Маппинг

использовать нельзя избегать

```java
@Value
@Builder(toBuilder = true)
public class ApplicationReportDto {
    UUID applicationId;
    UUID personId;
    String phone;
    String source;
    LocalDateTime createdAt;
    String type;
}
```

```java
@Value
@Builder(toBuilder = true)
public class ApplicationReportDto {
    UUID applicationId;
    UUID personId;
    String phone;
    String source;
    LocalDateTime createdAt;
    String type;
}
```

```java
@Value
@Builder(toBuilder = true)
public class ApplicationReportDto {
    UUID applicationId;
    UUID personId;
    String phone;
    String source;
    LocalDateTime createdAt;
    String type;
}
```

```java
public class ApplicationDto {
    UUID applicationId;
    UUID personId;
    String channel;
    LocalDateTime createdAt;
    String type;
}
```

```java
@Value
@Builder(toBuilder = true)
public class ApplicationReportDto {
    UUID applicationId;
    UUID personId;
    String phone;
    String source;
    LocalDateTime createdAt;
    String type;
}
```

```java
public class ApplicationApiBo {
    UUID applicationId;
    UUID personId;
    String channel;
    LocalDateTime createdAt;
    String phone;
    String type;
```

```java
public class ApplicationDto {
    UUID applicationId;
    UUID personId;
    String channel;
    LocalDateTime createdAt;
    String type;
}
```

```java
@Value
@Builder(toBuilder = true)
public class ApplicationReportDto {
    UUID applicationId;
    UUID personId;
    String phone;
    String source;
    LocalDateTime createdAt;
    String type;
}
```

```java
public class ApplicationApiBo {
    UUID applicationId;
    UUID personId;
    String channel;
    LocalDateTime createdAt;
    String phone;
    String type;
```

```java
public class ApplicationDto
    UUID applicationId;
    UUID personId;
    String channel;
    LocalDateTime createdAt
    String type;
}
```

```java
public class Application {
    @Id
    @GeneratedValue(generator = "uuid2")
    @GenericGenerator(name = "uuid2"
        , strategy = "uuid2")
    @Column(name = "application_id")
    UUID applicationId;

    @Column(name = "application_type"
        , insertable = false
        , updatable = false)
    String type;

    @Column(name = "person_id")
    UUID personId;

    @Column(name = "channel")
    String channel;

    @Column(name = "created_at")
    LocalDateTime createdAt;

    @Column(name = "phone")
    String phone;
```

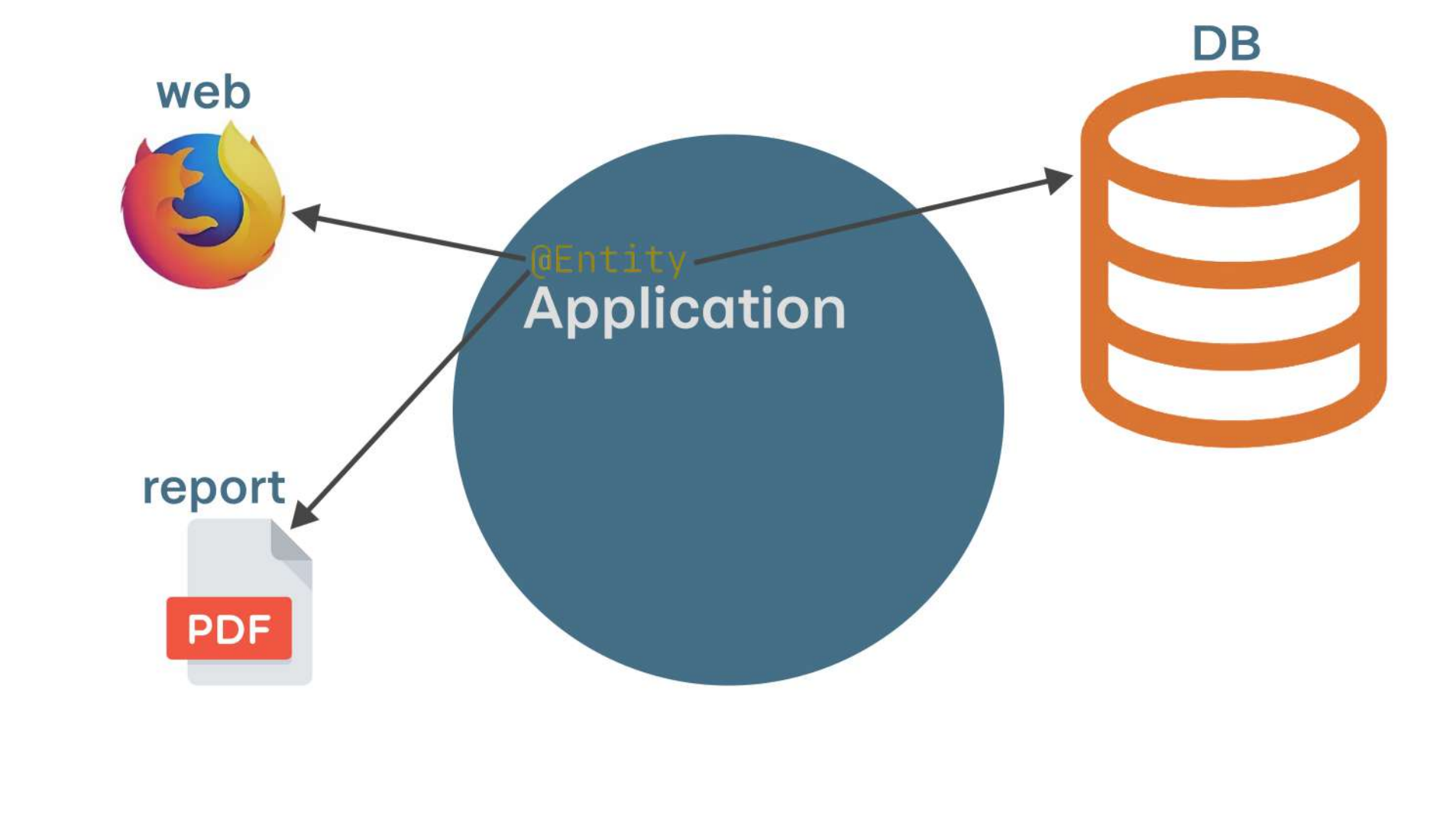
```java
@Value
@Builder(toBuilder = true)
public class ApplicationReportDto {
    UUID applicationId;
    UUID personId;
    String phone;
    String source;
    LocalDateTime createdAt;
    String type;
}
```

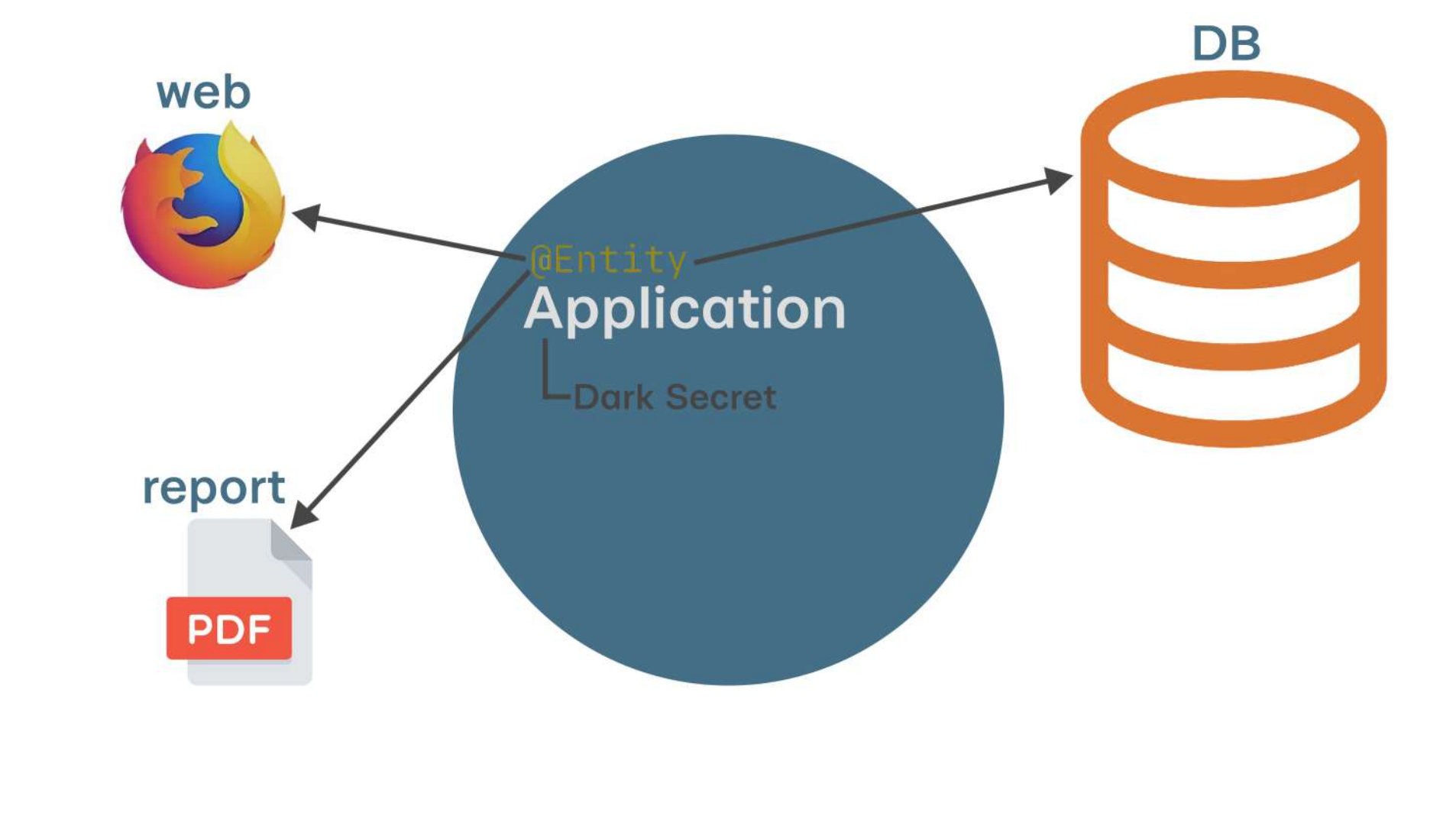
```java
public class ApplicationApiBo {
    UUID applicationId;
    UUID personId;
    String channel;
    LocalDateTime createdAt;
    String phone;
    String type;
```

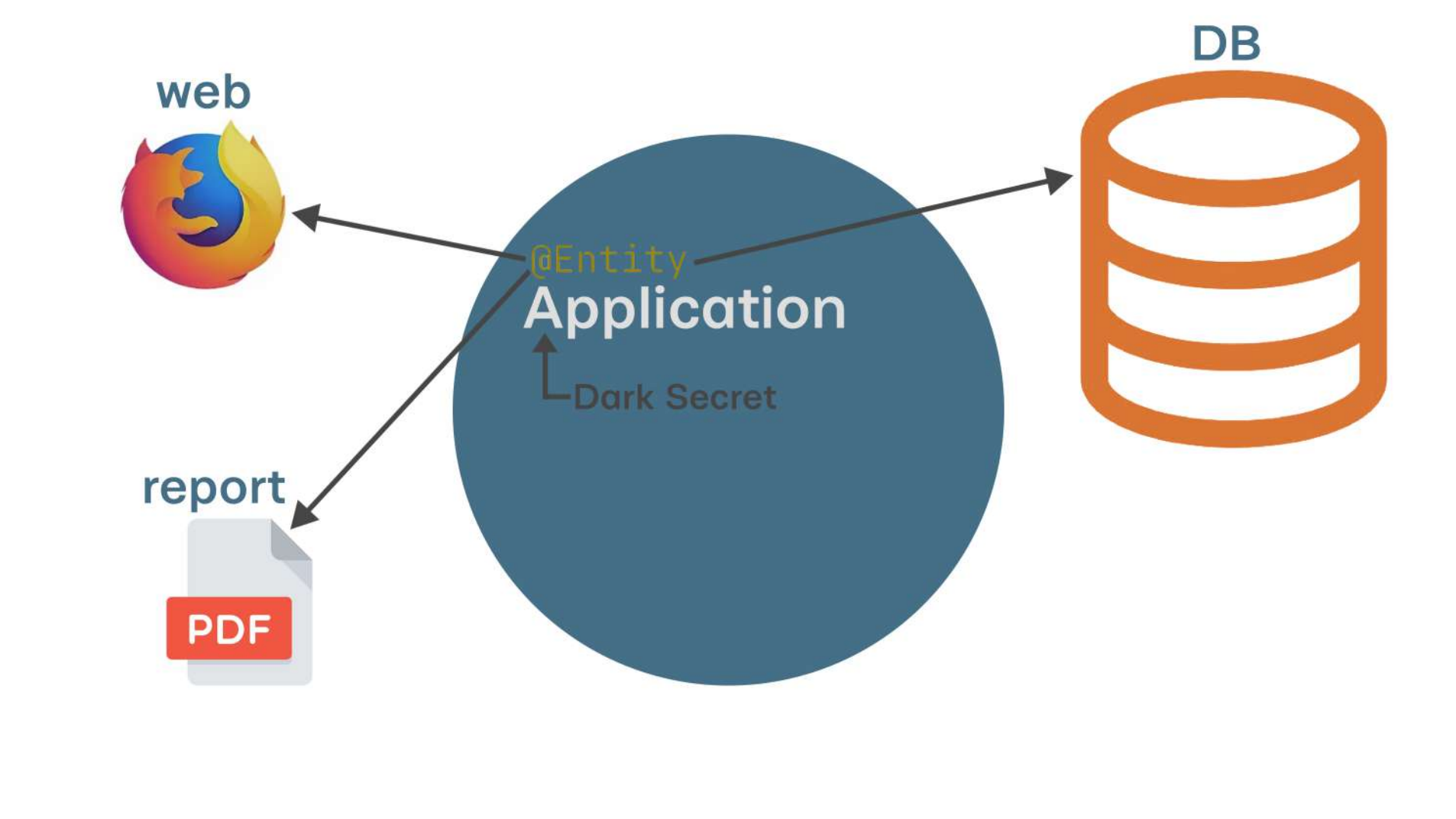
```java
public class Application {
    @Id
    @GeneratedValue(generator = "uuid2")
    @GenericGenerator(name = "uuid2"
        , strategy = "uuid2")
    @Column(name = "application_id")
    UUID applicationId;

    @Column(name = "application_type"
        , insertable = false
        , updatable = false)
    String type;

    @Column(name = "person_id")
    UUID personId;

    @Column(name = "channel")
    String channel;

    @Column(name = "created_at")
    LocalDateTime createdAt;

    @Column(name = "phone")
    String phone;
```

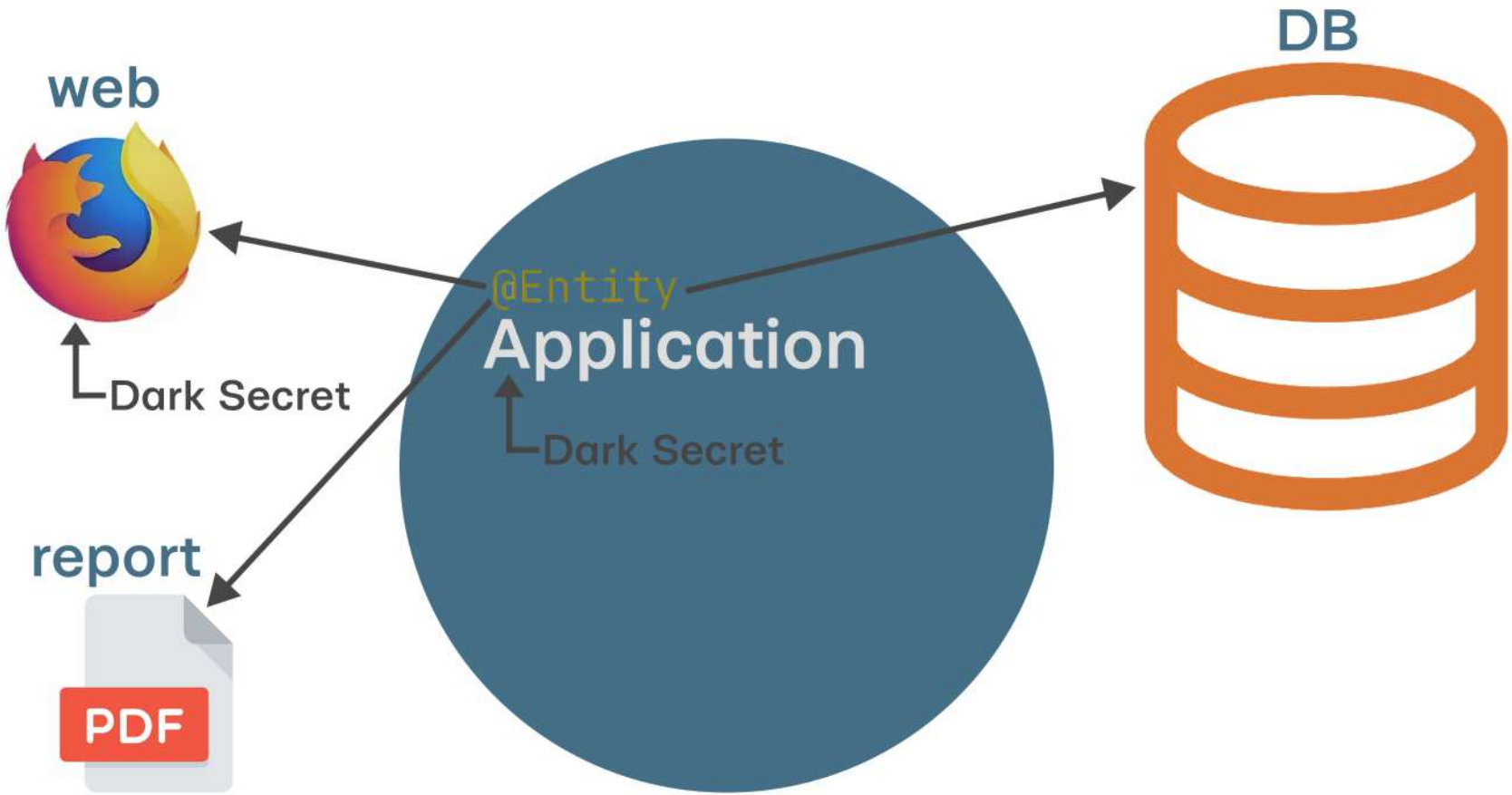

```java
@Value
@Builder(toBuilder = true)
public class ApplicationReportDto {
    UUID applicationId;
    UUID personId;
    String phone;
    String source;
    LocalDateTime createdAt;
    String type;
}
```

