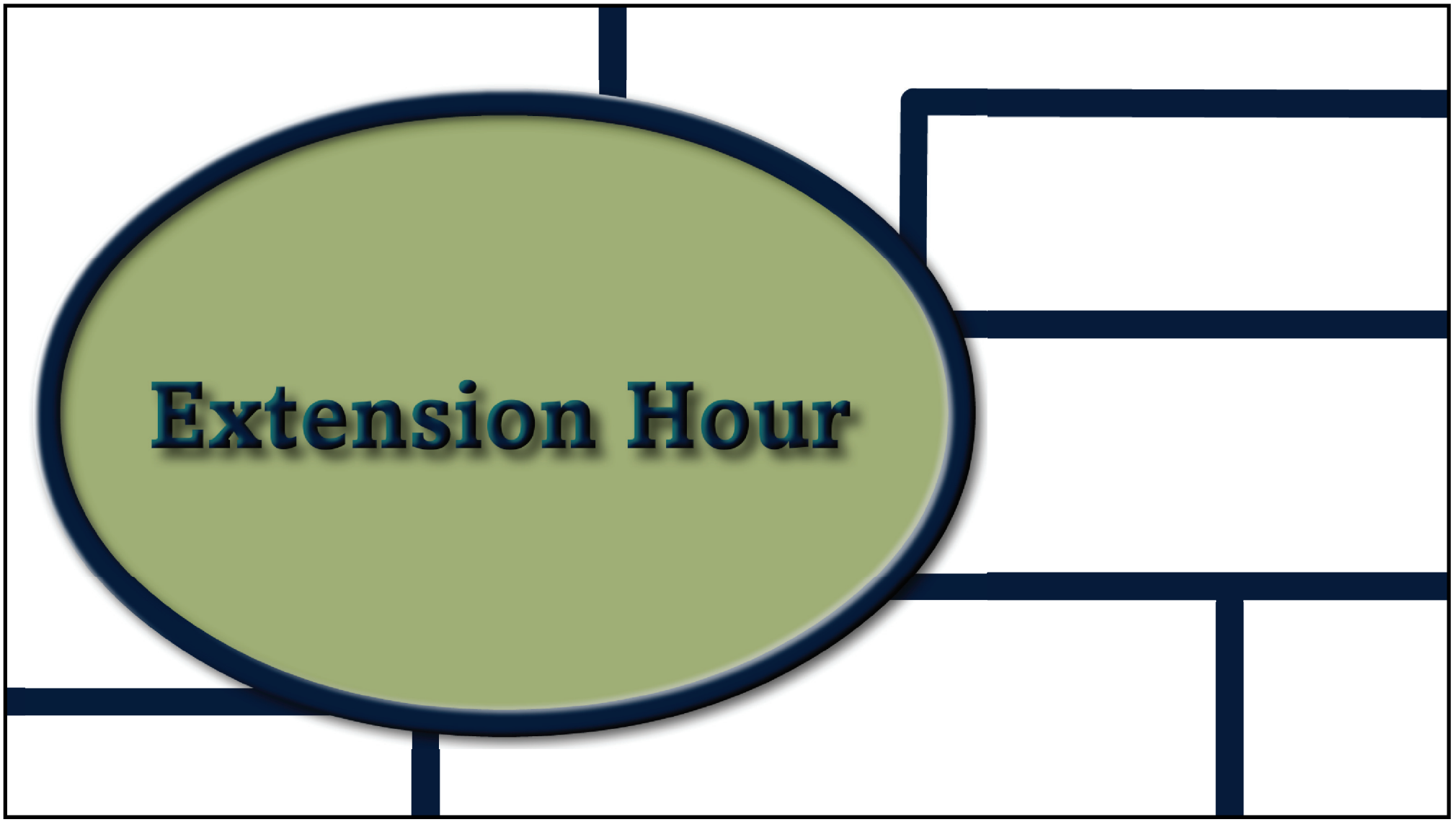

Extension Hour

Using CDI and Extensions to make a better Java EE

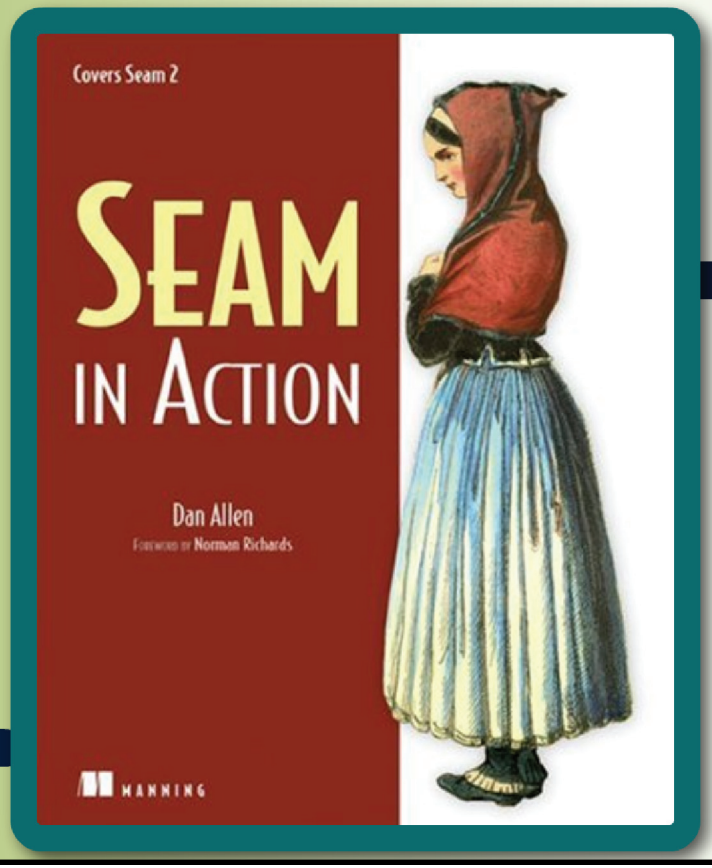
Dan Allen

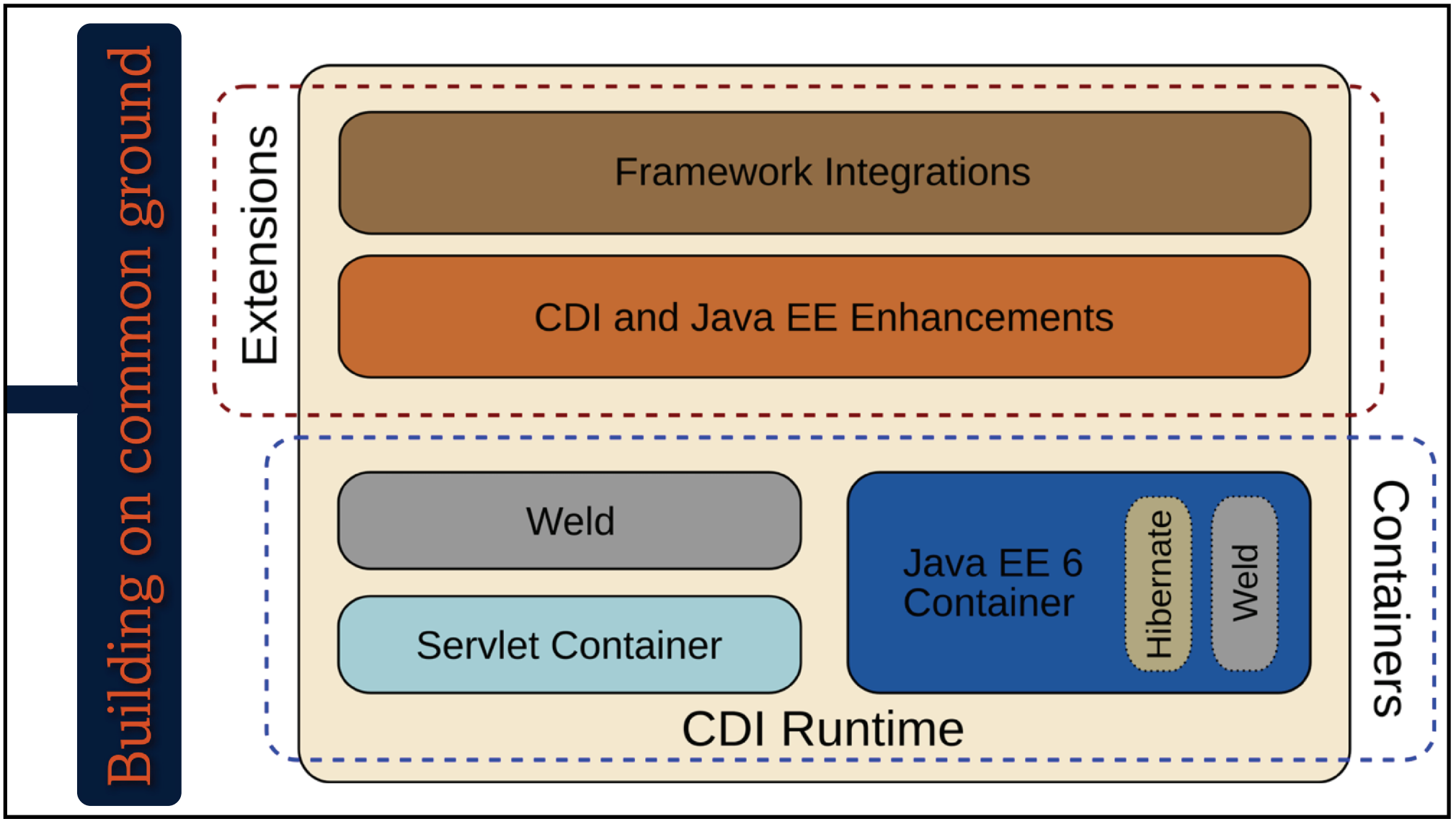
