SPRING BEST PRACTICES

GORAN ATANASOVSKI
NETCETERA
@gatanaso_

CONFIGURATION

Xml or Java

CONFIGURATION

- Java is type safe
- Xml configuration quickly grows
- Refactoring is safer and easier

CONFIGURATION

- Use annotations as much as possible
- Prefer component scanning
- @Configuration for special beans

DEPENDENCY INJECTION

Constructor or Setter Injection

DEPENDENCY INJECTION

- Use constructor based dependency injection
- If you absolutely must, then use setter injection
- Optional dependencies and defaults

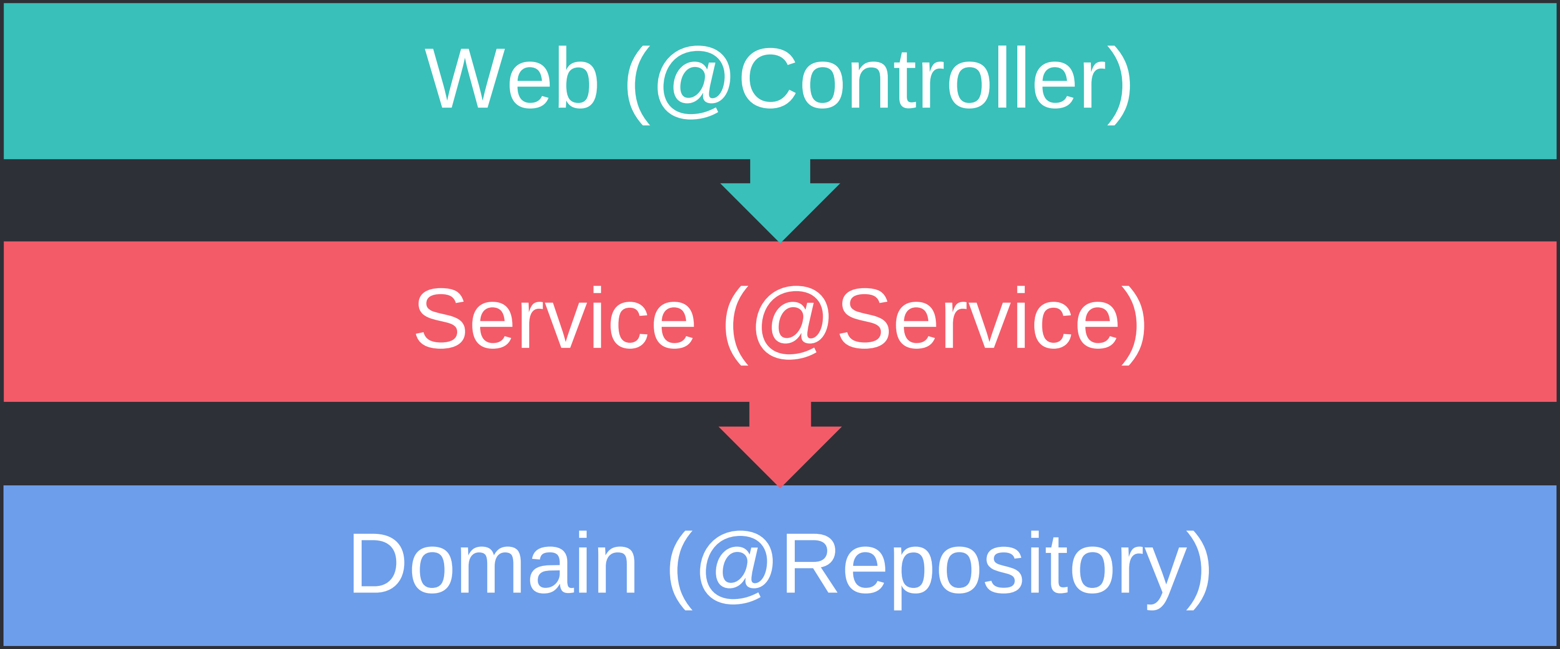
“Every time you use field injection, a unit test dies” – Josh Long, Spring Developer Advocate