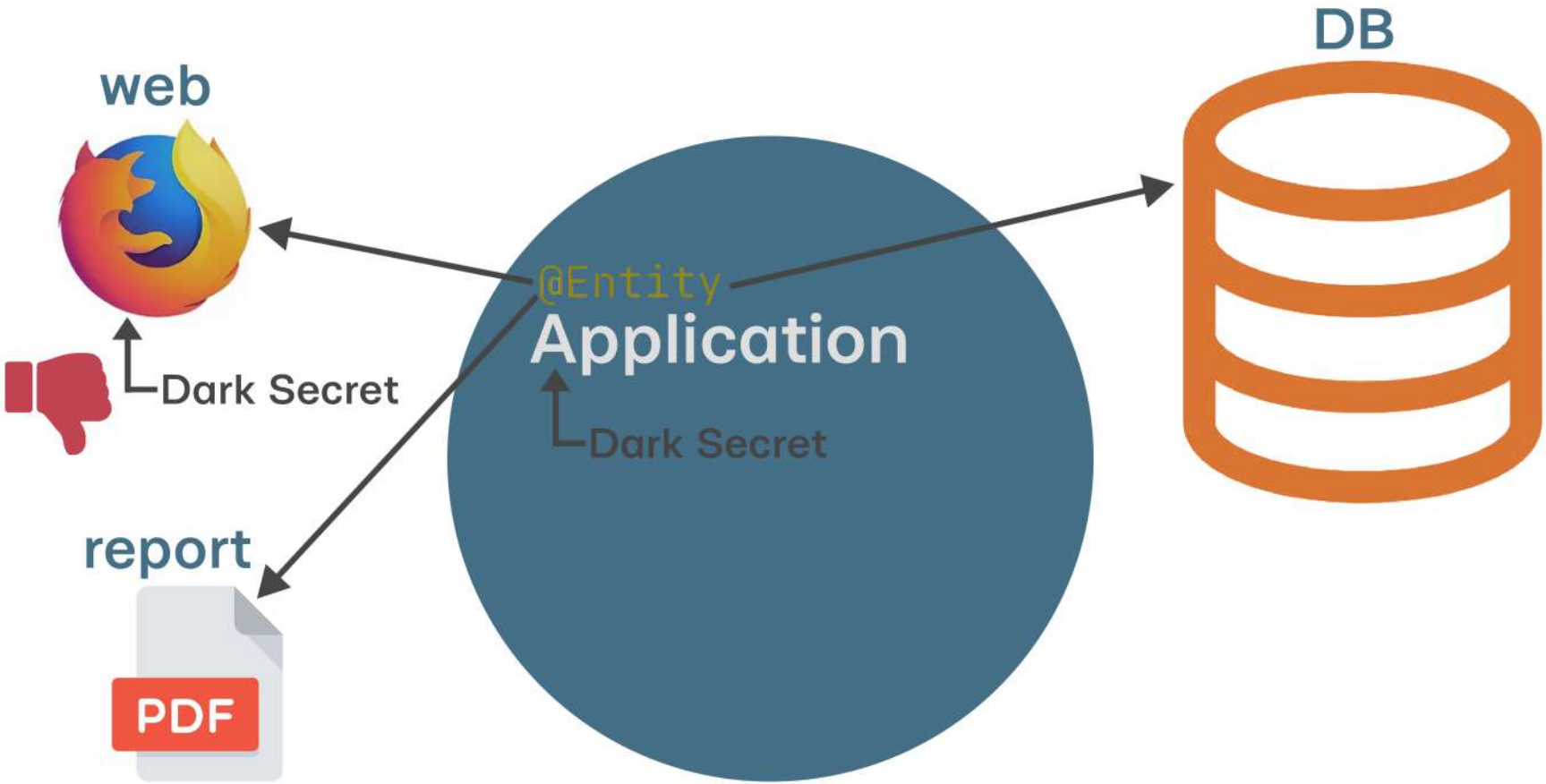

```java
public class ApplicationDto
    UUID applicationId;
    UUID personId;
    String channel;
    LocalDateTime createdAt
    String type;
}
```

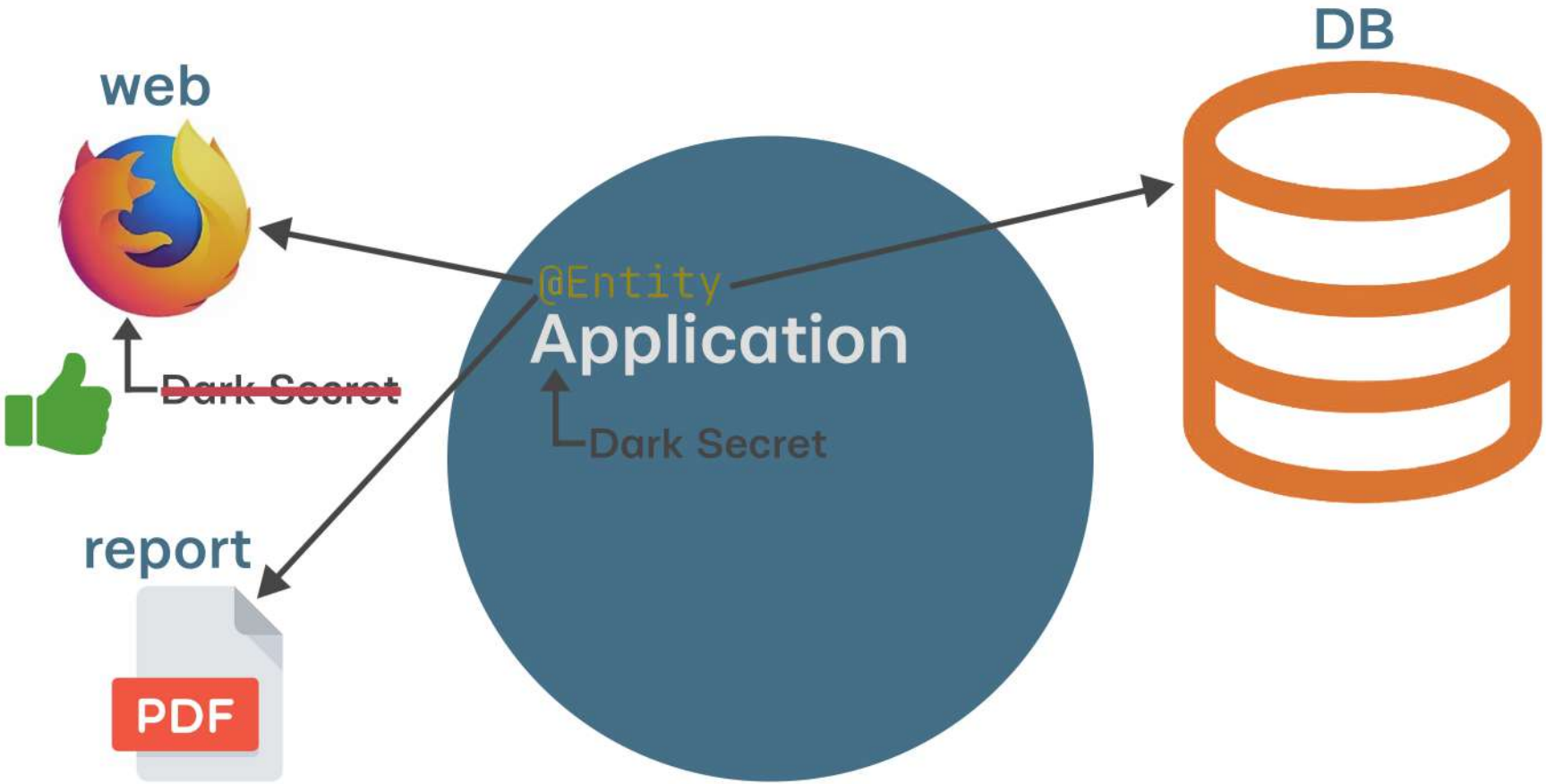

DB
web
@Entity
Application
report
PDF


web
DB
@Entity
Application
Dark Secret
report
PDF


web
DB
@Entity
Application
Dark Secret
report
PDF


DB
web
@Entity
Application
Dark Secret
Dark Secret
report
PDF


DB
web
@Entity
Application
Dark Secret
Dark Secret
report
PDF


DB
web
@Entity
Application
Dark Secret
Dark Secret
report
PDF

ApplicationDto ← @Entity Application

Dark Secret

web

@JsonSerialize
ApplicationDto

@Entity
Application

Dark Secret

DB

А если никаких секретов нет?

web

@JsonSerialize
ApplicationDto

@Entity
Application

Dark Secret

DB

А если никаких секретов нет?

Есть преобразование в JSON

web

@JsonSerialize
ApplicationDto

@Entity
Application

Dark Secret

DB

А если никаких секретов нет?

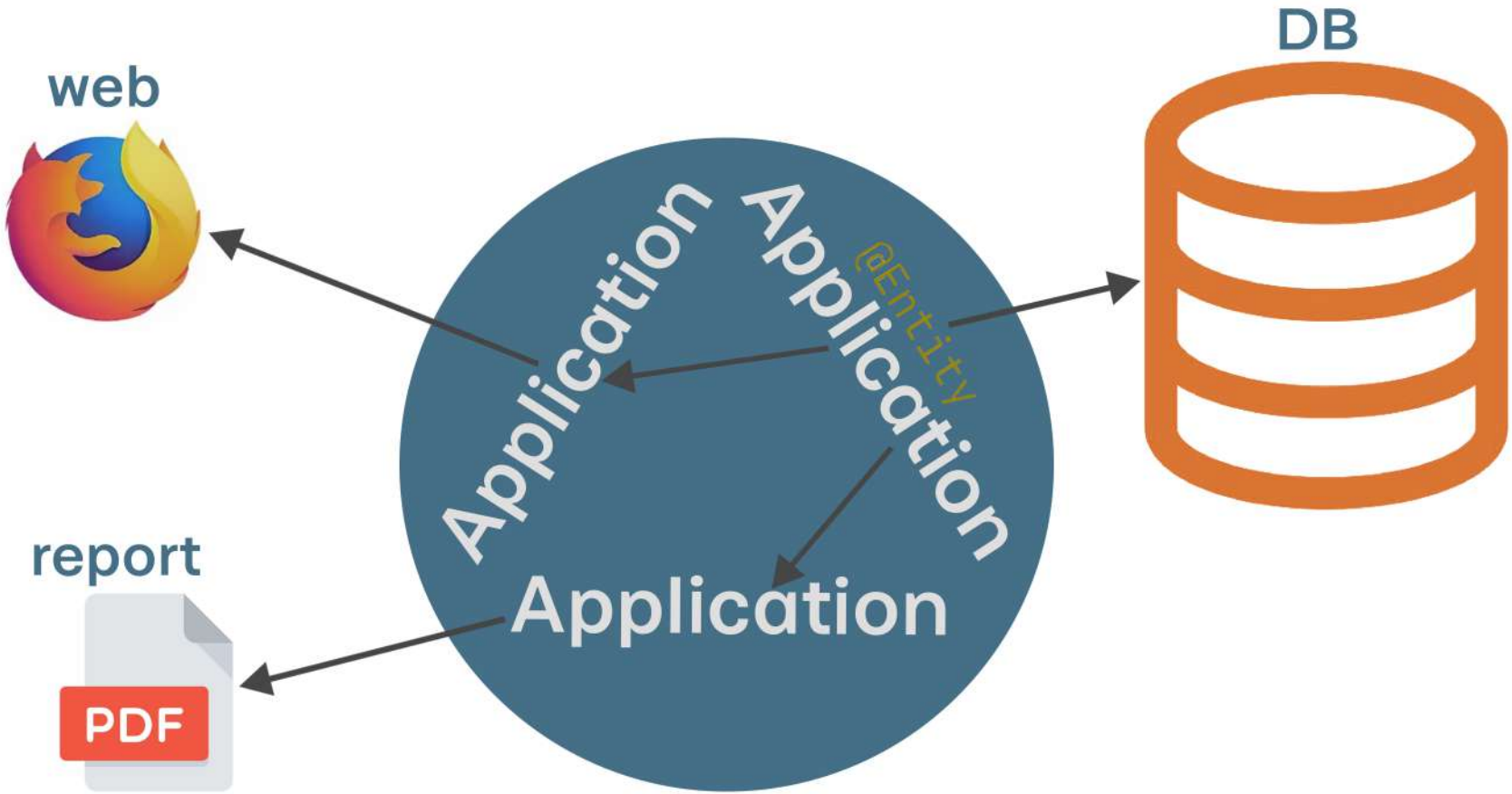
Есть преобразование в JSON

1) ApplicationDto нужен, чтобы контролировать какие данные уйдут наружу
2) ApplicationDto нужен, чтобы указать правила сериализации