Principal Software Engineer

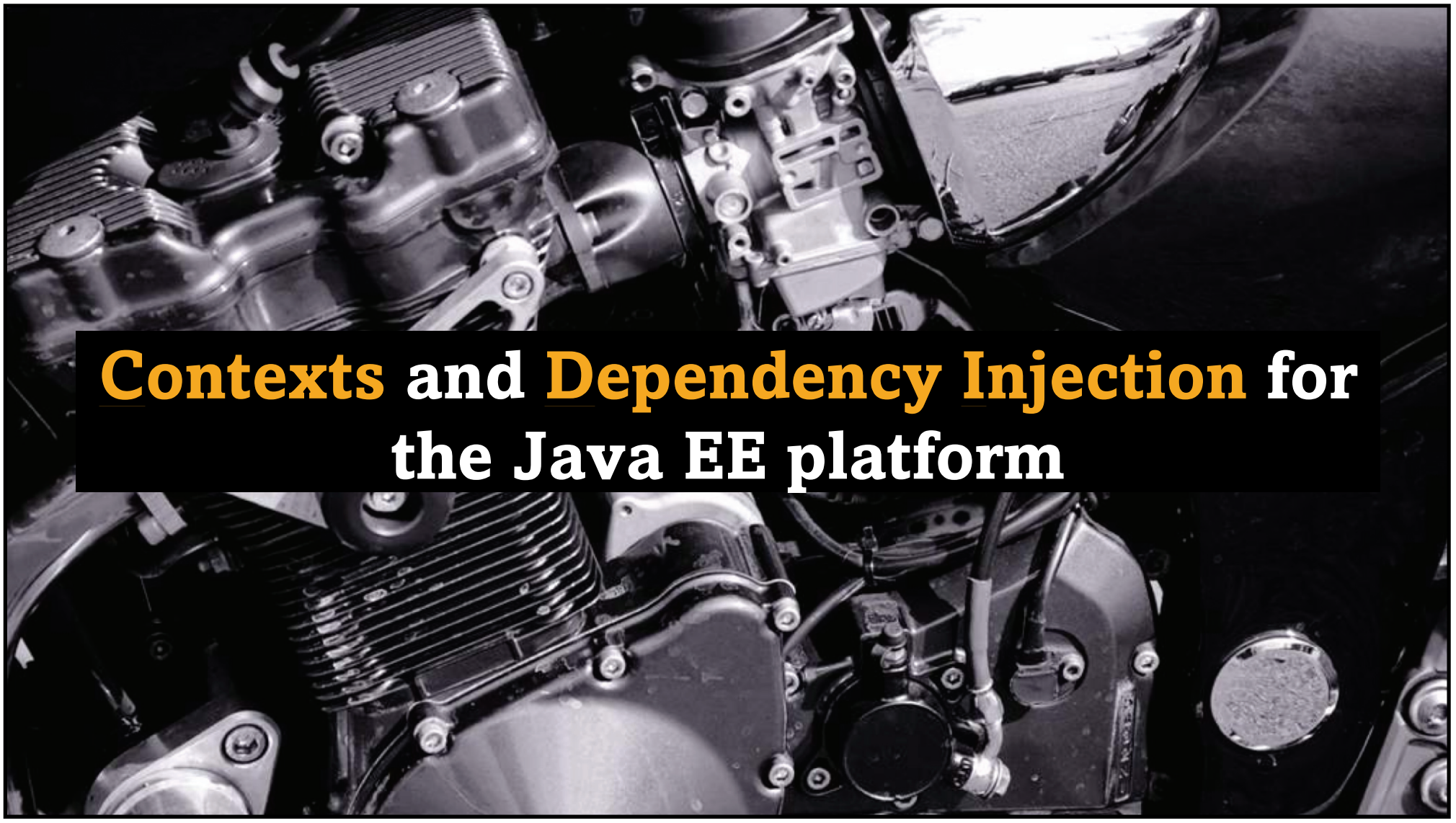
Seam Community Liaison

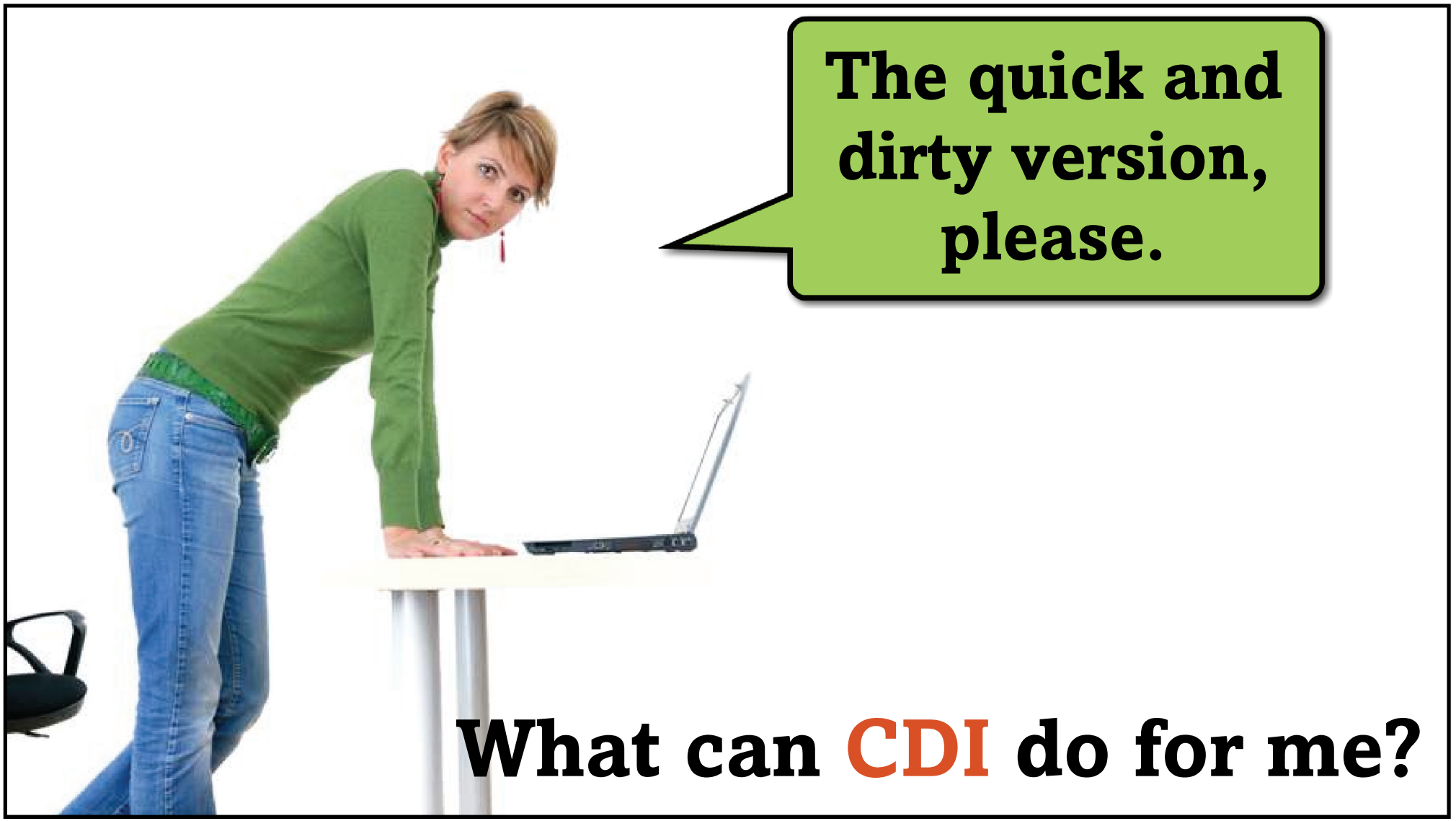
JBoss Community

@mojavelinux




Building on common ground
Extensions
Framework Integrations
CDI and Java EE Enhancements
Containers
Weld
Servlet Container
Java EE 6
Container
Hibernate
Weld
CDI Runtime


Contexts and Dependency Injection for
the Java EE platform


The quick and dirty version, please.
What can CDI do for me?

CDI can

- manage a bean's lifecycle
- inject dependent beans
- bind a bean to its context and name (if provided)

Bean class

Set of bean types

Set of qualifiers

Scope

Bean name (optional)

Set of interceptor bindings

Alternative classification

"When is a bean
recognized?"

"How are injections resolved?"

Types

Type parameters

Annotations

Method signatures

"What makes CDI different?"

