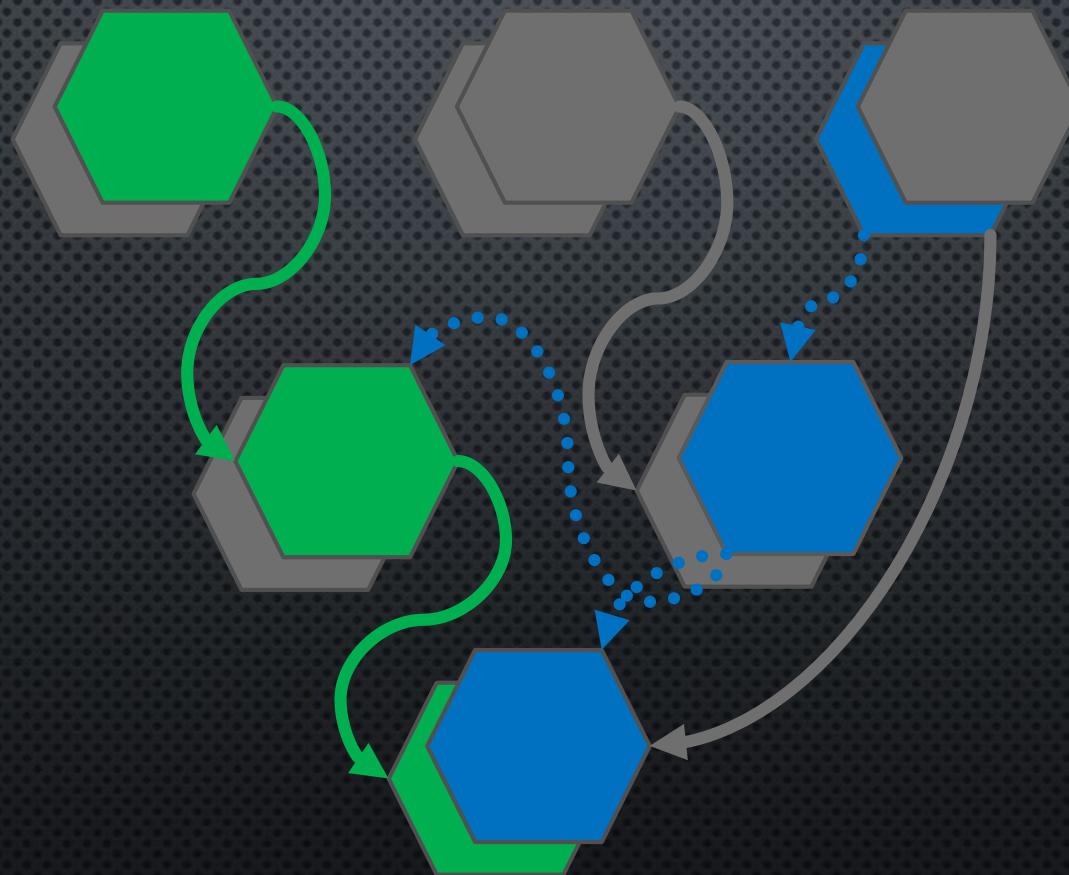
# Microservices calls







# Microservices events







# Spring Sleuth

- :> Distributed tracing solution
- :> Instruments Spring application
- :> Logs tracing data
  - :> Trace ID
  - :> Span ID
- :> Sends tracing data to remote collectors