LAYERING

Think architecture

LAYERING

- Separate layers
- Keep dependencies in one direction

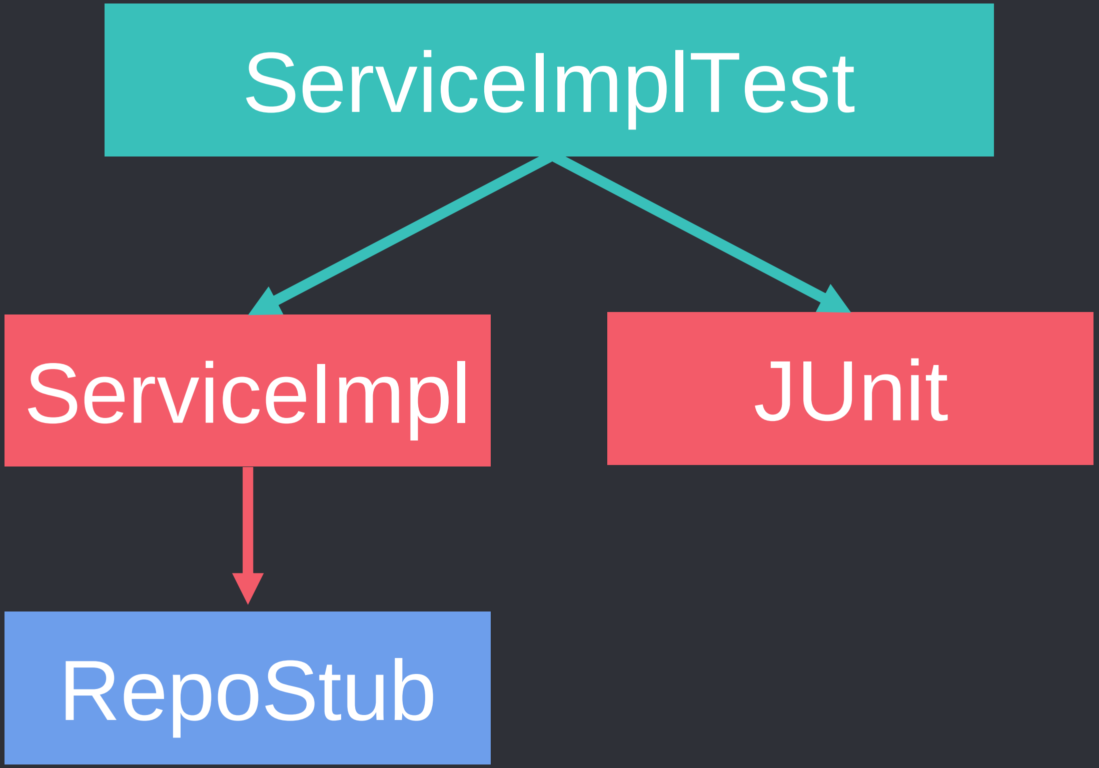


TESTING

Unit and Integration

TESTING

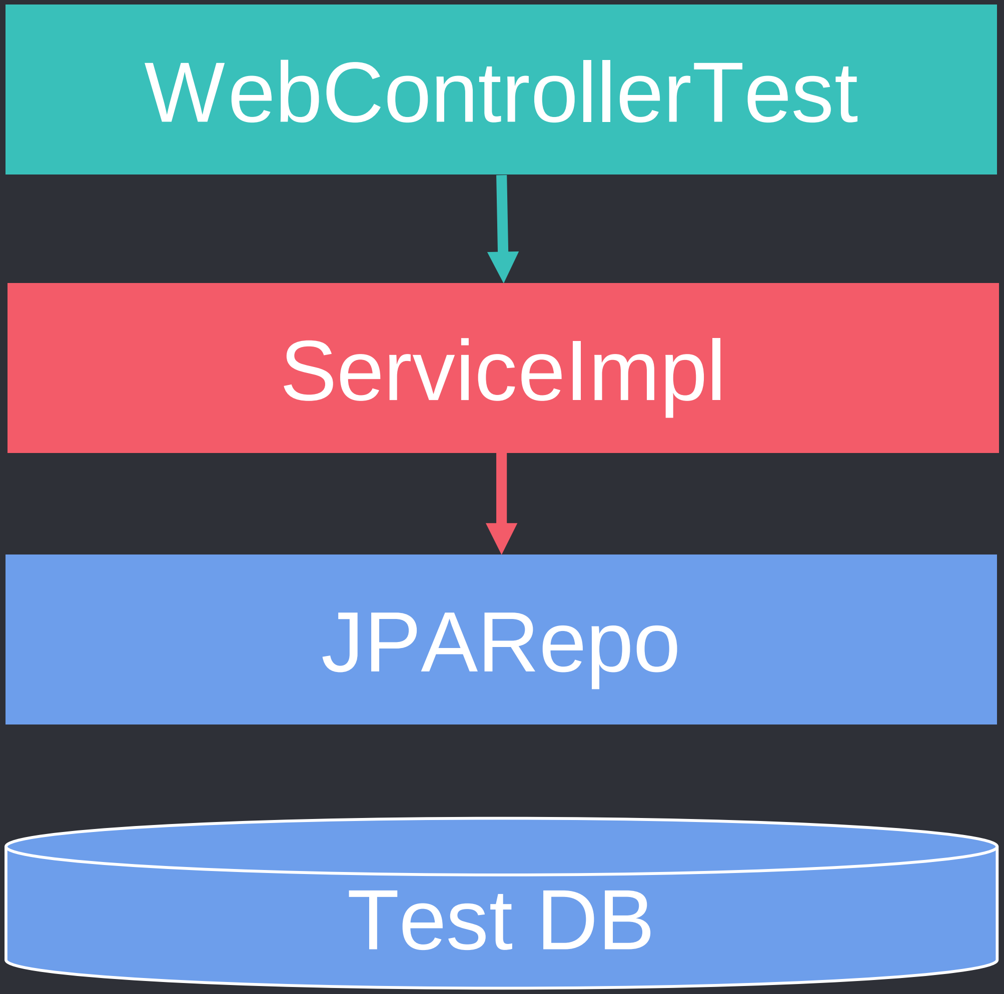
- Keep spring out of unit tests



TESTING

- Use spring for integration testing
- Infrastructure should be scaled down
- SpringJUnit4ClassRunner