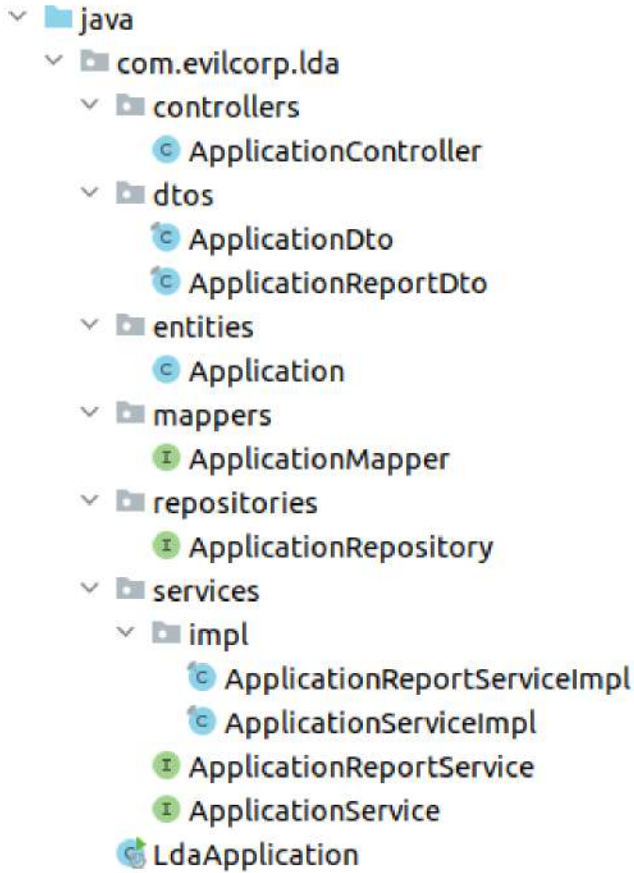

Делать Dto
Не делать Dto


Делать
Dto
Не делать
Dto

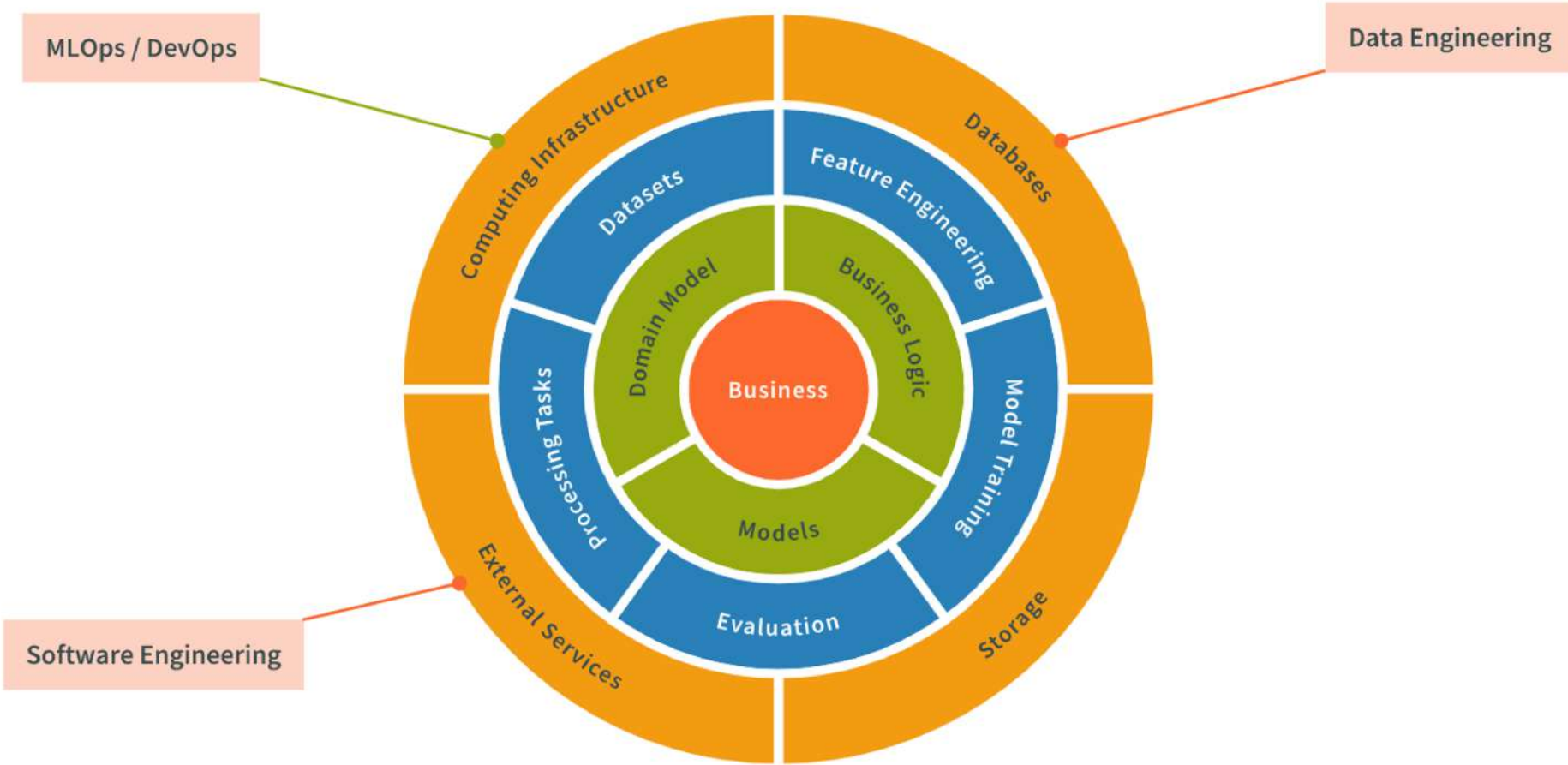

Делать
Dto
Не делать
Dto

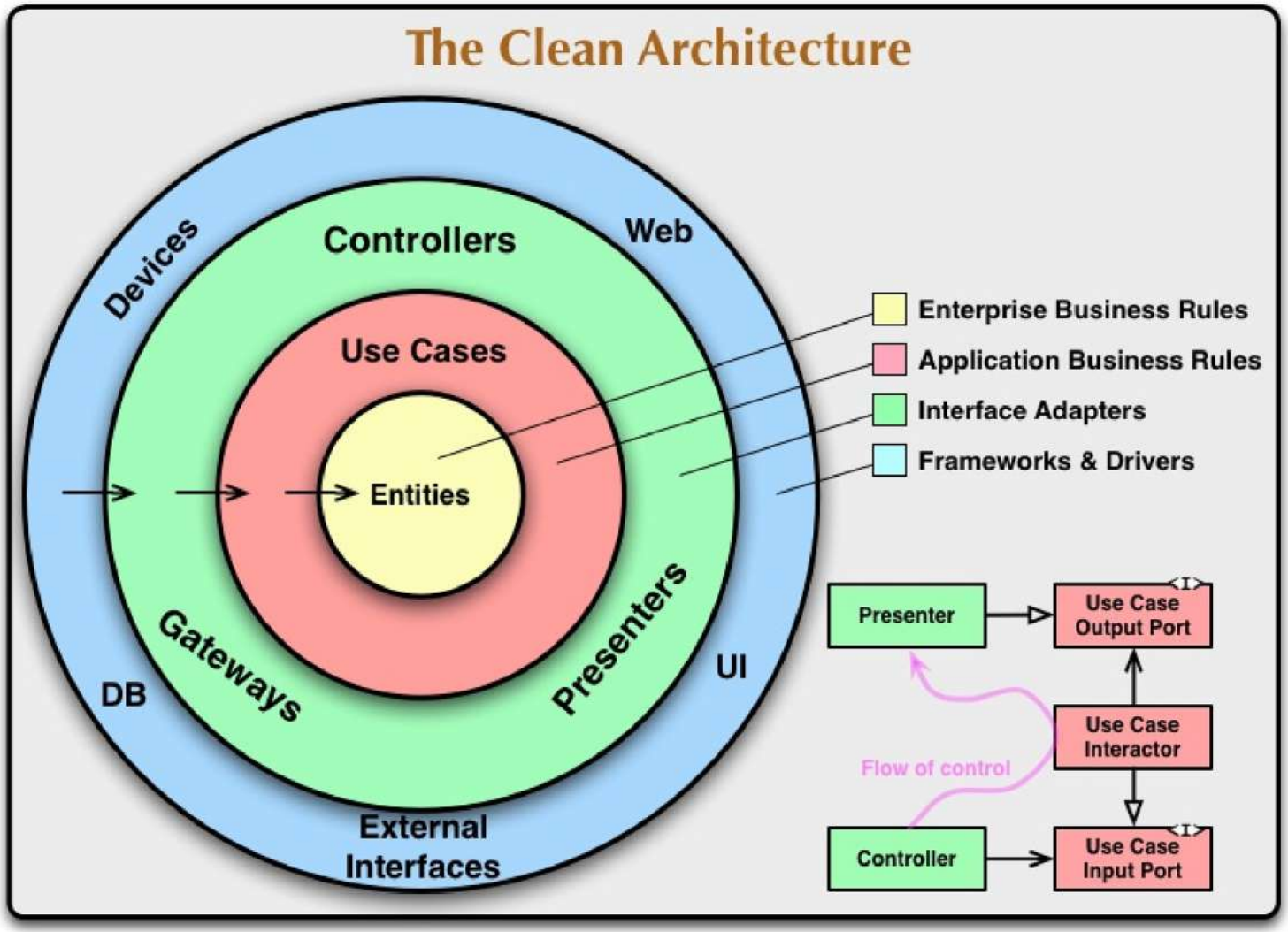

web
DB
report
PDF
Application
Application
@Entity
Application

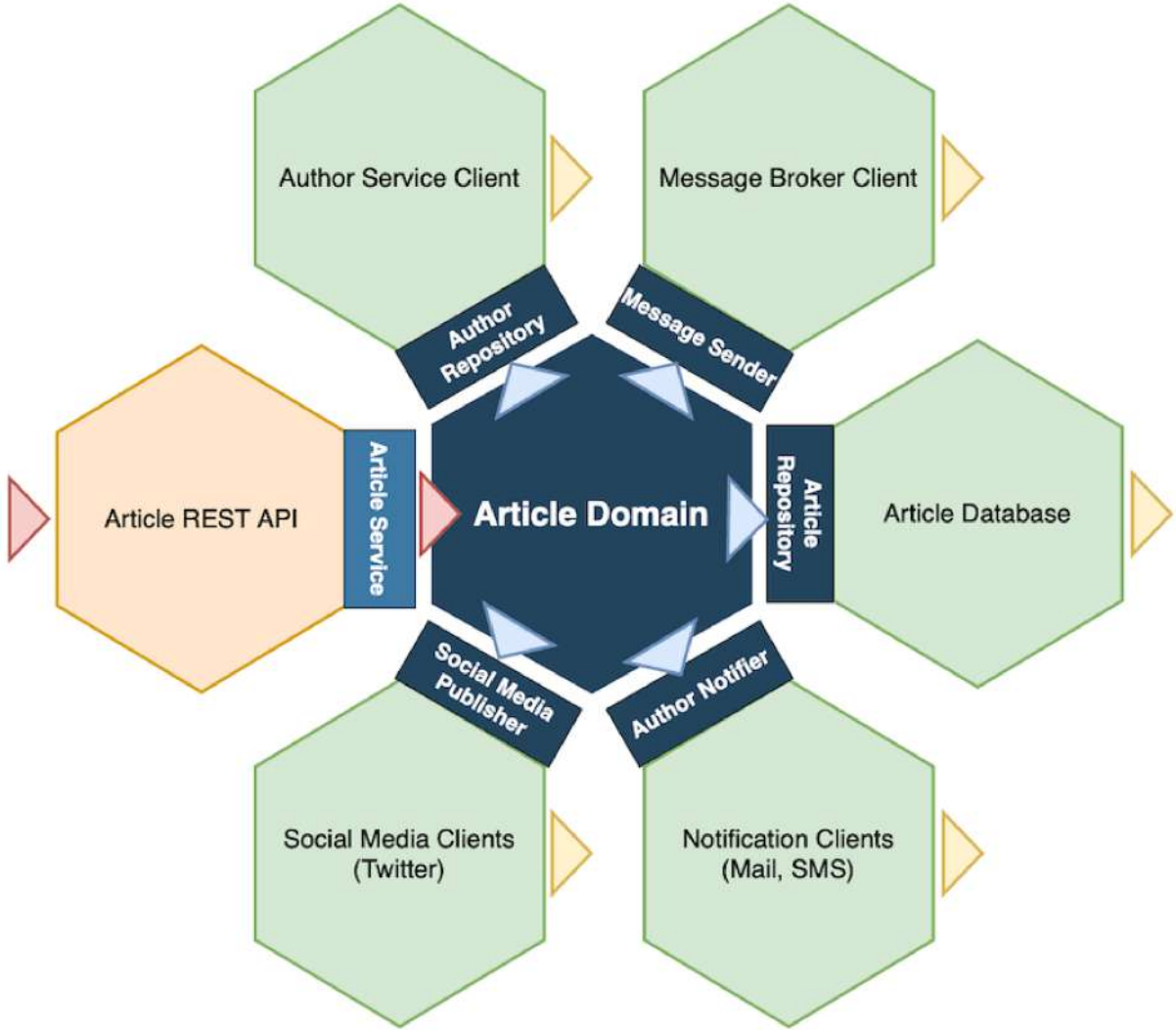

Стихийная архитектура
web
DB
report
PDF
ApplicationDto
-inn
@Entity
Application
ApplicationRep
-description

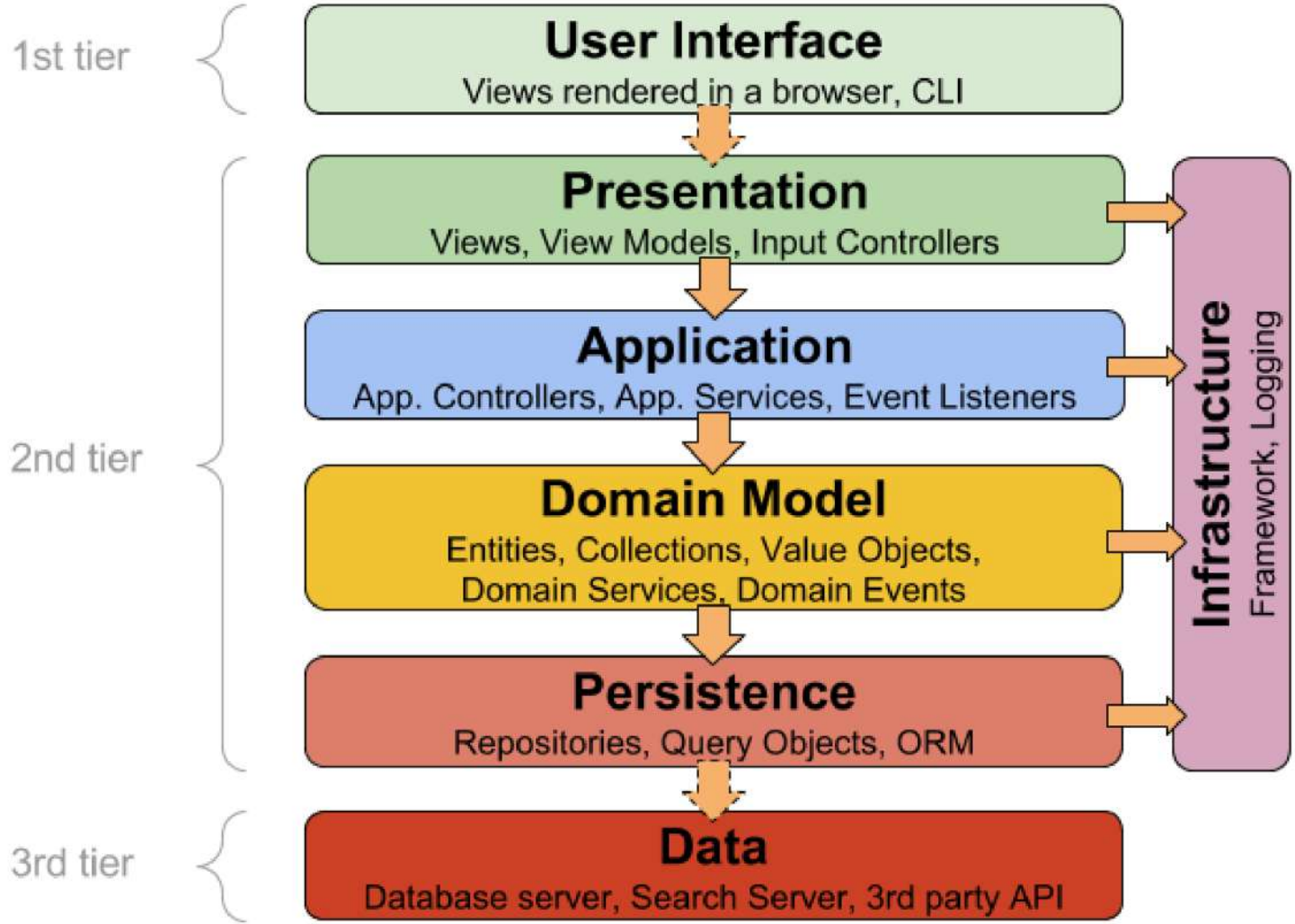

java
com.evilcorp.lda
controllers
ApplicationController
dtos
ApplicationDto
ApplicationReportDto
entities
Application
mappers
ApplicationMapper
repositories
ApplicationRepository
services
impl
ApplicationReportServiceImpl
ApplicationServiceImpl
ApplicationReportService
ApplicationService
LdaApplication

