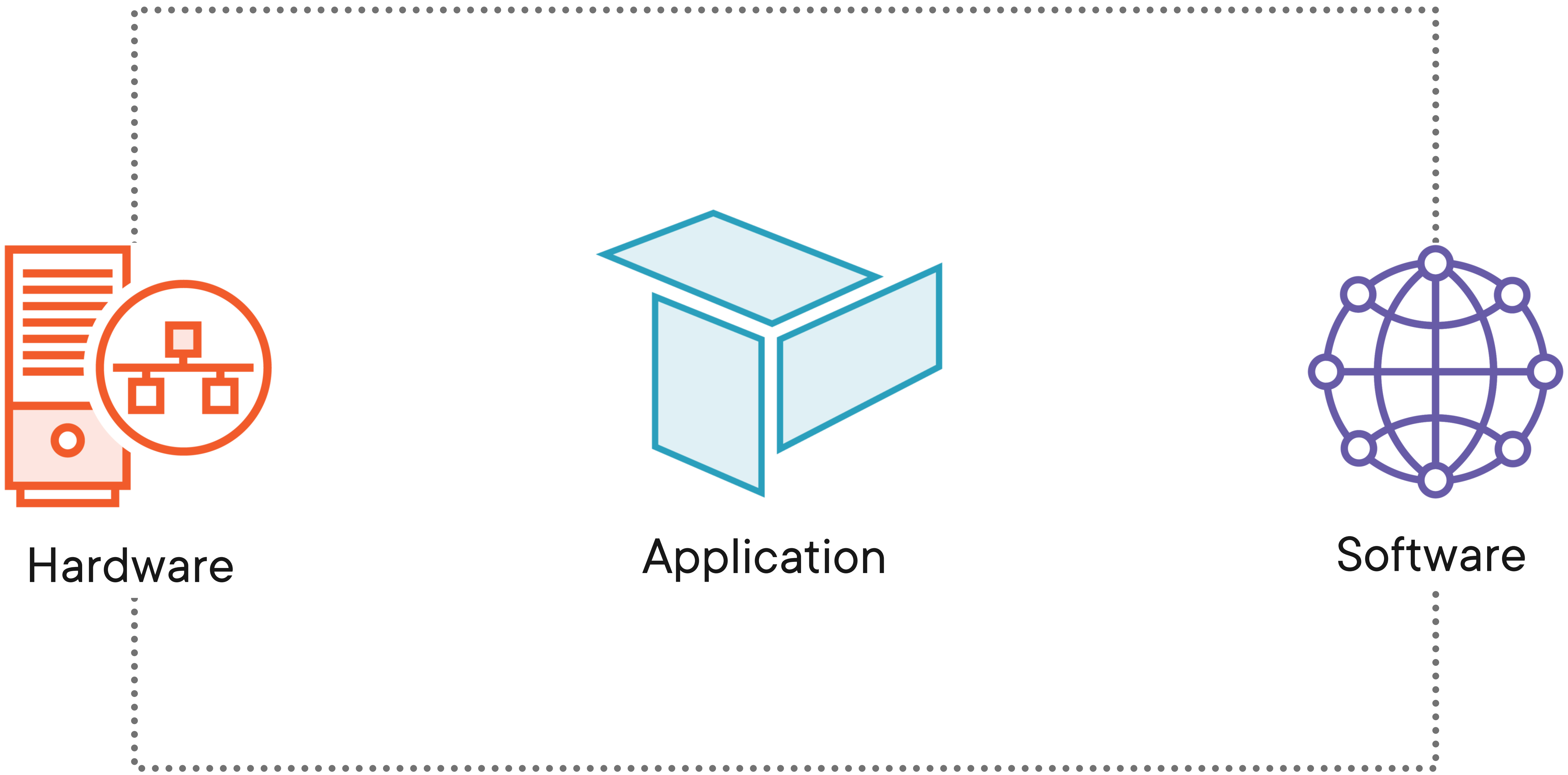
Configuring Micronaut Applications

Bogdan Sucaciu

Tech Lead

@bsucaciu bsucaciu.com

Environment



Hardware

Application

Software

Development Lifecycle

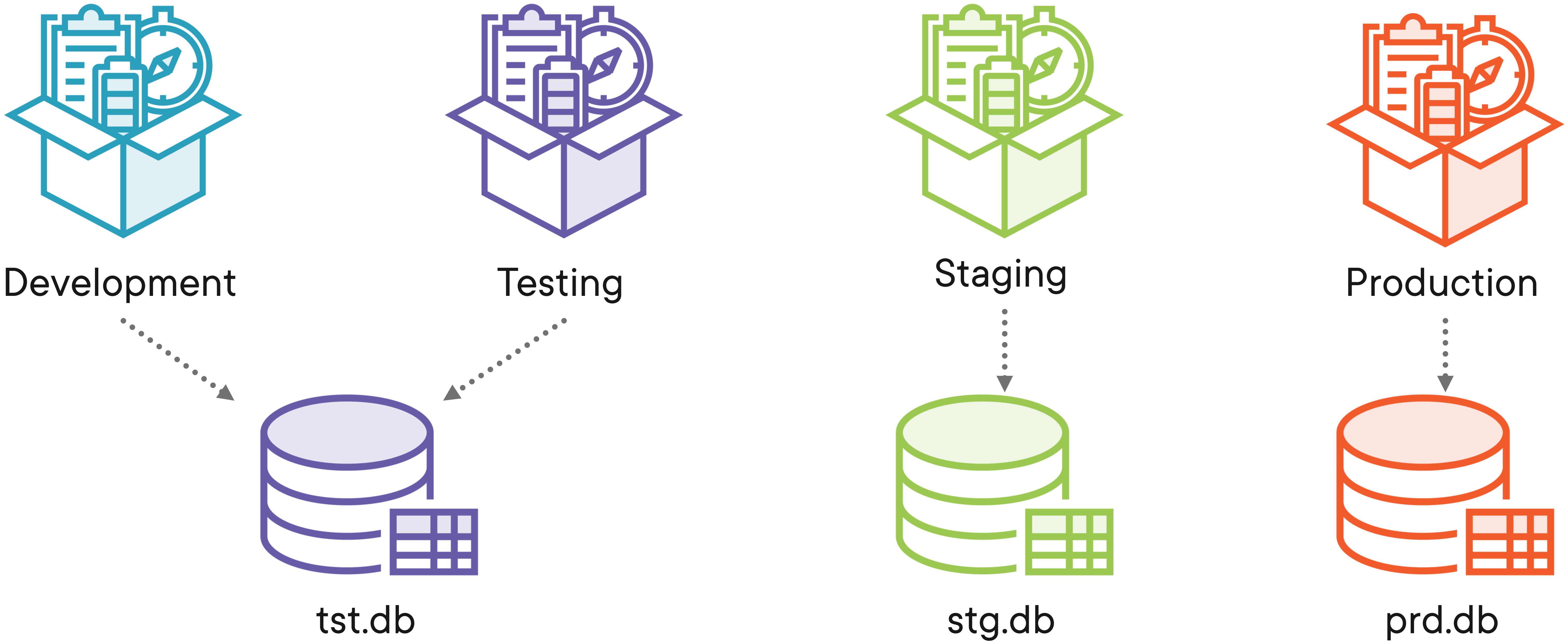
Development

Testing

Staging

Production

Environments



Development

Testing

Staging

Production

tst.db

stg.db

prd.db

Properties

Internal configuration

External resources

Credentials

Feature flags

Environment Priority