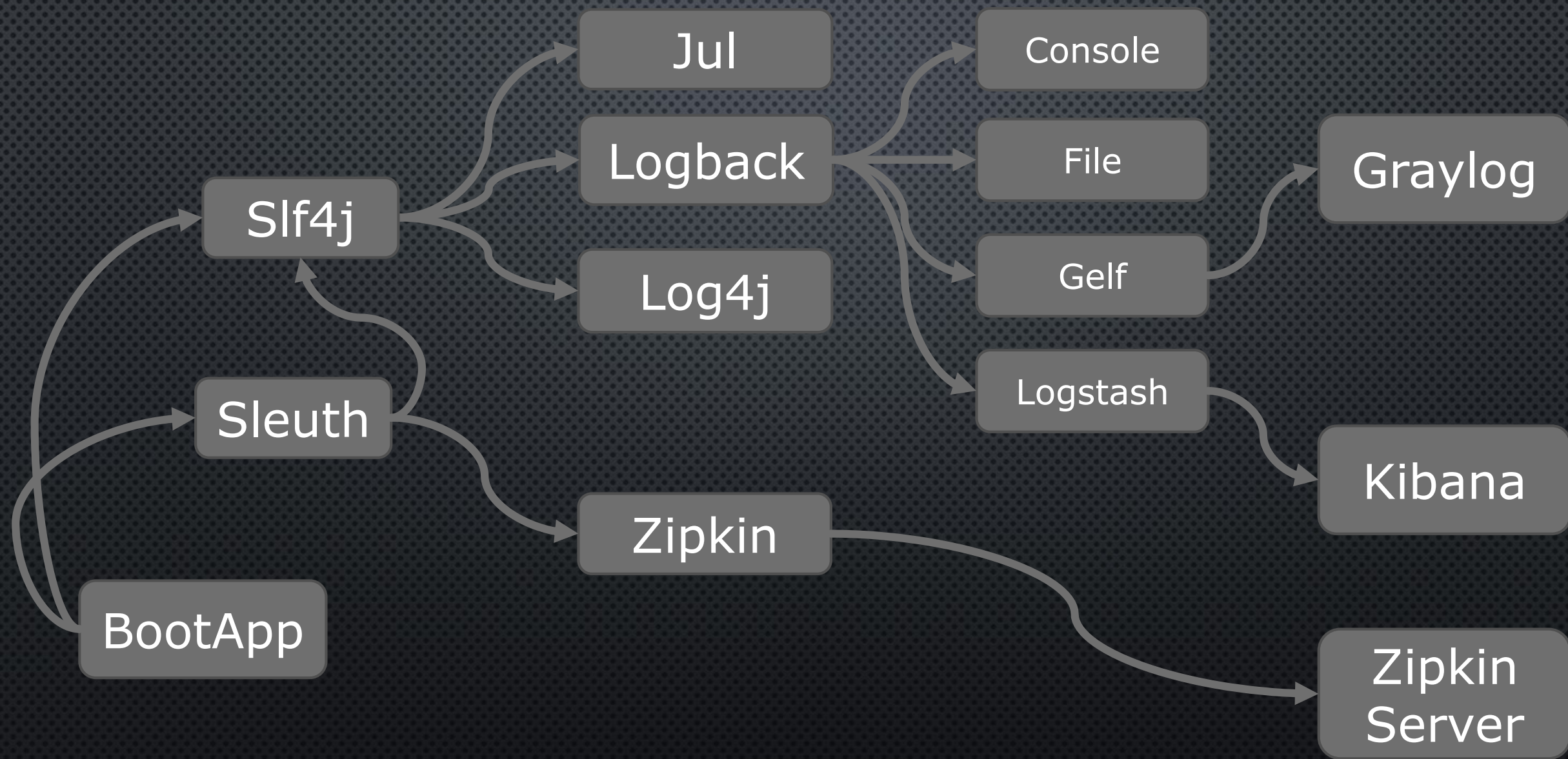




# Spring Zipkin

- :> Zipking traces from Sleuth logs
- :> Zipkin server and Web UI
- :> Traces collection via HTTP or Streams