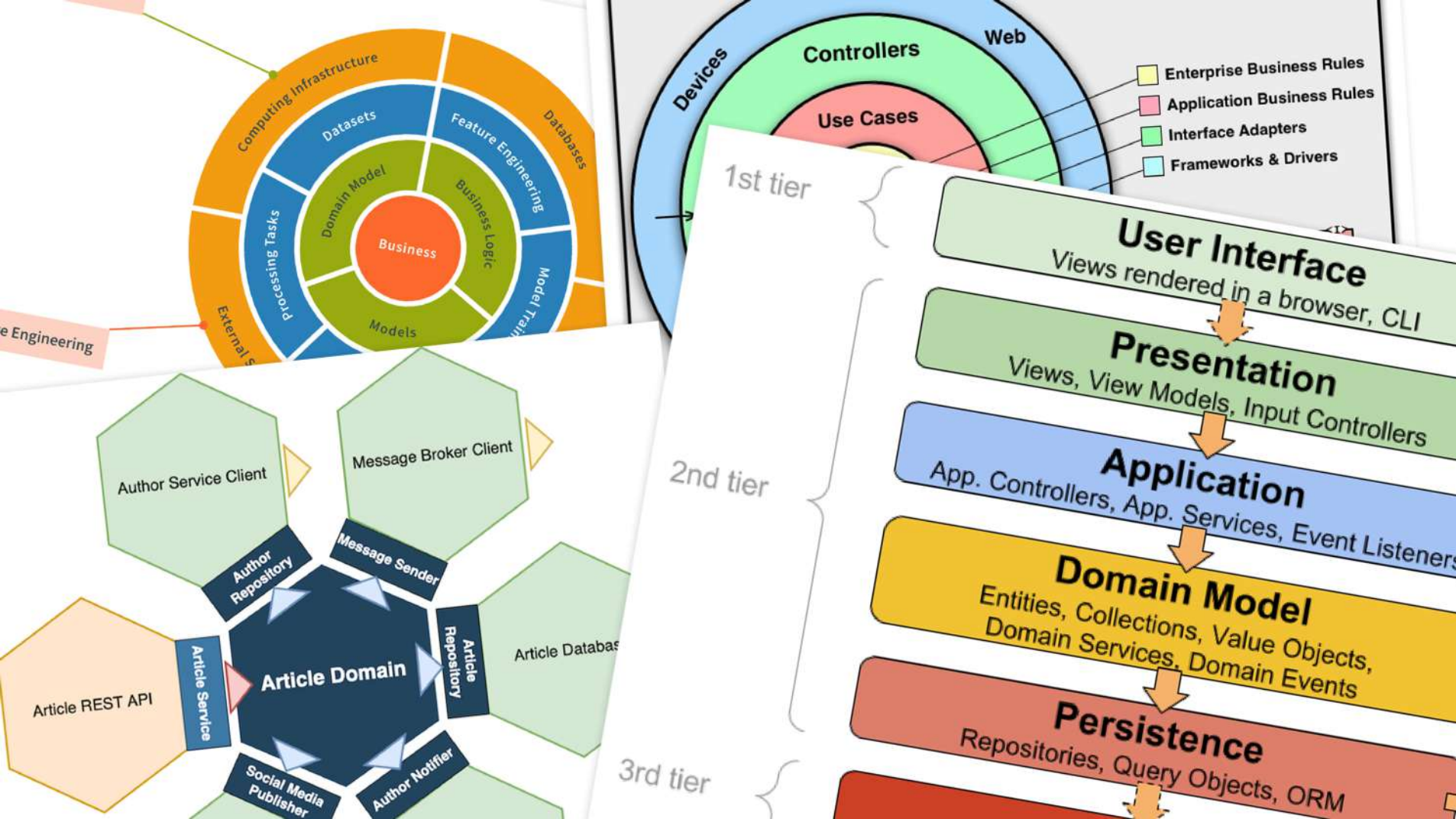
- java
  - com.evilcorp.lda
    - controllers
      - ApplicationController
    - dtos
      - ApplicationDto
      - ApplicationReportDto
    - entities
      - Application
    - mappers
      - ApplicationMapper
    - repositories
      - ApplicationRepository
    - services
      - impl
        - ApplicationReportServiceImpl
        - ApplicationServiceImpl
      - ApplicationReportService
      - ApplicationService
    - LdaApplication

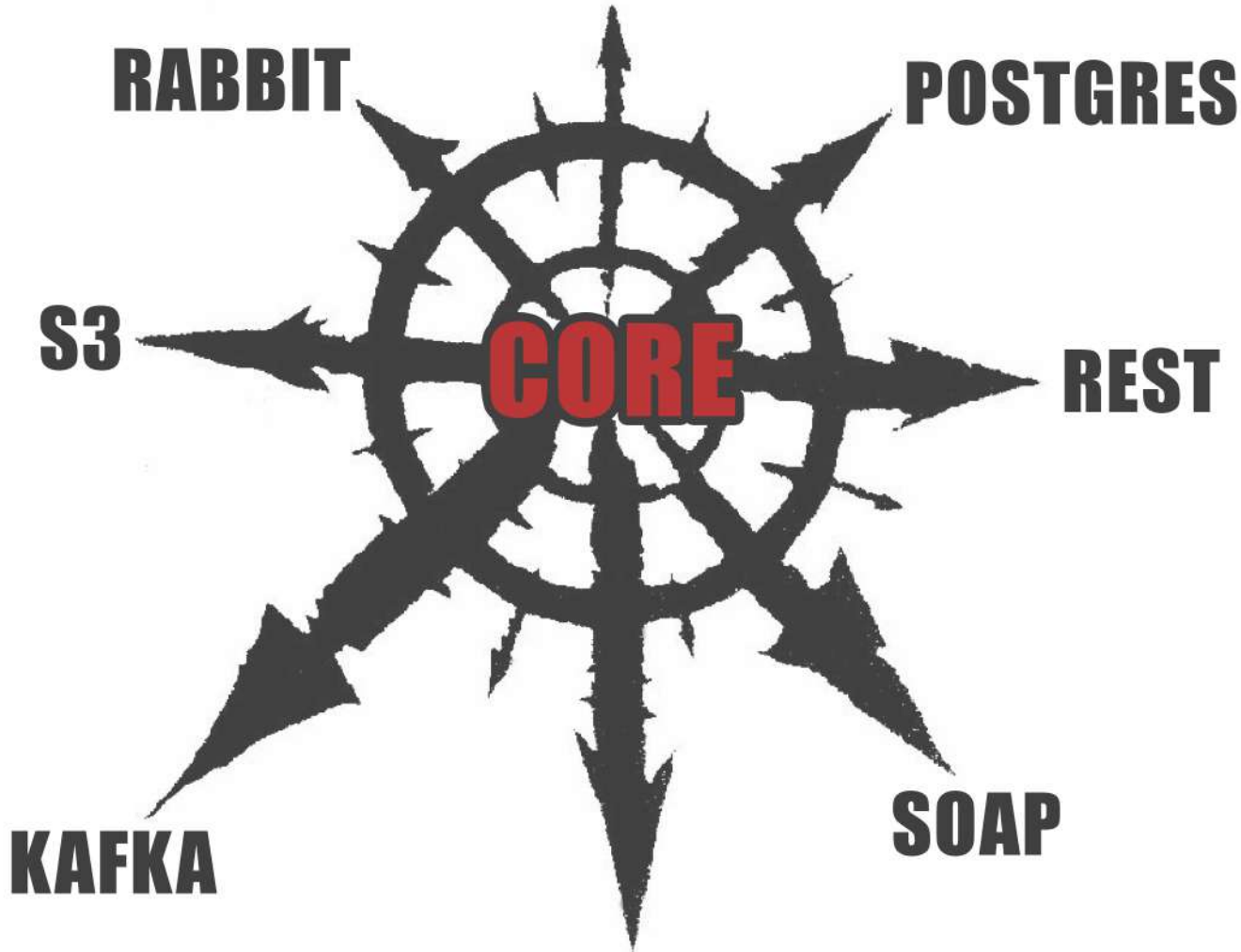

MLOps / DevOps
Data Engineering
Software Engineering
Computing Infrastructure
Databases
External Services
Storage
Datasets
Feature Engineering
Processing Tasks
Model Training
Domain Model
Business Logic
Models
Evaluation
Business


The Clean Architecture
Devices
Controllers
Web
Use Cases
Entities
Gateways
Presenters
DB
UI
External
Interfaces
Enterprise Business Rules
Application Business Rules
Interface Adapters
Frameworks & Drivers
Presenter
Use Case
Output Port
<I>
Use Case
Interactor
Flow of control
Controller
Use Case
Input Port
<I>

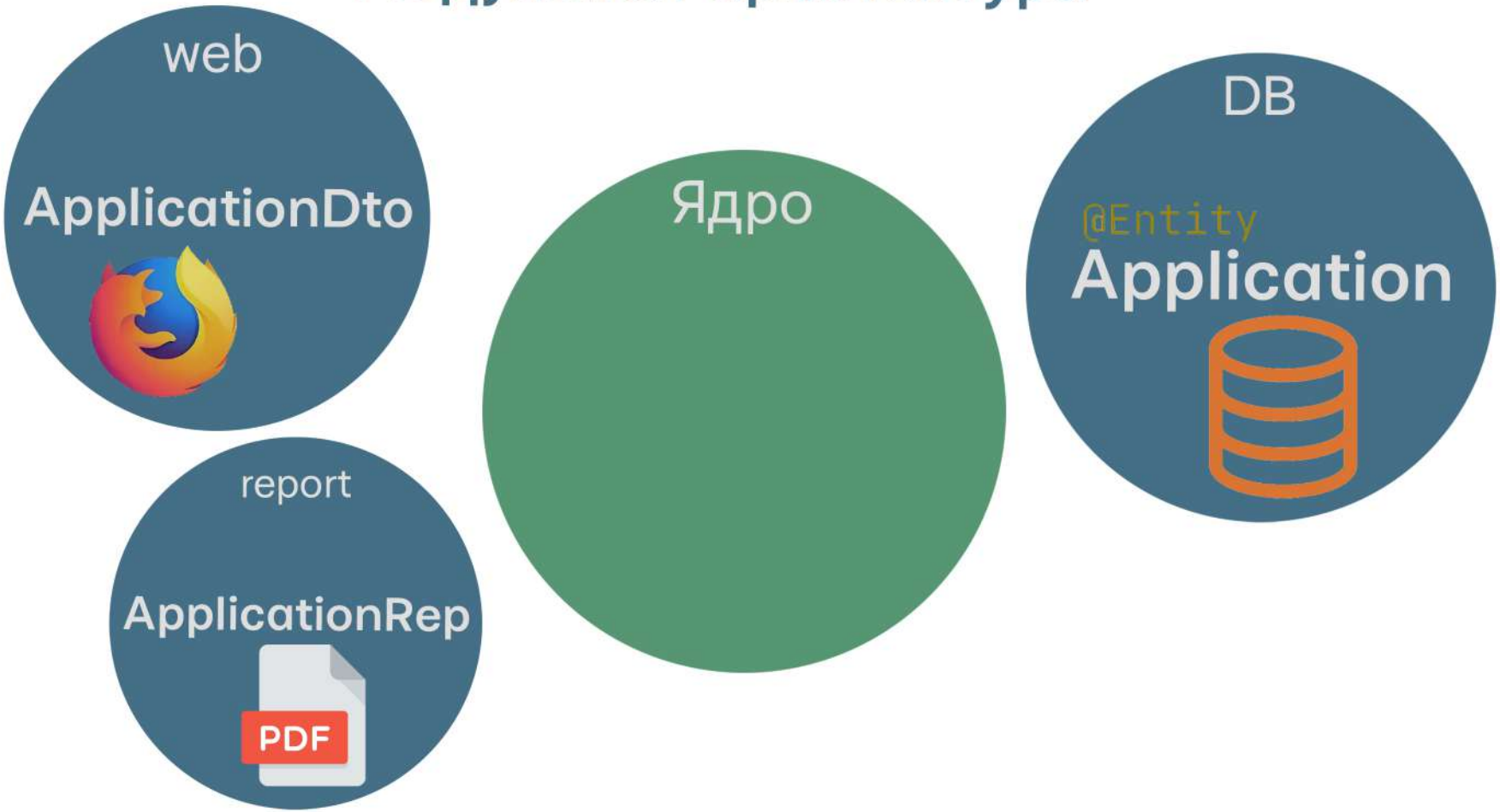

Author Service Client
Message Broker Client
Author Repository
Message Sender
Article REST API
Article Service
Article Domain
Article Repository
Article Database
Social Media Publisher
Author Notifier
Social Media Clients (Twitter)
Notification Clients (Mail, SMS)


1st tier
User Interface
Views rendered in a browser, CLI
2nd tier
Presentation
Views, View Models, Input Controllers
Application
App. Controllers, App. Services, Event Listeners
Domain Model
Entities, Collections, Value Objects,
Domain Services, Domain Events
Persistence
Repositories, Query Objects, ORM
Infrastructure
Framework, Logging
3rd tier
Data
Database server, Search Server, 3rd party API

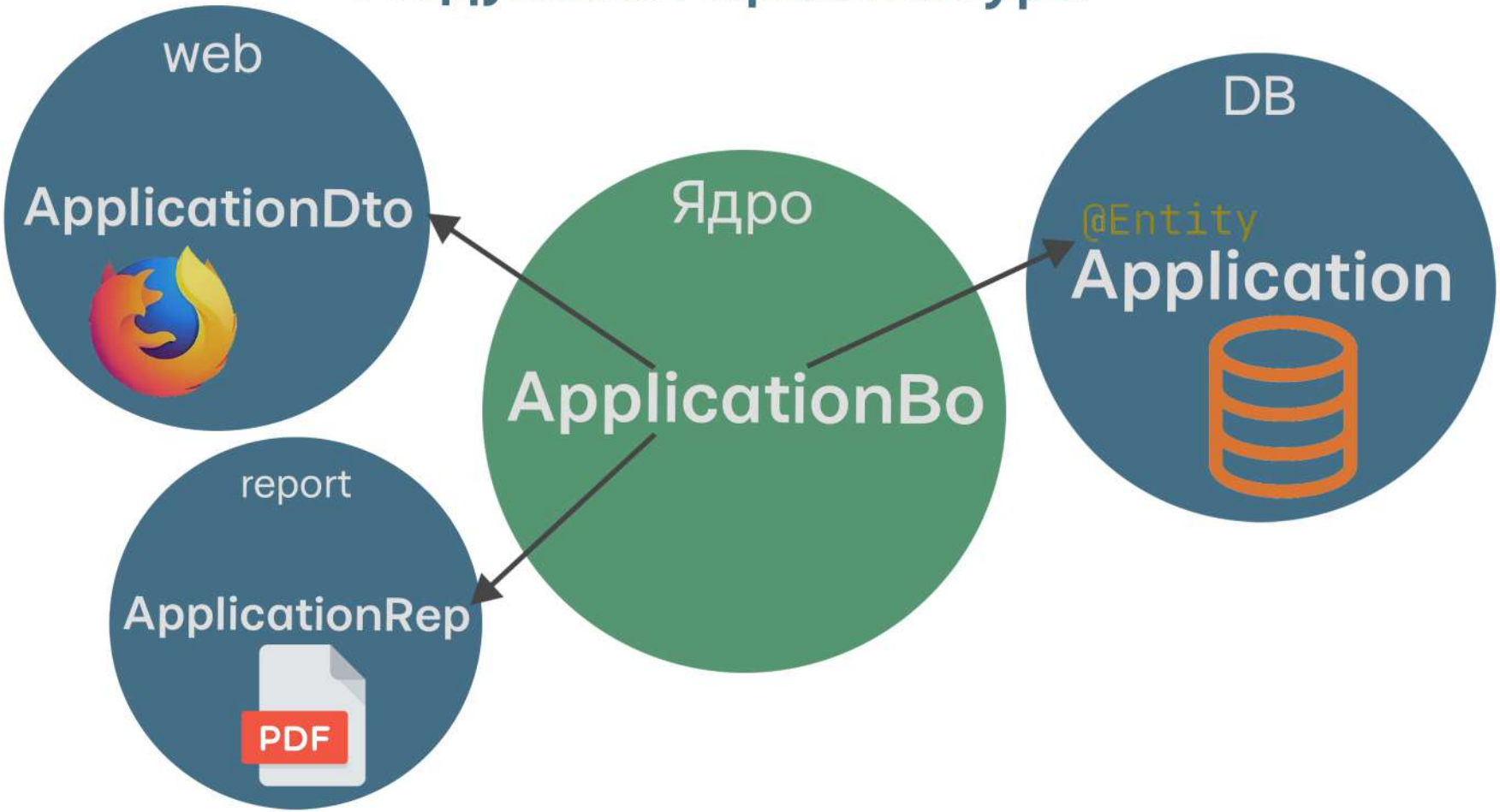

Computing Infrastructure
Databases
Datasets
Feature Engineering
Domain Model
Business Logic
Business
Processing Tasks
Models
Model Train
External S
e Engineering
Devices
Controllers
Web
Use Cases
Enterprise Business Rules
Application Business Rules
Interface Adapters
Frameworks & Drivers
Author Service Client
Message Broker Client
Author Repository
Message Sender
Article Repository
Article Databas
Article REST API
Article Service
Article Domain
Social Media Publisher
Author Notifier
1st tier
2nd tier
3rd tier
User Interface
Views rendered in a browser, CLI
Presentation
Views, View Models, Input Controllers
Application
App. Controllers, App. Services, Event Listeners
Domain Model
Entities, Collections, Value Objects,
Domain Services, Domain Events
Persistence
Repositories, Query Objects, ORM