Application

dev	test	prod
a	b	c
db.url	db.url	db.url

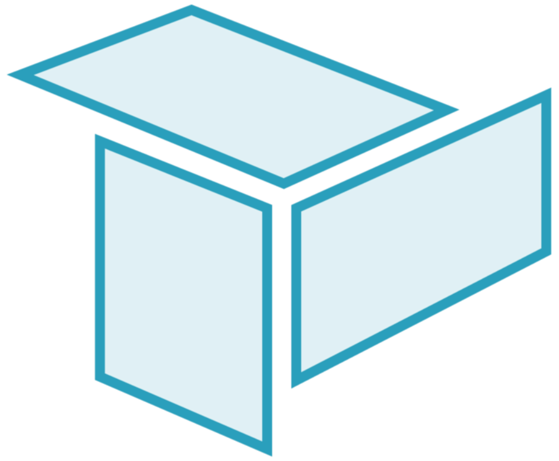
Application Context Builder

MICRONAUT_ENVIRONMENTS

micronaut.environments

Default environment

Property Sources



Application

- Command line arguments
- SPRING_APPLICATION_JSON
- MICRONAUT_APPLICATION_JSON
- Java System Properties
- OS environment variables
- micronaut.config.files
- application-{environment}.{extension}
- application.{extension}

