PRESENTED BY RED HAT

LEARN. NETWORK.
EXPERIENCE OPEN SOURCE.

www.theredhatsummit.com

Copyright 2011 Emmanuel Bernard and Red Hat Inc.

Bean Validation

Best practices for real life

Emmanuel Bernard
Platform Architect (but still does stuff)
JBoss by Red Hat

SUMMIT JBoss WORLD
PRESENTED BY RED HAT

Copyright 2011 Emmanuel Bernard and Red Hat Inc.

WILIWIG

- Understand Bean Validation
- Improve your apps
- Learn advanced usage and best practices

Copyright 2011 Emmanuel Bernard and Red Hat Inc.

Emmanuel Bernard

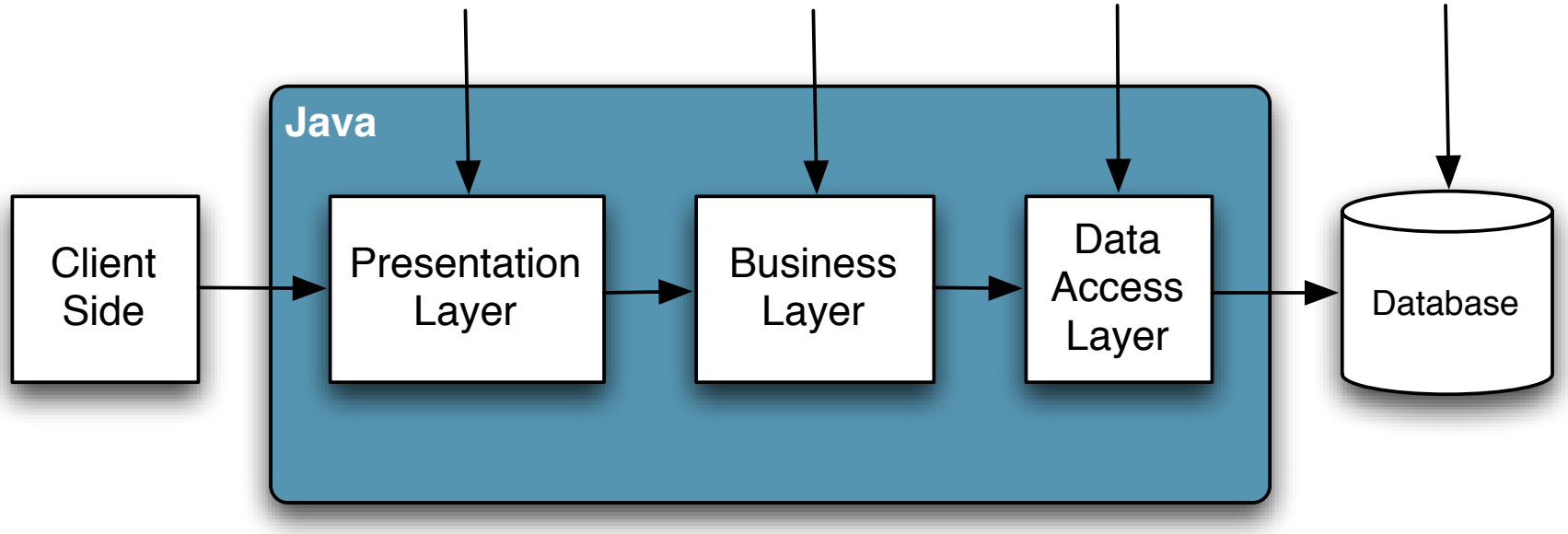
- Work for JBoss by Red Hat
- Hibernate * (founder, lead or contributor)
- JCP (Bean Validation, Java Persistence)
- Founder of Les Cast Codeurs Podcast
- Author of Hibernate Search in Action

- @emmanuelbernard

Copyright 2011 Emmanuel Bernard and Red Hat Inc.

Why Bean Validation

- Before
  - duplication of constraints
  - multiple implementations / inconsistencies



Copyright 2011 Emmanuel Bernard and Red Hat Inc.

Why Bean Validation

- After
  - constrain once, run anywhere
  - standardized
  - overall integration
  - a subtle feeling of happiness

Copyright 2011 Emmanuel Bernard and Red Hat Inc.

Constraint declaration

```
class Address {
  @NotNull @Size(max=50)
  String getStreet1() { return street1 };

  @ZipCode(message="Postal code does not look right")
  String getZipCode() { return zipCode; }

  @NotNull @Valid
  public Country getCountry() { return country; }
}

class Country {
  @NotEmpty String getISO2() { return iso2; }

}
```


Copyright 2011 Emmanuel Bernard and Red Hat Inc.