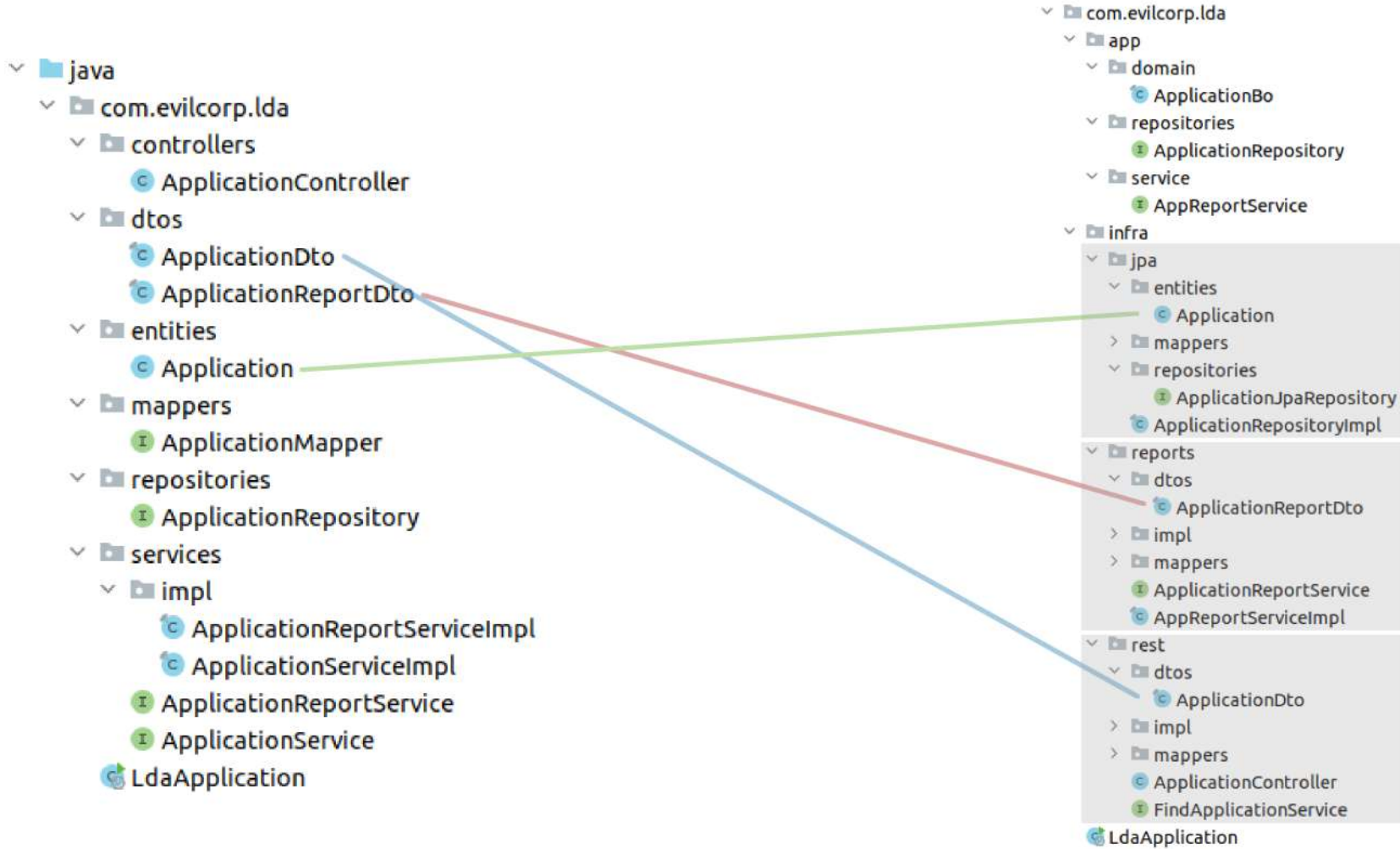

RABBIT
POSTGRES
S3
CORE
REST
KAFKA
SOAP

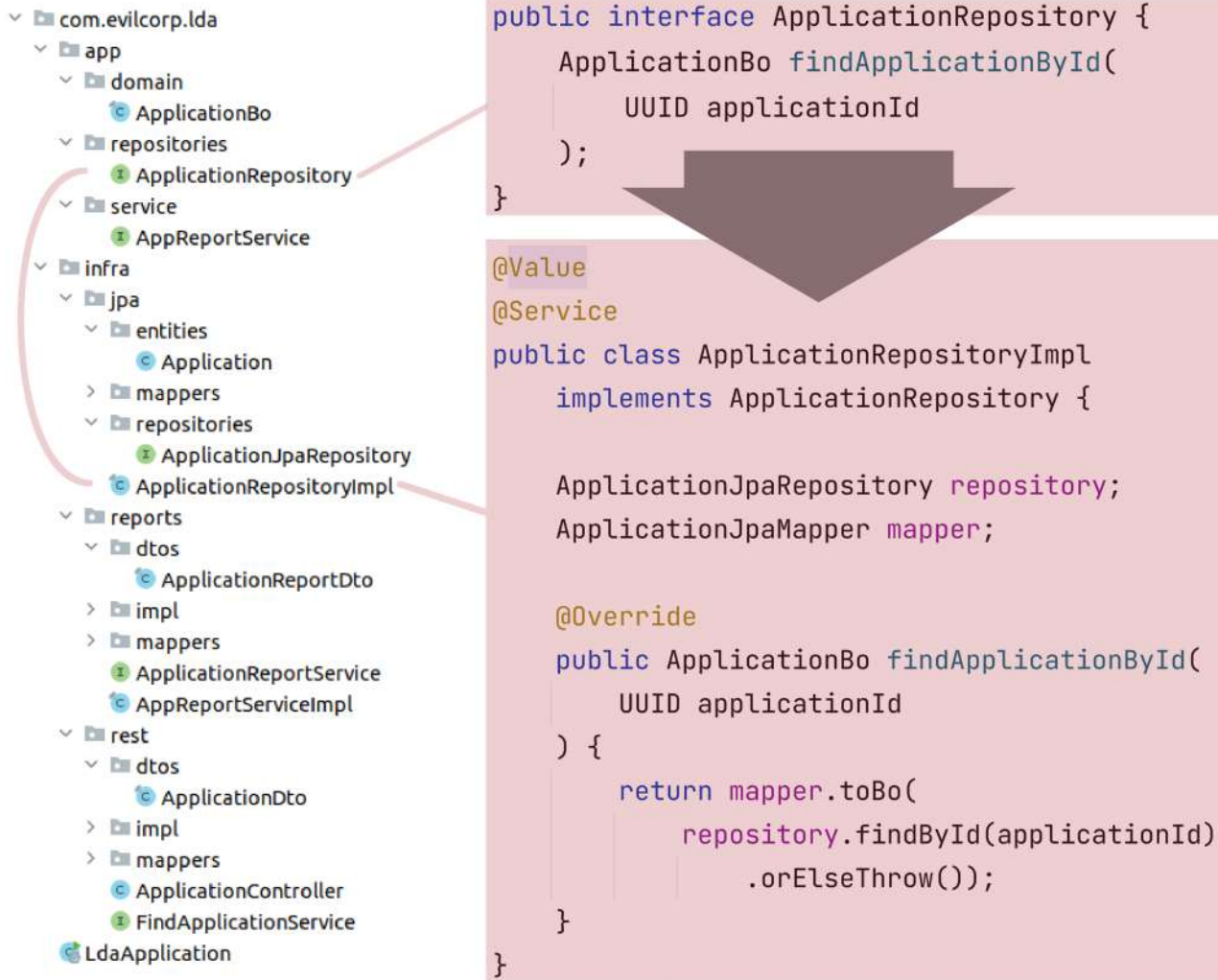
Модульная архитектура



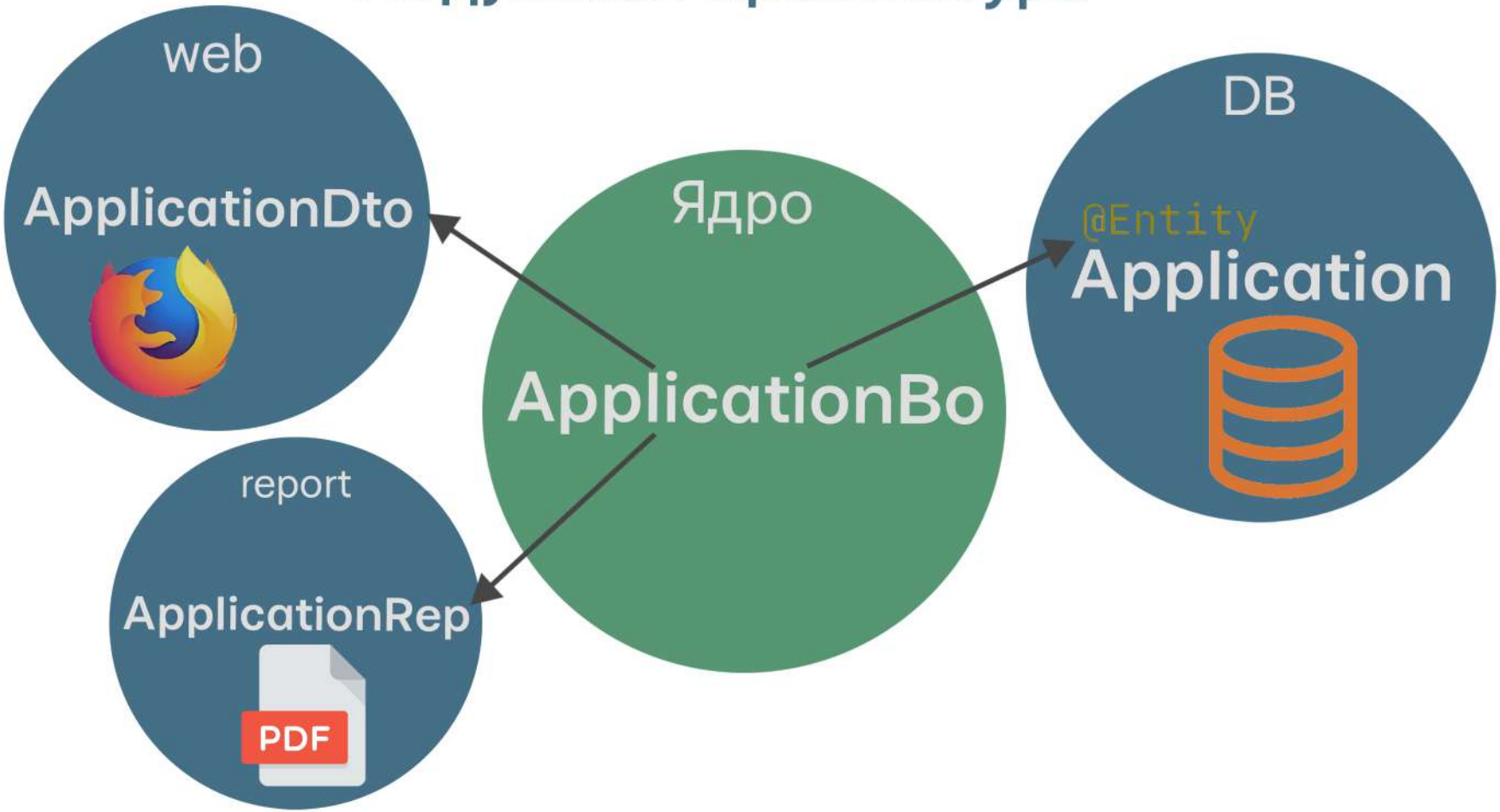
Модульная архитектура



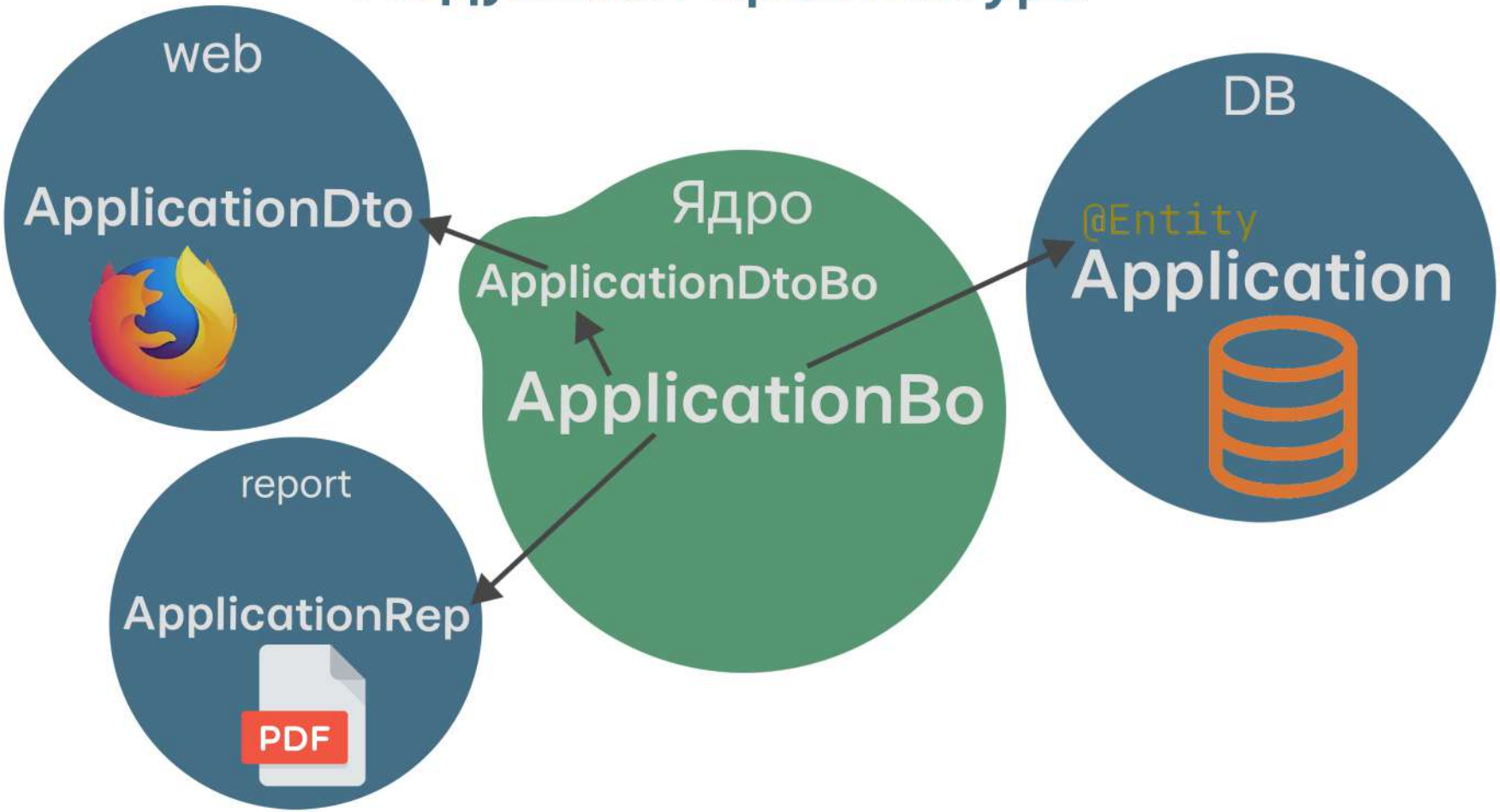

java
com.evilcorp.lda
controllers
ApplicationController
dtos
ApplicationDto
ApplicationReportDto
entities
Application
mappers
ApplicationMapper
repositories
ApplicationRepository
services
impl
ApplicationReportServiceImpl
ApplicationServiceImpl
ApplicationReportService
ApplicationService
LdaApplication
com.evilcorp.lda
app
domain
ApplicationBo
repositories
ApplicationRepository
service
AppReportService
infra
jpa
entities
Application
mappers
repositories
ApplicationJpaRepository
ApplicationRepositoryImpl
reports
dtos
ApplicationReportDto
impl
mappers
ApplicationReportService
AppReportServiceImpl
rest
dtos
ApplicationDto
impl
mappers
ApplicationController
FindApplicationService
LdaApplication


com.evilcorp.lda
app
domain
ApplicationBo
repositories
ApplicationRepository
service
AppReportService
infra
jpa
entities
Application
mappers
repositories
ApplicationJpaRepository
ApplicationRepositoryImpl
reports
dtos
ApplicationReportDto
impl
mappers
ApplicationReportService
AppReportServiceImpl
rest
dtos
ApplicationDto
impl
mappers
ApplicationController
FindApplicationService
LdaApplication
public interface ApplicationRepository {
ApplicationBo findApplicationById(
UUID applicationId
);
}
@Value
@Service
public class ApplicationRepositoryImpl
implements ApplicationRepository {
ApplicationJpaRepository repository;
ApplicationJpaMapper mapper;
@Override
public ApplicationBo findApplicationById(
UUID applicationId
) {
return mapper.toBo(
repository.findById(applicationId)
.orElseThrow());
}
}