.yaml

.properties

.json

Demo

Configuration injection

Environments

Injecting Maps and Lists

Injecting Maps

application.yaml

```yaml
sensor:
 thresholds:
   temp: 550
   humidity: 80
   fan-speed: 1500
```

LightProperties.java

```java
@ConfigurationProperties(”sensor")
class SensorsProperties {

@MapFormat(transformation =
    MapFormat.MapTransformation.FLAT)
Map<String, String> thresholds;

}
```

@EachProperty

application.yaml

```yaml
updates:
  one:
    url: https://updates.one
  two:
    url: https://updates.two
  three:
    url: https://updates.three
```

ReportsProperties.java

```java
@EachProperty(value = ”updates")
class UpdatesSourceConfiguration {

  String name;
  String url;

}
```

Injecting Lists

application.yaml

```yaml
updates:
 urls:
 - https://updates.one
 - https://updates.two
 - https://updates.three
```

ReportsProperties.java

```java
@Property(name= “updates.urls”)
List<String> urls;
```

@EachProperty

application.yaml

```yaml
updates:
 urls:
 - https://updates.one
 - https://updates.two
 - https://updates.three
```

ReportsProperties.java

```java
@EachProperty(value = ”updates.urls”,
              list = true)
class UpdatesSourceConfiguration {

String url;

}
```

@EachBean

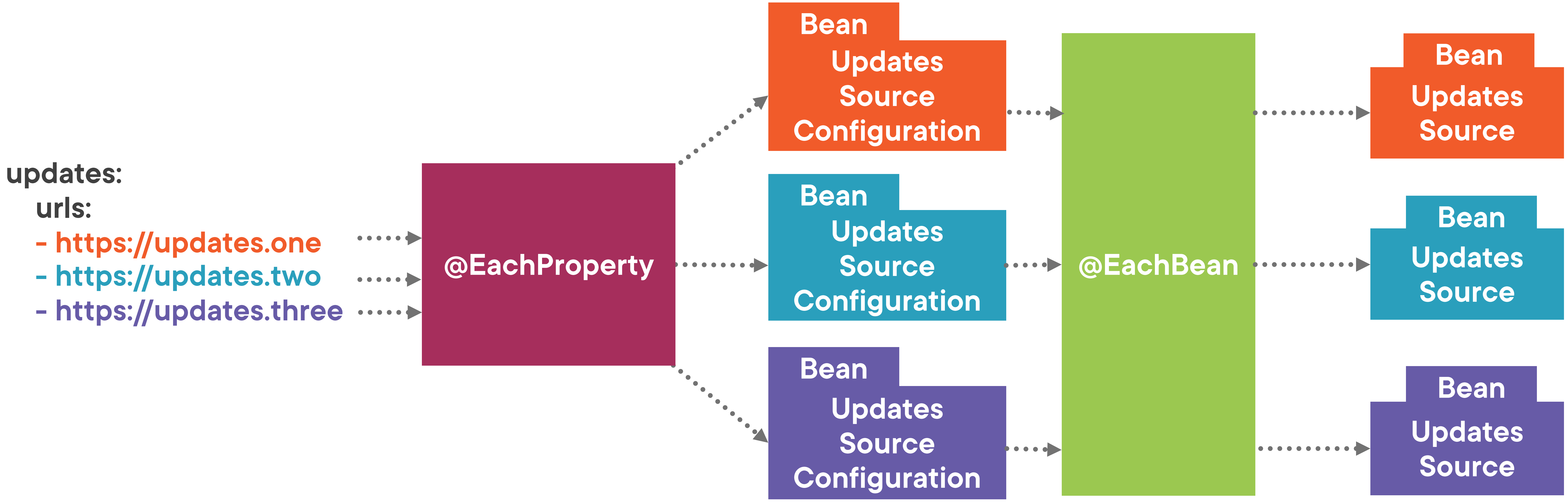
Filename.code

```
@Factory
class ReportService {

  @EachBean(UpdatesSourceConfiguration.class)
  UpdatesSource updatesSource(UpdatesSourceConfiguration configuration) {
    return new UpdatesSource(configuration);
  }

}
```

@EachProperty and @EachBean



```yaml
updates:
  urls:
  - https://updates.one
  - https://updates.two
  - https://updates.three
```

@EachProperty

Bean Updates Source Configuration

Bean Updates Source Configuration

Bean Updates Source Configuration

@EachBean

Bean Updates Source

Bean Updates Source

Bean Updates Source

Demo

Dynamic bean creation

Easter egg

Up Next:
Creating Aspects