TESTING

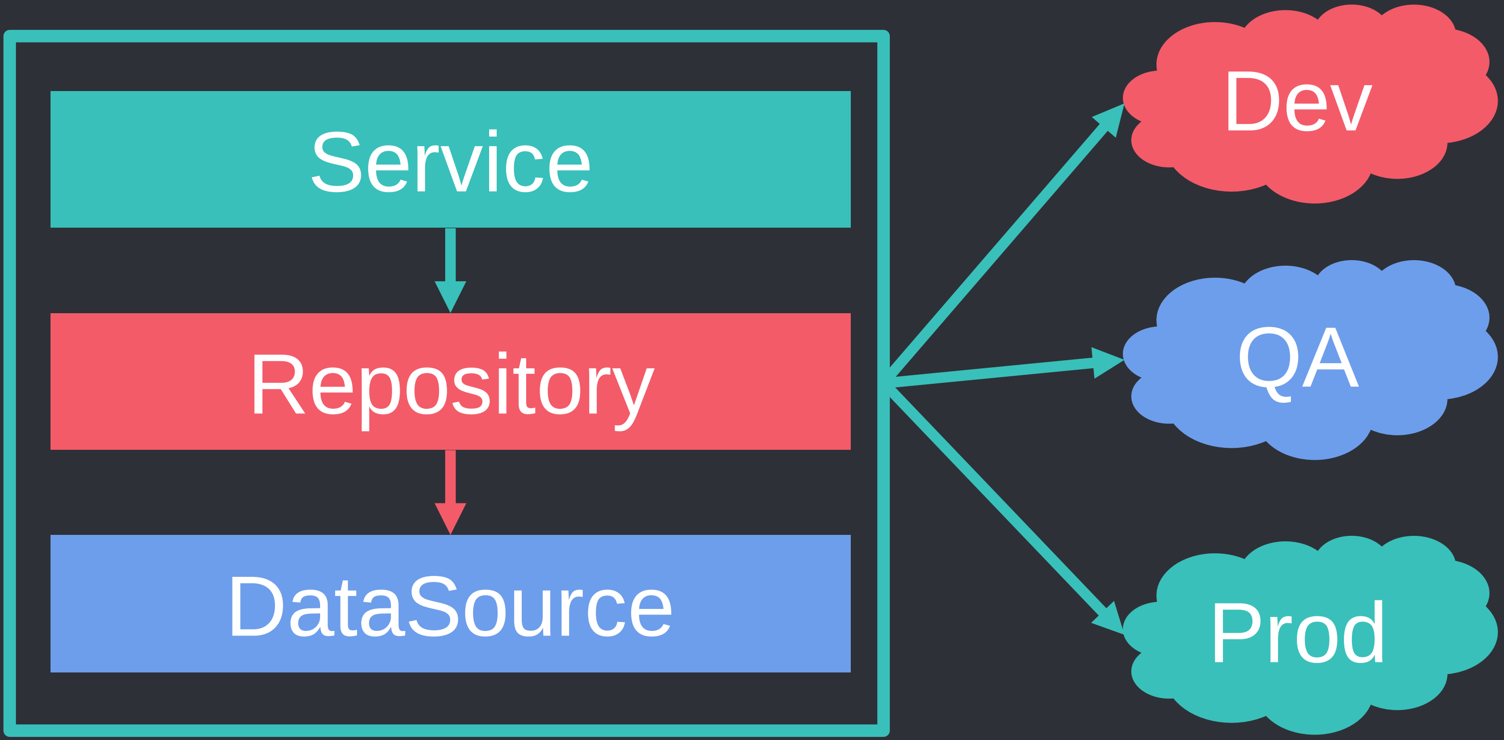


ENVIRONMENT

Profiles

ENVIRONMENT

- profile xml attribute
- @Profile annotation



SPRING TEMPLATES

JdbcTemplate, RestTemplate, JmsTemplate…

SPRING TEMPLATES

- Delegate the processing task to user implemented callbacks
- Don’t reinvent the wheel
- Remove boilerplate code and reduce errors

PERFORMANCE

Application context

PERFORMANCE

- Each bean is eagerly instantiated by default
- @Lazy initialization for expensive bean
- Limit component scanning

QUESTIONS?

THANK YOU!

@gatanaso_