Модульная архитектура



Модульная архитектура




У DTO могут быть
разные поля


У DTO могут быть разные поля
Поэтому надо их мапить


У DTO могут быть
разные поля


Поэтому надо их
мапить


Да, сейчас наши DTO
идентичны


У DTO могут быть разные поля
Поэтому надо их мапить
Да, сейчас наши DTO идентичны
Но мы закладываемся на будущее

```java
@Value
@Builder(toBuilder = true)
public class ApplicationReportDto {
    UUID applicationId;
    UUID personId;
    String phone;
    String source;
    LocalDateTime createdAt;
    String type;
}
```

```java
@Value
@Builder(toBuilder = true)
public class ApplicationReportDto {
    UUID applicationId;
    UUID personId;
    String phone;
    String source;
    LocalDateTime createdAt;
    String type;
}
```

```java
public class ApplicationDto {
    UUID applicationId;
    UUID personId;
    String channel;
    LocalDateTime createdAt;
    String type;
}
```

```java
@Value
@Builder(toBuilder = true)
public class ApplicationReportDto {
    UUID applicationId;
    UUID personId;
    String phone;
    String source;
    LocalDateTime createdAt;
    String type;
}
```

```java
public class ApplicationApiBo {
    UUID applicationId;
    UUID personId;
    String channel;
    LocalDateTime createdAt;
    String phone;
    String type;
```

```java
public class ApplicationDto {
    UUID applicationId;
    UUID personId;
    String channel;
    LocalDateTime createdAt;
    String type;
}
```

```java
@Value
@Builder(toBuilder = true)
public class ApplicationReportDto {
    UUID applicationId;
    UUID personId;
    String phone;
    String source;
    LocalDateTime createdAt;
    String type;
}
```

```java
public class ApplicationApiBo {
    UUID applicationId;
    UUID personId;
    String channel;
    LocalDateTime createdAt;
    String phone;
    String type;
```

```java
public class ApplicationDto
    UUID applicationId;
    UUID personId;
    String channel;
    LocalDateTime createdAt
    String type;
}
```

```java
public class Application {
    @Id
    @GeneratedValue(generator = "uuid2")
    @GenericGenerator(name = "uuid2"
        , strategy = "uuid2")
    @Column(name = "application_id")
    UUID applicationId;

    @Column(name = "application_type"
        , insertable = false
        , updatable = false)
    String type;

    @Column(name = "person_id")
    UUID personId;

    @Column(name = "channel")
    String channel;

    @Column(name = "created_at")
    LocalDateTime createdAt;

    @Column(name = "phone")
    String phone;
```

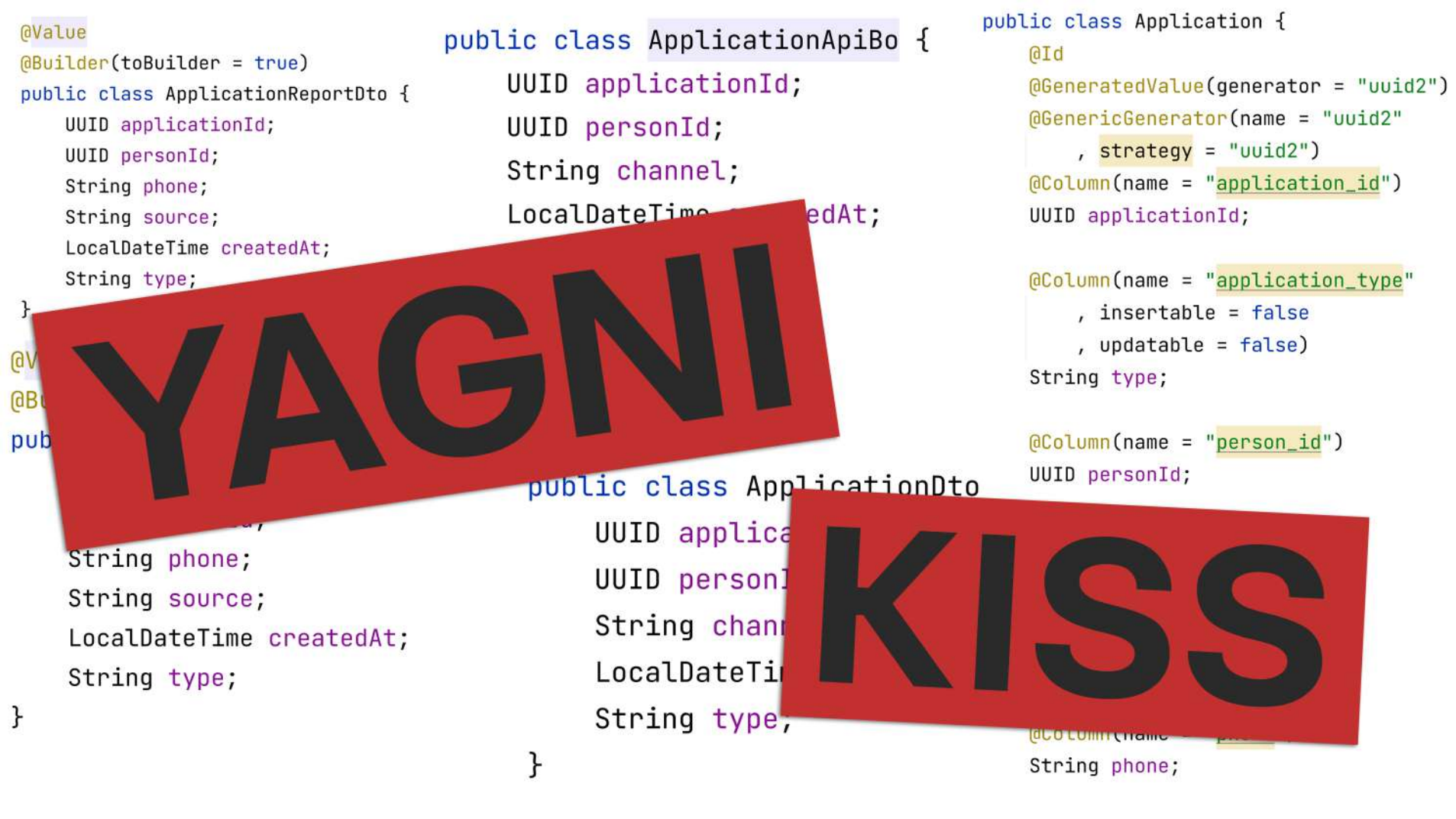
```java
@Value
@Builder(toBuilder = true)
public class ApplicationReportDto {
    UUID applicationId;
    UUID personId;
    String phone;
    String source;
    LocalDateTime createdAt;
    String type;
}
```

```java
public class ApplicationApiBo {
    UUID applicationId;
    UUID personId;
    String channel;
    LocalDateTime createdAt;
    String phone;
    String type;
```

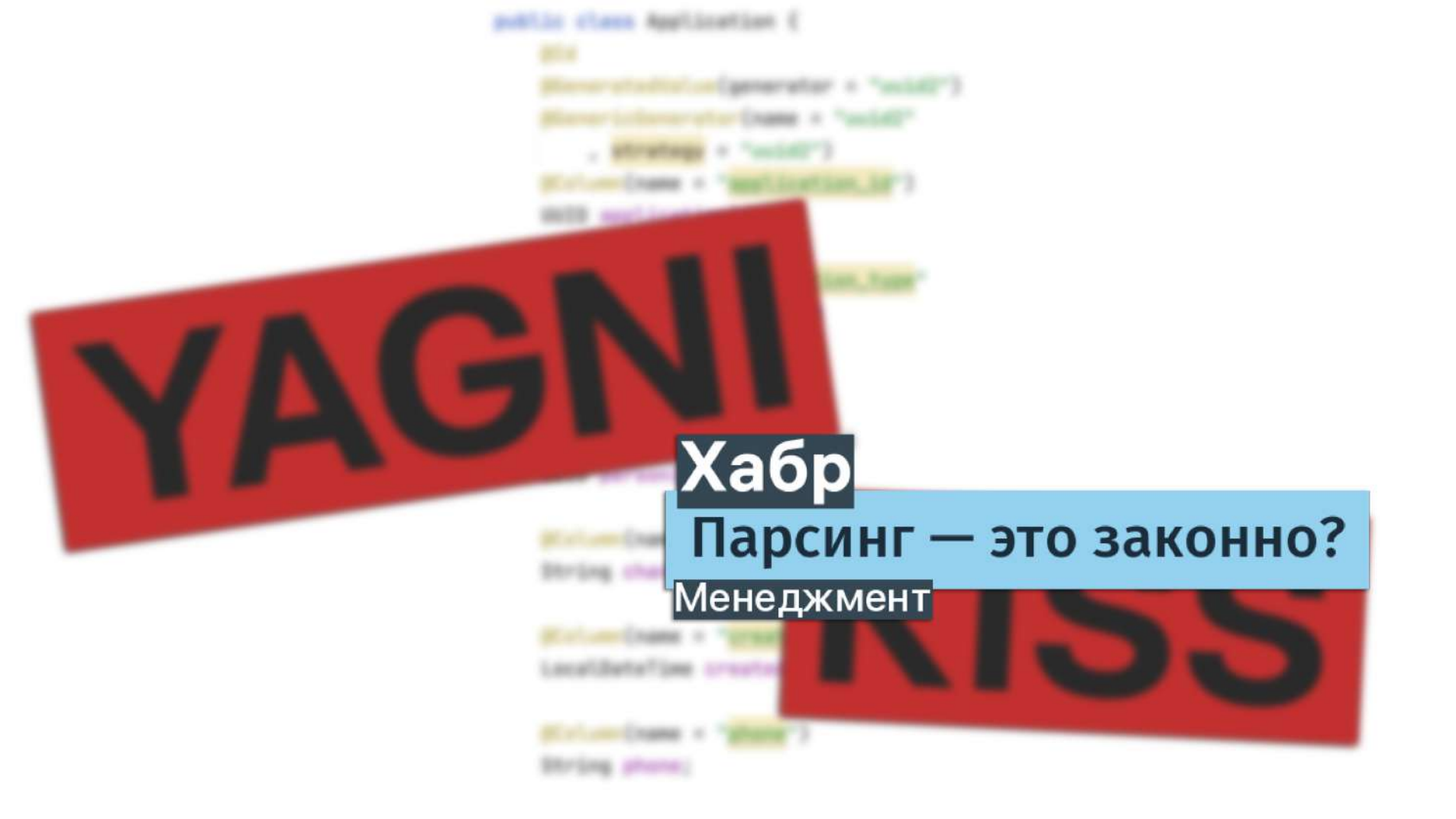
```java
public class Application {
    @Id
    @GeneratedValue(generator = "uuid2")
    @GenericGenerator(name = "uuid2"
        , strategy = "uuid2")
    @Column(name = "application_id")
    UUID applicationId;

    @Column(name = "application_type"
        , insertable = false
        , updatable = false)
    String type;

    @Column(name = "person_id")
    UUID personId;

    @Column(name = "channel")
    String channel;

    @Column(name = "created_at")
    LocalDateTime createdAt;

    @Column(name = "phone")
    String phone;
```

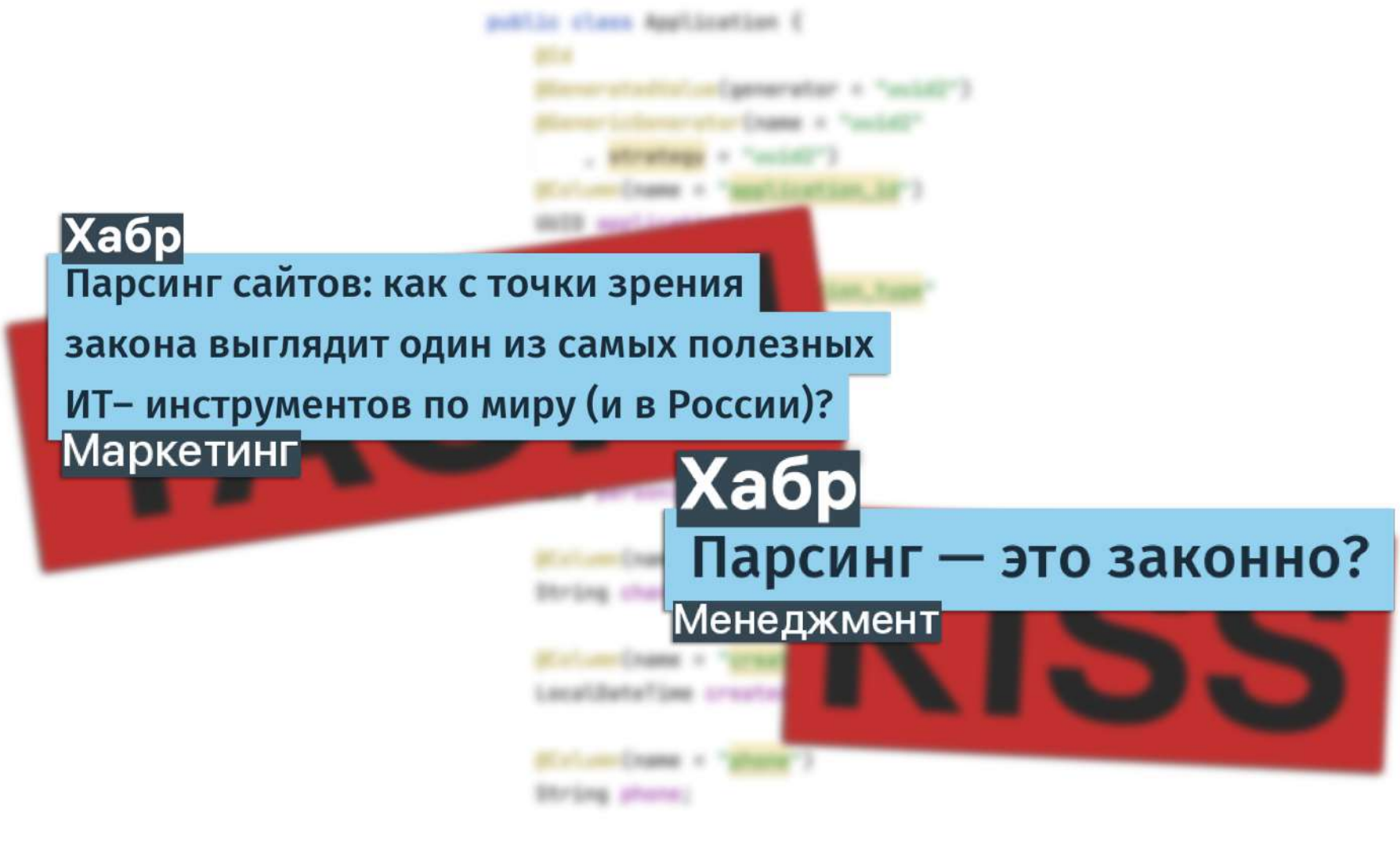

```java
@Value
@Builder(toBuilder = true)
public class ApplicationReportDto {
    UUID applicationId;
    UUID personId;
    String phone;
    String source;
    LocalDateTime createdAt;
    String type;
}
```