Class level constraint

```
@ZipCode(message="Zipcode not valid for country")
class Address {
  @NotNull @Size(max=50)
  String getStreet1() { return street1 };

  @NotNull @Size(max=5)
  String getZipCode() { return zipCode; }

  @NotNull
  public Country getCountry() { return country; }
}
```


Copyright 2011 Emmanuel Bernard and Red Hat Inc.

Groups

- Subset of constraints
- Use case
  - partial validation
  - use case validation (state driven)
  - order constraints

Copyright 2011 Emmanuel Bernard and Red Hat Inc.

Demo

Copyright 2011 Emmanuel Bernard and Red Hat Inc.

Writing constraints

- Create an annotation
- Create a validation class
  - implement isValid
- Associate annotation and validation

Copyright 2011 Emmanuel Bernard and Red Hat Inc.

Show me the code

```
@Target({ METHOD, FIELD, ANNOTATION_TYPE, PARAMETER })
@Retention(RUNTIME)
@Constraint(validatedBy=StringZipCodeValidator.class)
@interface ZipCode {
  String message() default
    “{com.jboss.sample.ZipCode.message}”;
  Class<?>[] groups() default {};
  Class<? extends Payload> payload() default {};

  String country() default “fr”;
}
```


Copyright 2011 Emmanuel Bernard and Red Hat Inc.

```
class StringZipCodeValidator
    implements ConstraintValidator<ZipCode,String> {
  private String country;
  private Set<String> validDept = new HashSet<>();

  public void initialize(ZipCode zipCode) {
    this.country = zipCode.country();
    if ( !“fr”.equalsIgnoreCase( this.country ) ) {
      throw new IllegalStateException(UNKNOWN_COUNTRY);
    }
  }

  public boolean isValid(String value,
      ConstraintValidationContext context) {
    if ( value == null ) return true;
    return validDept.contains( value.substring(0,2) );
  }
}
```


Copyright 2011 Emmanuel Bernard and Red Hat Inc.

Compile time checking

- Annotation processor
  - extends the compiler
  - compile time, not build time
  - standard in Java 6
- Make use of Bean Validation's type-safety

Copyright 2011 Emmanuel Bernard and Red Hat Inc.

Composition & Context

- Composition
  - Reuse existing constraints
  - Reusability mechanism
- Context: altering error reports
  - use case #1: multi-property validation

Copyright 2011 Emmanuel Bernard and Red Hat Inc.

Demo

Copyright 2011 Emmanuel Bernard and Red Hat Inc.

Customizing the runtime

- MessageInterpolator
  - influence message generation
- TraversableResolver
  - is an association reachable / lazy
- ConstraintValidatorFactory

Copyright 2011 Emmanuel Bernard and Red Hat Inc.

Using the metadata

- Beyond Java or for metaprogramming

```
Set<ConstraintDescriptor<?>> constraints =
v.getConstraintsForClass(Address.class)
    .getConstraintsForProperty("zipCode")
    .findConstraints()
      .unorderedAndMatchingGroup(Billable.class)
      .getConstraintDescriptors();

for (ConstraintDescriptor<?> cd : constraints) {
  processIt( cd );
  for (ConstrDescr<?> scd : cd.getComposingConstraints()) {
    processIt(scd);
  }
}
```


Copyright 2011 Emmanuel Bernard and Red Hat Inc.

Sustainable development

- Frameworks do the integration for you
- Integrates into your environments
  - Only requires SE 5
  - EE 6 ecosystem (JPA 2, JSF 2)
  - Wicket
  - Tapestry
  - Spring Framework
  - Flex via GraniteDS
  - MetaWidget

Copyright 2011 Emmanuel Bernard and Red Hat Inc.

Conclusion

- Just Use It(tm)

- Hibernate Validator 4.2
  - programmatic mapping API
  - method level validation
  - more complete annotation processor
  - boolean composition
  - joda types support

Copyright 2011 Emmanuel Bernard and Red Hat Inc.

Bean Validation 1.1

- Integration with other specs
- Method level validation
- Group propagation
- Boolean composition
- And many more

- http://opensource.atlassian.com/projects/hibernate/browse/BVAL

Copyright 2011 Emmanuel Bernard and Red Hat Inc.

Q&A

- http://validator.hibernate.org
- http://in.relation.to/tag/Bean+Validation
- http://opensource.atlassian.com/projects/hibernate/browse/BVAL

Copyright 2011 Emmanuel Bernard and Red Hat Inc.

LIKE US ON FACEBOOK

www.facebook.com/redhatinc

FOLLOW US ON TWITTER

www.twitter.com/redhatsummit

TWEET ABOUT IT

#redhat

READ THE BLOG

summitblog.redhat.com

GIVE US FEEDBACK

www.redhat.com/summit/survey