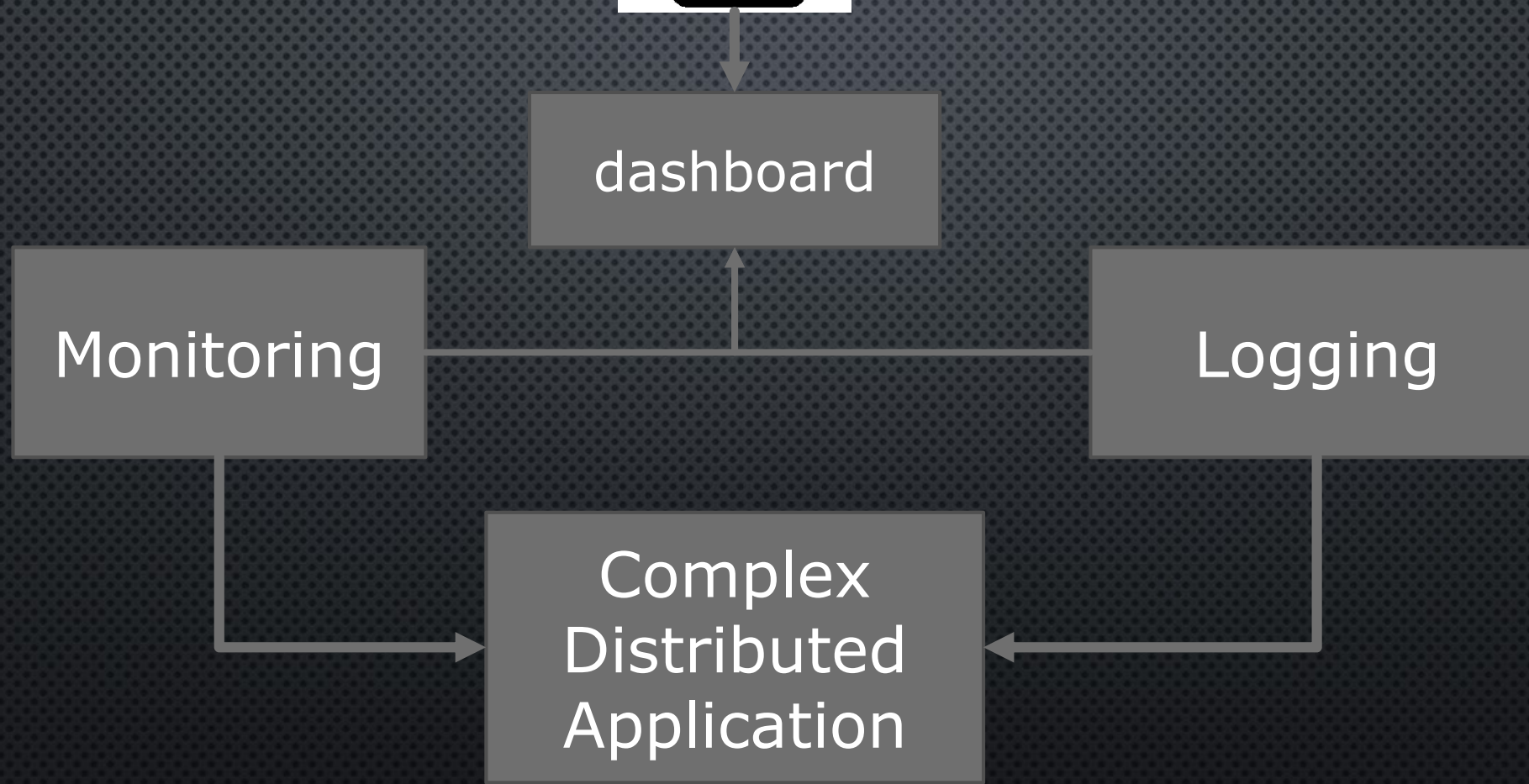




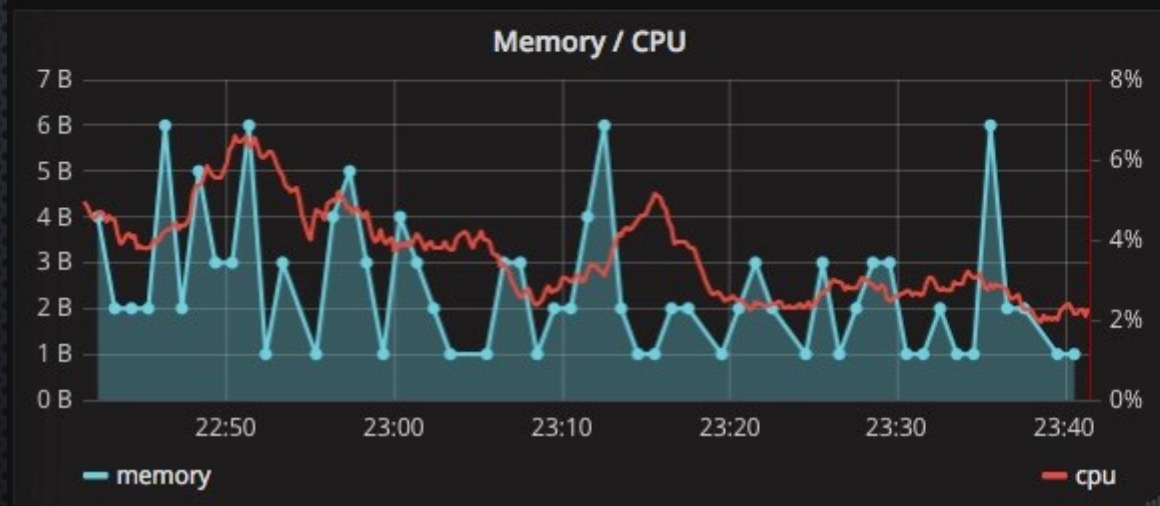
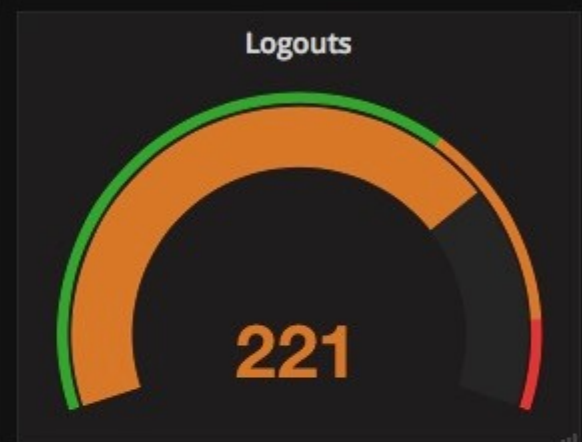
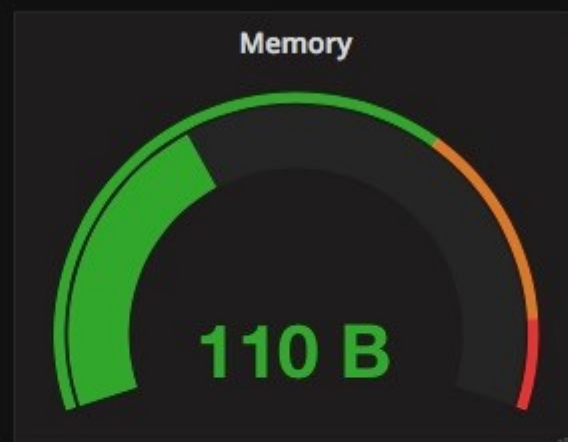
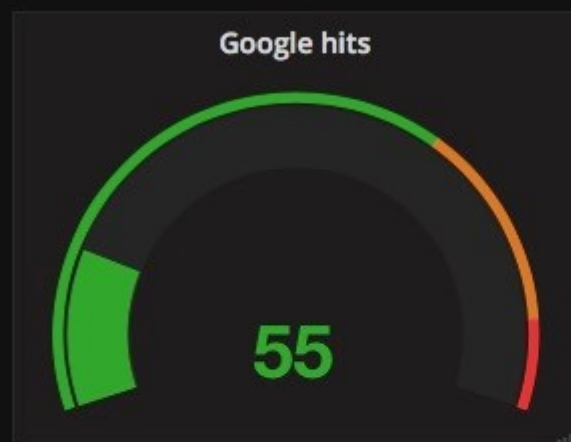




DEV-OPS









# Thanks for your attention!

[www.JavaDays.cz](http://www.JavaDays.cz)

[www.gopas.cz](http://www.gopas.cz)