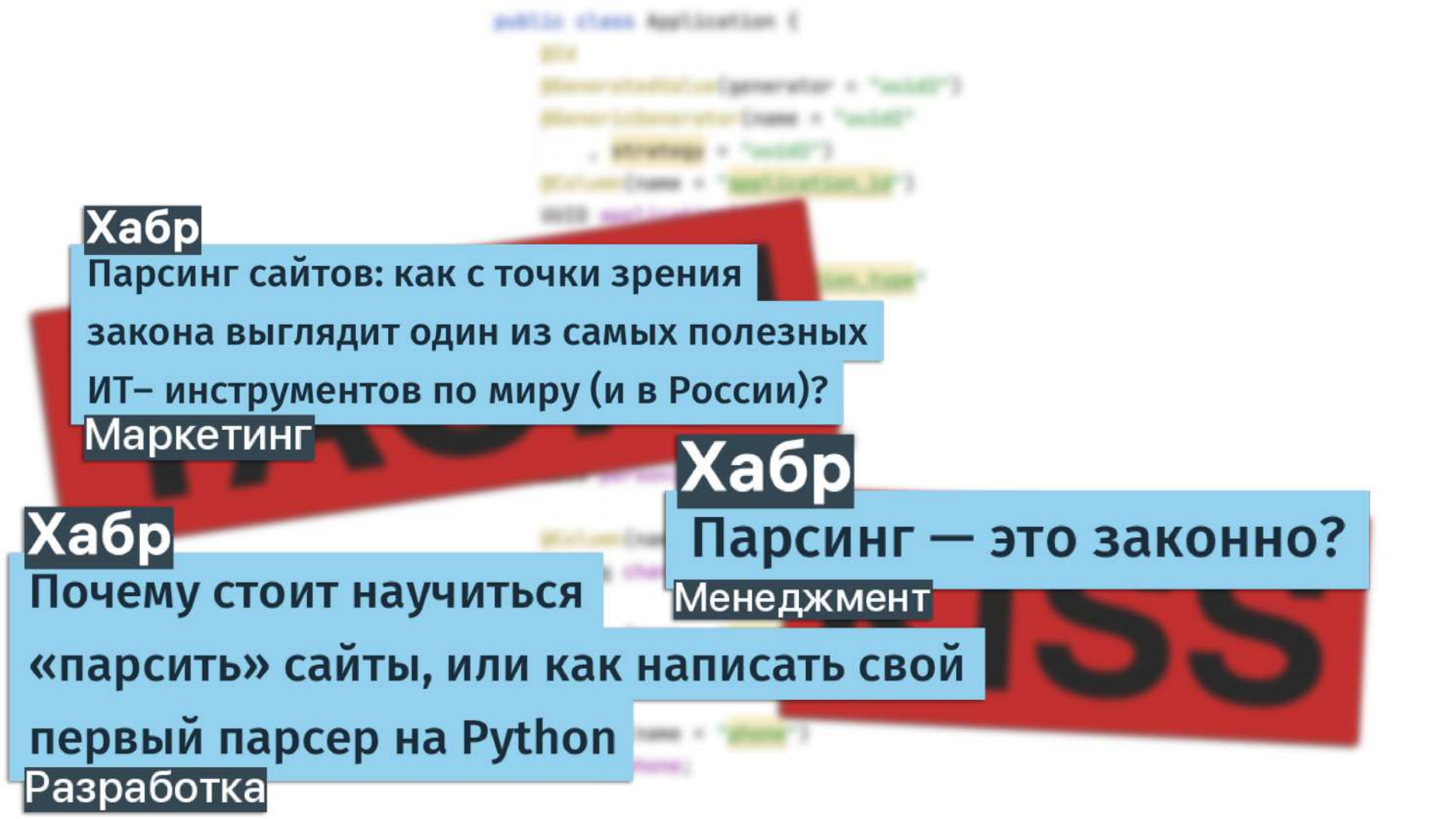

```java
public class ApplicationDto
    UUID applicationId;
    UUID personId;
    String channel;
    LocalDateTime createdAt
    String type;
}
```

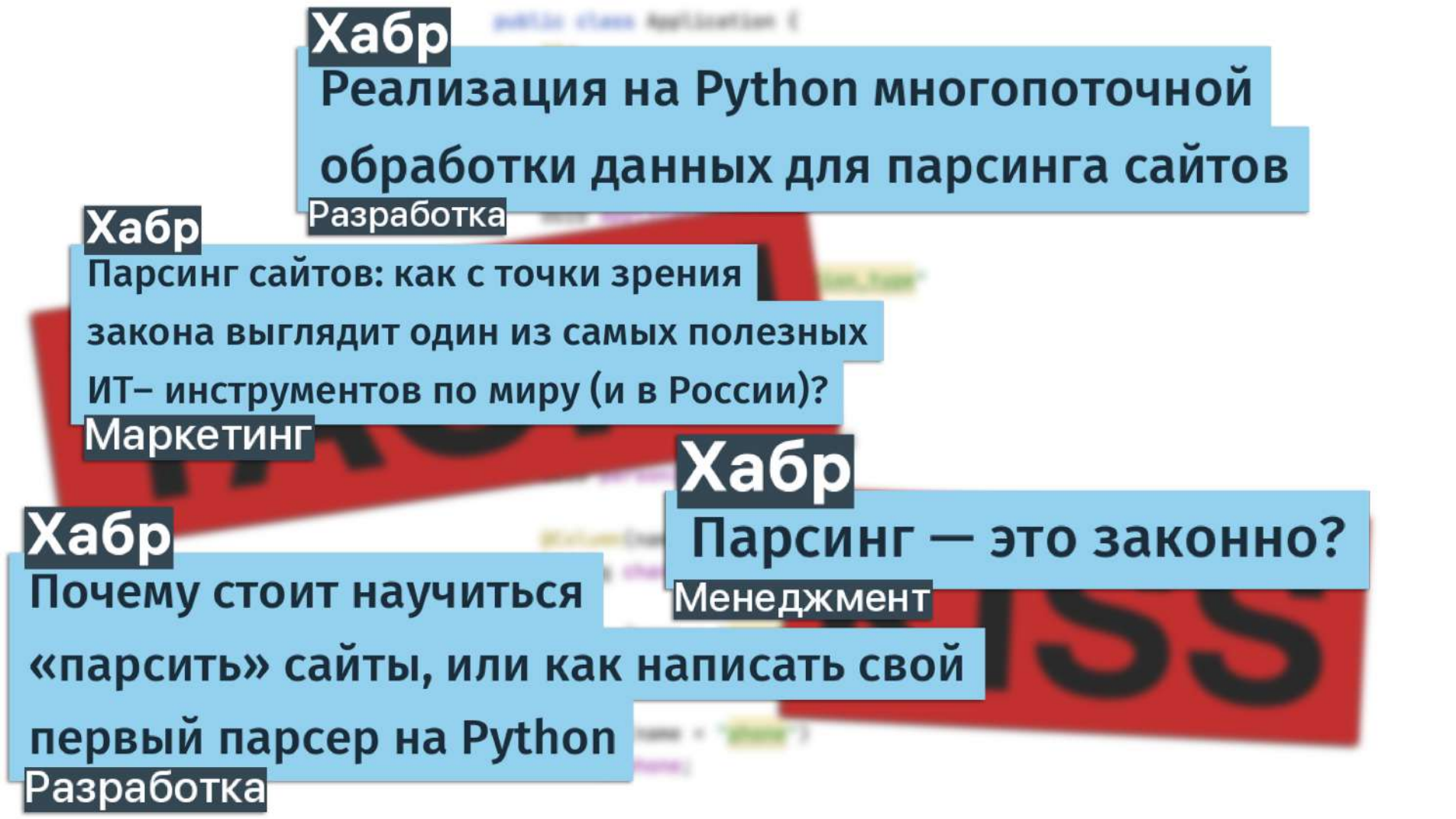

@Value
@Builder(toBuilder = true)
public class ApplicationReportDto {
UUID applicationId;
UUID personId;
String phone;
String source;
LocalDateTime createdAt;
String type;
}
public class ApplicationApiBo {
UUID applicationId;
UUID personId;
String channel;
String type;
}
public class Application {
@Id
@GeneratedValue(generator = "uuid2")
@GenericGenerator(name = "uuid2"
, strategy = "uuid2")
@Column(name = "application_id")
UUID applicationId;
@Column(name = "application_type"
, insertable = false
, updatable = false)
String type;
@Column(name = "person_id")
UUID personId;
String phone;
YAGNI
KISS

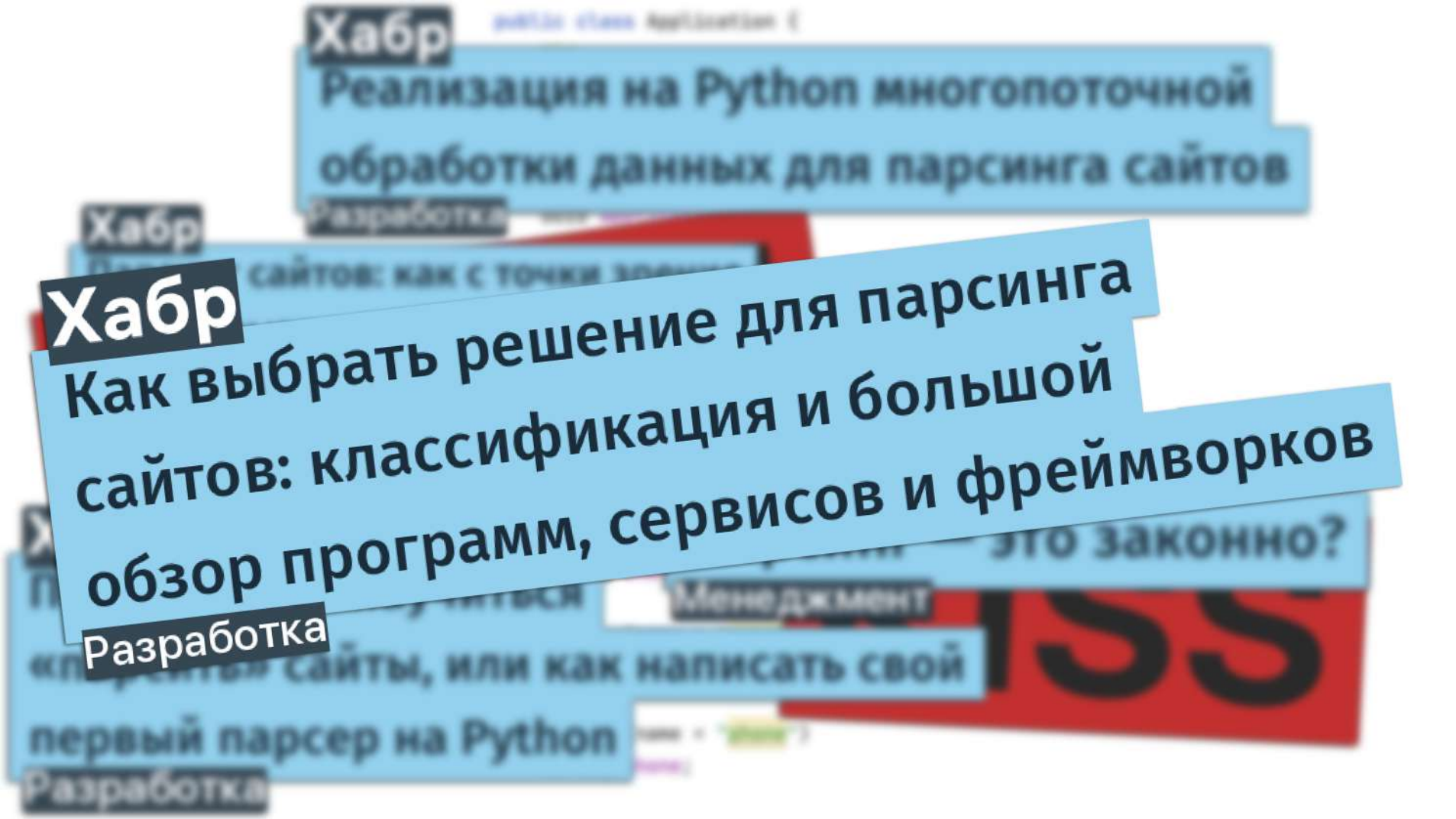

public class Application {
@Id
@GeneratedValue(generator = "uuid2")
@GenericGenerator(name = "uuid2"
, strategy = "uuid2")
@Column(name = "application_id")
UUID applicati
tion_type"
id")
personId;
@Column(name = "chann
String channel;
@Column(name = "creat
LocalDateTime create
@Column(name = "phone")
String phone;
YAGNI
KISS

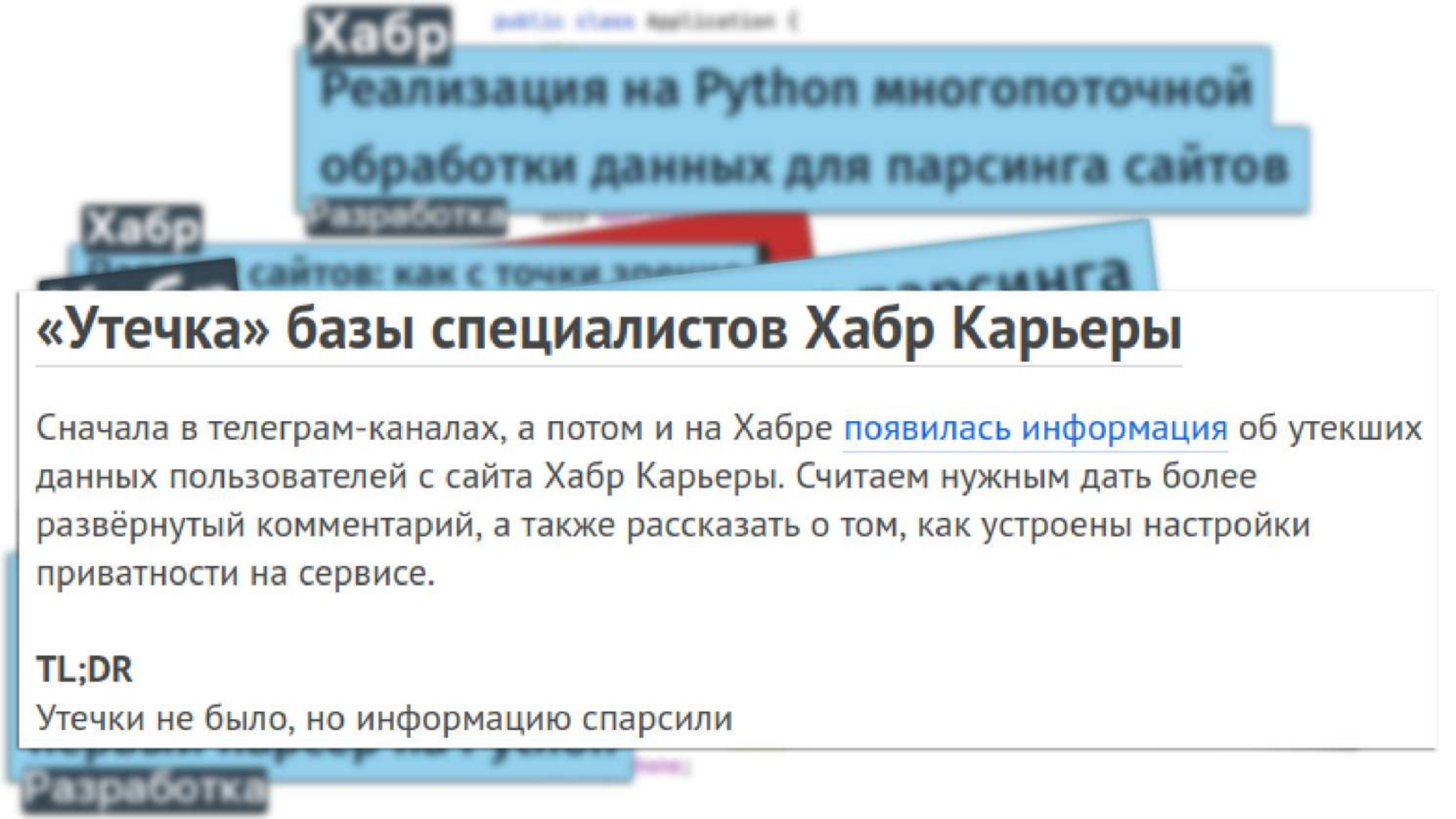

YAGNI
KISS
Хабр
Парсинг — это законно?
Менеджмент


Хабр
Парсинг сайтов: как с точки зрения закона выглядит один из самых полезных ИТ- инструментов по миру (и в России)?
Маркетинг
Хабр
Парсинг — это законно?
Менеджмент


Хабр
Парсинг сайтов: как с точки зрения закона выглядит один из самых полезных ИТ– инструментов по миру (и в России)?
Маркетинг
Хабр
Парсинг — это законно?
Менеджмент
Хабр
Почему стоит научиться «парсить» сайты, или как написать свой первый парсер на Python
Разработка


Хабр
Реализация на Python многопоточной обработки данных для парсинга сайтов
Разработка
Хабр
Парсинг сайтов: как с точки зрения закона выглядит один из самых полезных ИТ- инструментов по миру (и в России)?
Маркетинг
Хабр
Парсинг — это законно?
Менеджмент
Хабр
Почему стоит научиться «парсить» сайты, или как написать свой первый парсер на Python
Разработка


Хабр
Реализация на Python многопоточной обработки данных для парсинга сайтов
Разработка
Хабр
Как выбрать решение для парсинга сайтов: классификация и большой обзор программ, сервисов и фреймворков
Разработка
Менеджмент
Это законно?
первый парсер на Python
Разработка



«Утечка» базы специалистов Хабр Карьеры

Сначала в телеграм-каналах, а потом и на Хабре появилась информация об утекших данных пользователей с сайта Хабр Карьеры. Считаем нужным дать более развёрнутый комментарий, а также рассказать о том, как устроены настройки приватности на сервисе.

TL;DR

Утечки не было, но информацию спарсили

«Лучше ужасный конец,
чем ужас без конца»

Фердинанд фон Шилль

«Лучше, конечно,
помучаться»

Фёдор Иванович Сухов

HAULMONT

DTO

Живи быстро, гори ярко

Андрей Беляев

@belyaev_andrey

Project Lombok


```
@Value
@Builder
public class ApplicationBo {
    UUID applicationId;
    UUID personId;
    String channel;
    LocalDateTime createdAt;
    String phone;
}
```


```java
@Value
@Builder
public class ApplicationBo {
    UUID applicationId;
    UUID personId;
    String channel;
    LocalDateTime createdAt;
    String phone;
}
```

```java
public class Application {
    @Id
    @Column(name = "application_id")
    UUID applicationId;

    @Column(name = "person_id")
    UUID personId;

    @Column(name = "channel")
    String channel;

    @Column(name = "created_at")
    LocalDateTime createdAt;

    @Column(name = "phone")
    String phone;
```

```java
public interface ApplicationJpaMapper {
    ApplicationBo toBo(Application app);
}
```


```java
public interface ApplicationJpaMapper {
    ApplicationBo toBo(Application app);
}
```


```
@Mapper(
    componentModel = "spring",
    unmappedSourcePolicy = ReportingPolicy.ERROR,
    unmappedTargetPolicy = ReportingPolicy.ERROR
)
public interface ApplicationJpaMapper {
    ApplicationBo toBo(Application app);
}
```

```java
@Component
public class ApplicationJpaMapperImpl implements ApplicationJpaMapper {

    @Override
    public ApplicationBo toBo(Application app) {
        if ( app == null ) {
            return null;
        }

        ApplicationBo.ApplicationBoBuilder applicationBo = ApplicationBo.builder();

        applicationBo.applicationId( app.getApplicationId() );
        applicationBo.personId( app.getPersonId() );
        applicationBo.channel( app.getChannel() );
        applicationBo.createdAt( app.getCreatedAt() );
        applicationBo.phone( app.getPhone() );

        return applicationBo.build();
    }
}
```

@SpringBootTest

@Mock

@DataJpaTest

Ничто в современой разработке не имеет смысла кроме как в свете юнит-тестирования

Testcontainers

@JsonTest

JUnit 5

@MockBean

Theodosius Dobzhansky. Nothing in biology makes sense except in the light of evolution // The American Biology Teacher. March 1973. V. 35. P. 125–129

Coverage must flow

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines
com.evilcorp.lda.app.domain.api		19%		0%	68	88	12	20
com.evilcorp.lda.app.domain		47%		10%	53	88	6	20
com.evilcorp.lda.infra.reports.dtos		55%		11%	29	47	0	8
com.evilcorp.lda.infra.jpa.entities		56%		0%	20	55	10	26
com.evilcorp.lda.app.mappers		39%		40%	6	9	22	37
com.evilcorp.lda.infra.reports		49%		10%	16	22	0	6
com.evilcorp.lda.app.service.impl		48%		10%	12	17	0	5
com.evilcorp.lda.infra.jpa		48%		10%	12	17	0	6
com.evilcorp.lda.infra.rest.impl		48%		10%	12	17	0	5
com.evilcorp.lda.infra.reports.impl		74%		0%	4	8	2	15
com.evilcorp.lda.infra.rest.mappers		91%		50%	2	10	2	20
com.evilcorp.lda.infra.jpa.mappers		95%		70%	3	9	3	37
com.evilcorp.lda		37%		n/a	1	2	2	3
com.evilcorp.lda.infra.reports.mappers		95%		50%	1	3	1	11
com.evilcorp.lda.infra.rest		100%		n/a	0	3	0	3
Total	1,433 of 2,805	48%	302 of 338	10%	239	395	60	222

Created with JaCoCo 0.8.8.202204050719

JaCoCo

ApplicationBo

Element	Missed Instructions	Cov.	Missed Branches	Cov.
equals(Object)		0%		0%
ApplicationBo(UUID, UUID, String, LocalDateTime, String, String)		0%		n/a
toString()		0%		n/a
hashCode()		88%		58%
canEqual(Object)		0%		n/a
ApplicationBo(ApplicationBo.ApplicationBoBuilder)		100%		n/a
builder()		100%		n/a
getApplicationId()		100%		n/a
getPersonId()		100%		n/a
getChannel()		100%		n/a
getCreatedAt()		100%		n/a
getPhone()		100%		n/a
getType()		100%		n/a
Total	171 of 300	43%	47 of 54	12%

Project Lombok

```
@Value
@Builder
public class ApplicationBo {
    UUID applicationId;
```

Project Lombok

```
@Value
@Builder
public class ApplicationBo {
    UUID applicationId;
```

```
public UUID getApplicationId() {
    return this.applicationId;
}
```

Project Lombok

```
@Value
@Builder
public class ApplicationBo {
    UUID applicationId;
```

```
@Generated
public UUID getApplicationId() {
    return this.applicationId;
}
```

Project Lombok

```
@Value
@Builder
public class ApplicationBo {
    UUID applicationId;
```

```
@Generated
public UUID getApplicationId() {
    return this.applicationId;
}
```

```
lombok.addLombokGeneratedAnnotation=true
```

Project Lombok + JACOCO

Чтобы игнорировать геттеры и сеттеры

lombok.config

```
lombok.addLombokGeneratedAnnotation=true
```


```java
@Mapper(
    componentModel = "spring",
    unmappedTargetPolicy = ReportingPolicy.ERROR,
    unmappedSourcePolicy = ReportingPolicy.ERROR
)
public interface ApplicationReportMapper {
    @Mapping(target = "source", source = "channel")
    ApplicationReportDto toRepDto(ApplicationBo applicationBo);
}
```

Framework

MapStruct Support

★ ★ ★ ★ ★

MapStruct

Overview Versions Reviews

MapStruct support for IntelliJ IDEA

Website | GitHub | Issue Tracker |

README.md

mapstruct-idea

An IntelliJ IDEA plugin for working with MapStruct

License Apache 2.0

CI passing codecov 80%

- What is MapStruct?
- Features
- Requirements
- Building from Source
- Licensing

What is MapStruct?


```java
@Mapper(
    componentModel = "spring",
    unmappedTargetPolicy = ReportingPolicy.ERROR,
    unmappedSourcePolicy = ReportingPolicy.ERROR
)
public interface ApplicationReportMapper {
    @Mapping(target = "source", source = "channel")
    ApplicationReportDto toRepDto(ApplicationBo applicationBo);
}
```

Может и не надо ничего тестить

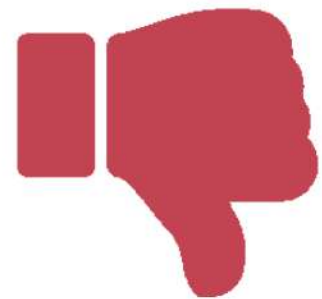
+ JACOCO

```
@Generated(
    value = "org.mapstruct.ap.MappingProcessor",
    date = "2023-06-12T17:49:16+0300",
    comments = "version: 1.5.3.Final, compiler: javac, environment: Java 17.0.1 (GraalVM Community)"
)
@Component
public class ApplicationApiMapperImpl implements ApplicationApiMapper {
```

```java
@Generated(
    value = "org.mapstruct.ap.MappingProcessor",
    date = "2023-07-04T22:47:23+0300",
    comments = "version: 1.5.3.Final, compiler: javac, environment: Java 17.0.1 (GraalVM Community)"
)
@Component
public class ApplicationApiMapperImpl implements ApplicationApiMapper {

    @Override
    public ApplicationApiBo toApp(ApplicationBo app) {
        if ( app == null ) {
            return null;
        }

        if (app instanceof CreditLineApplicationBo) {
            return creditLineApplicationBoToCreditLineApplicationApiBo( (CreditLineApplicationBo) app );
        }
        else if (app instanceof TrancheApplicationBo) {
            return trancheApplicationBoToTrancheApplicationApiBo( (TrancheApplicationBo) app );
        }
        else {
            ApplicationApiBo.ApplicationApiBoBuilder<?, ?> applicationApiBo = ApplicationApiBo.builder();

            applicationApiBo.applicationId( app.getApplicationId() );
            applicationApiBo.personId( app.getPersonId() );
            applicationApiBo.channel( app.getChannel() );
            applicationApiBo.createdAt( app.getCreatedAt() );
            applicationApiBo.phone( app.getPhone() );
            applicationApiBo.type( app.getType() );

            return applicationApiBo.build();
        }
    }
```

+ JACOCO

```
@Generated(
    value = "org.mapstruct.ap.MappingProcessor",
    date = "2023-06-12T17:49:16+0300",
    comments = "version: 1.5.3.Final, compiler: javac, environment: Java 17.0.1 (GraalVM Community)"
)
@Component
public class ApplicationApiMapperImpl implements ApplicationApiMapper {
```



```
@Documented
@Retention(SOURCE)
@Target({PACKAGE, TYPE, METHOD, CONSTRUCTOR, FIELD,
    LOCAL_VARIABLE, PARAMETER})
public @interface Generated {
```

Improvement: @Generated annotation to mark code that should be ignored by JaCoCo #1528

New issue

Closed **empovit** opened this issue on Jun 18, 2018 · 16 comments

empovit commented on Jun 18, 2018

It would be nice to have an annotation with CLASS retention policy that can be ignored by code coverage tools like JaCoCo. This should work similar to how it's done for Immutables and Lombok.

12

Assignees

No one assigned

Labels

feature for:team-discussion

Add annotations to Generated code #1574

New issue

Closed **vadim-shb** opened this issue on Aug 5, 2018 · 20 comments · Fixed by #2792

Zegveld commented on Nov 17, 2022 Contributor ...

It does not allow you to replace the default `@Generated` but it will allow you to add a secondary `@Generated` annotation.

For example:

```
@Mapper
@AnnotateWith( com.example.Generated.class )
interface MyMapper {
...
```

could turn into:

```
import javax.annotation.processing.Generated;

@Generated(
    value = "org.mapstruct.ap.MappingProcessor",
    date = "2022-11-17T15:57:41+0200",
    comments = "version: 1.6.0, compiler: javac, environment: Java 17.0.4 (Azul Systems, Inc.)"
)
@com.example.Generated
public class MyMapperImpl implements MyMapper {
...
```

+ JACOCO

```
<plugin>
    <groupId>org.jacoco</groupId>
    <artifactId>jacoco-maven-plugin</artifactId>
    <version>0.8.8</version>
    <configuration>
        <excludes>
            <exclude>**/**IgnoreCoverage.class</exclude>
        </excludes>
    </configuration>
    <executions>
```

```
@Mapper(
    componentModel = "spring",
    unmappedTargetPolicy = ReportingPolicy.ERROR,
    unmappedSourcePolicy = ReportingPolicy.ERROR,
    implementationName = "CreditLineApplicationMapperIgnoreCoverage"
)
public interface CreditLineApplicationMapper extends ApplicationConc
```

```java
ApplicationJpaMapper mapper = Mappers.getMapper(ApplicationJpaMapper.class);

@Test
void toBo() {
    final Application in = Application.builder()
        .applicationId(UUID.randomUUID())
        .personId(UUID.randomUUID())
        .channel("TEST_CHANNEL")
        .phone("123")
        .createdAt(LocalDateTime.now().truncatedTo(ChronoUnit.SECONDS))
        .build();

    final ApplicationBo out = mapper.toBo(in);

    assertThat(in.getApplicationId()).isEqualTo(out.getApplicationId());
    assertThat(in.getPersonId()).isEqualTo(out.getPersonId());
    assertThat(in.getChannel()).isEqualTo(out.getChannel());
    assertThat(in.getPhone()).isEqualTo(out.getPhone());
    assertThat(in.getCreatedAt()).isEqualTo(out.getCreatedAt());
```

```
ApplicationJpaMapper mapper = Mappers.getMapper(ApplicationJpaMapper.class);

@Test
void toBo() {
    final Application in = Application.builder()
        .applicationId(UUID.randomUUID())
        .personId(UUID.randomUUID())
        .channel("TEST_CHANNEL")
        .phone("123")
        .createdAt(LocalDateTime.now().truncatedTo(ChronoUnit.SECONDS))
        .build();

    final ApplicationBo out = mapper.toBo(in);

    assertThat(in.getApplicationId()).isEqualTo(out.getApplicationId());
    assertThat(in.getPersonId()).isEqualTo(out.getPersonId());
    assertThat(in.getChannel()).isEqualTo(out.getChannel());
    assertThat(in.getPhone()).isEqualTo(out.getPhone());
    assertThat(in.getCreatedAt()).isEqualTo(out.getCreatedAt());
```

Бессмысленный тест?

```java
ApplicationJpaMapper mapper = Mappers.getMapper(ApplicationJpaMapper.class);

@Test
void toBo() {
    final Application in = Application.builder()
        .applicationId(UUID.randomUUID())
        .personId(UUID.randomUUID())
        .channel("TEST_CHANNEL")
        .phone("123")
        .createdAt(LocalDateTime.now().truncatedTo(ChronoUnit.SECONDS))
        .build();
    final Application in = factory.build(Application.class);
    final ApplicationBo out = mapper.toBo(in);

    assertThat(in.getApplicationId()).isEqualTo(out.getApplicationId());
    assertThat(in.getPersonId()).isEqualTo(out.getPersonId());
    assertThat(in.getChannel()).isEqualTo(out.getChannel());
    assertThat(in.getPhone()).isEqualTo(out.getPhone());
    assertThat(in.getCreatedAt()).isEqualTo(out.getCreatedAt());
```

Бессмысленный тест?

```java
ApplicationJpaMapper mapper = Mappers.getMapper(ApplicationJpaMapper.class);

@Test
void toBo() {
    final Application in = Application.builder()
        .applicationId(UUID.randomUUID())
        .personId(UUID.randomUUID())
        .channel("TEST_CHANNEL")
        .phone("123")
        .createdAt(LocalDateTime.now().truncatedTo(ChronoUnit.SECONDS))
        .build();
    final Application in = factory.build(Application.class);
    final ApplicationBo out = mapper.toBo(in);

    assertThat(in.getApplicationId()).isEqualTo(out.getApplicationId());
    assertThat(in.getPersonId()).isEqualTo(out.getPersonId());
    assertThat(in.getChannel()).isEqualTo(out.getChannel());
    assertThat(in.getPhone()).isEqualTo(out.getPhone());
    assertThat(in.getCreatedAt()).isEqualTo(out.getCreatedAt());

    assertThat(in).usingRecursiveComparison()
        .isEqualTo(out);
```

Бессмысленный тест?

```java
ApplicationJpaMapper mapper = Mappers.getMapper(ApplicationJpaMapper.class);

@Test
void toBo() {
    final Application in = Application.builder()
        .applicationId(UUID.randomUUID())
        .personId(UUID.randomUUID())
        .channel("TEST_CHANNEL")
        .phone("123")
        .createdAt(LocalDateTime.now().truncatedTo(ChronoUnit.SECONDS))
        .build();
    final Application in = factory.build(Application.class);
    final ApplicationBo out = mapper.toBo(in);

    assertThat(in.getApplicationId()).isEqualTo(out.getApplicationId());
    assertThat(in.getPersonId()).isEqualTo(out.getPersonId());
    assertThat(in.getChannel()).isEqualTo(out.getChannel());
    assertThat(in.getPhone()).isEqualTo(out.getPhone());
    assertThat(in.getCreatedAt()).isEqualTo(out.getCreatedAt());

    assertThat(in).usingRecursiveComparison()
        .isEqualTo(out);
```

Бессмысленный тест?

JPoint 2020

assertTrue, как много в этом слове

Владимир Ситников
Netcracker

```java
    @Test
    void toRepDto() {
        final ApplicationBo in = easyRandom.nextObject(TrancheApplicationBo.class);

        final ApplicationReportDto out = mapper.toRepDto(in);

        assertThat(out).usingRecursiveComparison()
            .ignoringFields("source")
            .isEqualTo(in);

        assertThat(out.getSource()).isEqualTo(in.getChannel());
    }
}
```

```java
class ApplicationReportMapperAgainstMapperTest {
    private final ApplicationReportMapper mapper = Mappers.getMapper(ApplicationReportMapper.class);
    private final TestAppMapper testAppMapper = Mappers.getMapper(TestAppMapper.class);
    private EasyRandom easyRandom = new EasyRandom();

    @Test
    void toRepDto() {
        final ApplicationBo in = easyRandom.nextObject(TrancheApplicationBo.class);

        final ApplicationReportDto expected = testAppMapper.toApp(in)
            .toBuilder()
            .source(in.getChannel())
            .build()
        ;

        ApplicationReportDto actual = mapper.toRepDto(in);

        assertThat(actual).usingRecursiveComparison()
            .isEqualTo(expected);
    }
}
```

```java
@Mapper(
    unmappedSourcePolicy = ReportingPolicy.ERROR,
    unmappedTargetPolicy = ReportingPolicy.ERROR
)
interface TestAppMapper {
    @Mapping(target = "source", ignore = true)
    ApplicationReportDto toApp(ApplicationBo in);
}
```

1. Есть причины чтобы делать много ДТО

2. Сейчас для этого есть удобные инструменты

3. Тестируйте маппинг!


OCEANO OCCIDENTALE
TROPICO DE CANCRO
MEDITERANEO

Спасибо!


ATACAMA
OCEANO MERIDIONALE

Илья Сазонов

@imsazonov

poxvuibr@gmail

Фёдор Сазонов

@sazonovfm

sazonovfm@gmail