STANDARDIZED

TYPE-SAFE

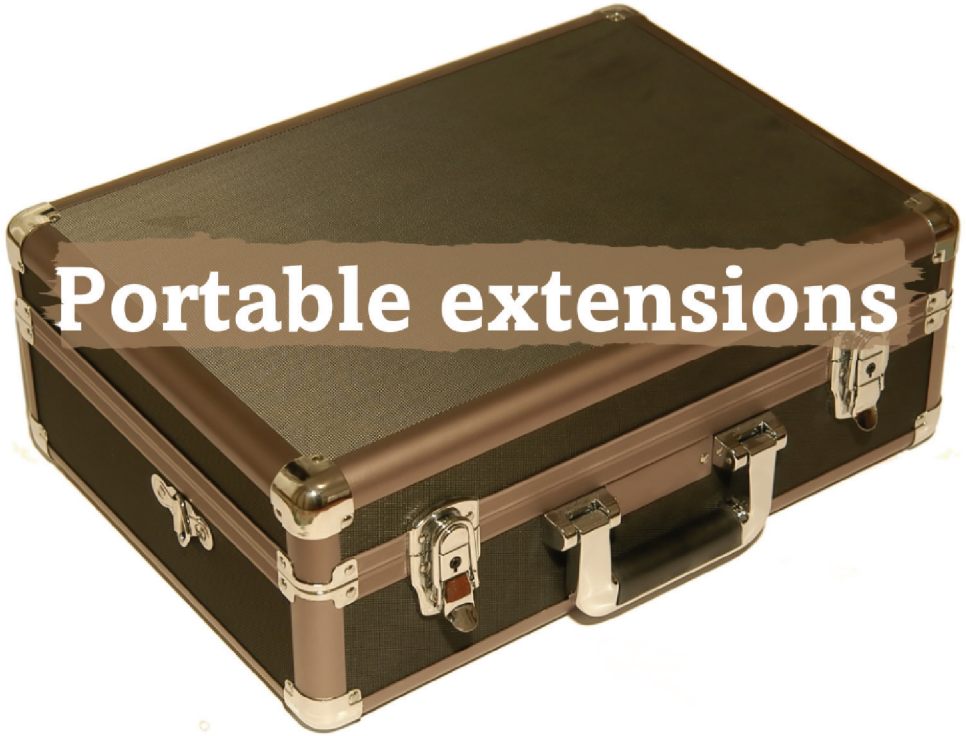
EXTENSIBLE

"All we have to decide is what to do with the beans that are given to us."

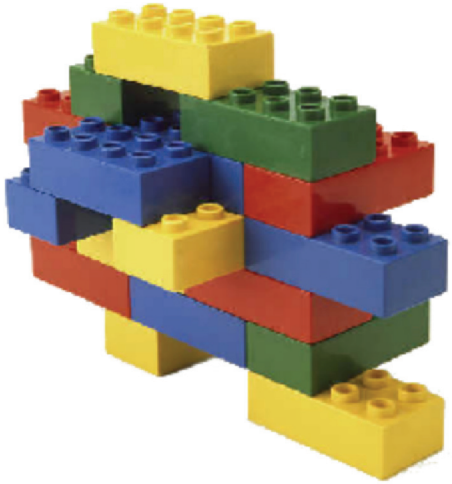
CDI also provides

- bean producers
- bean alternatives
- an event notification model
- semantic interceptor bindings
- bean decorators

Portable extensions



SPI for hacking Java EE to:



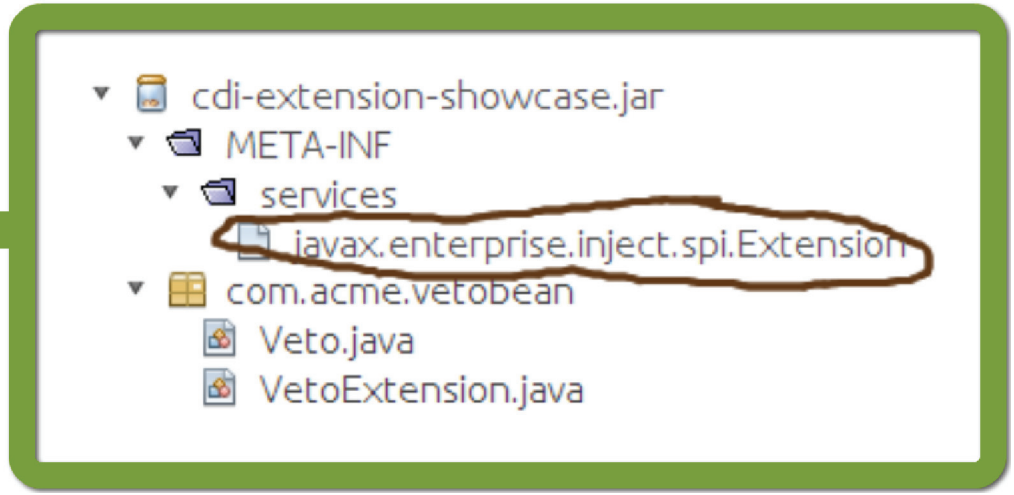
- ☑ register additional beans
- ☑ satisfy injection points
- ☑ introduce custom scope and context bean store
- ☑ augment or override bean metadata

"How does it work?"

Based on Service Provider Interface (SPI) pattern

1. Implement empty Extension interface
2. Observe container events to alter deployment
3. Register extension

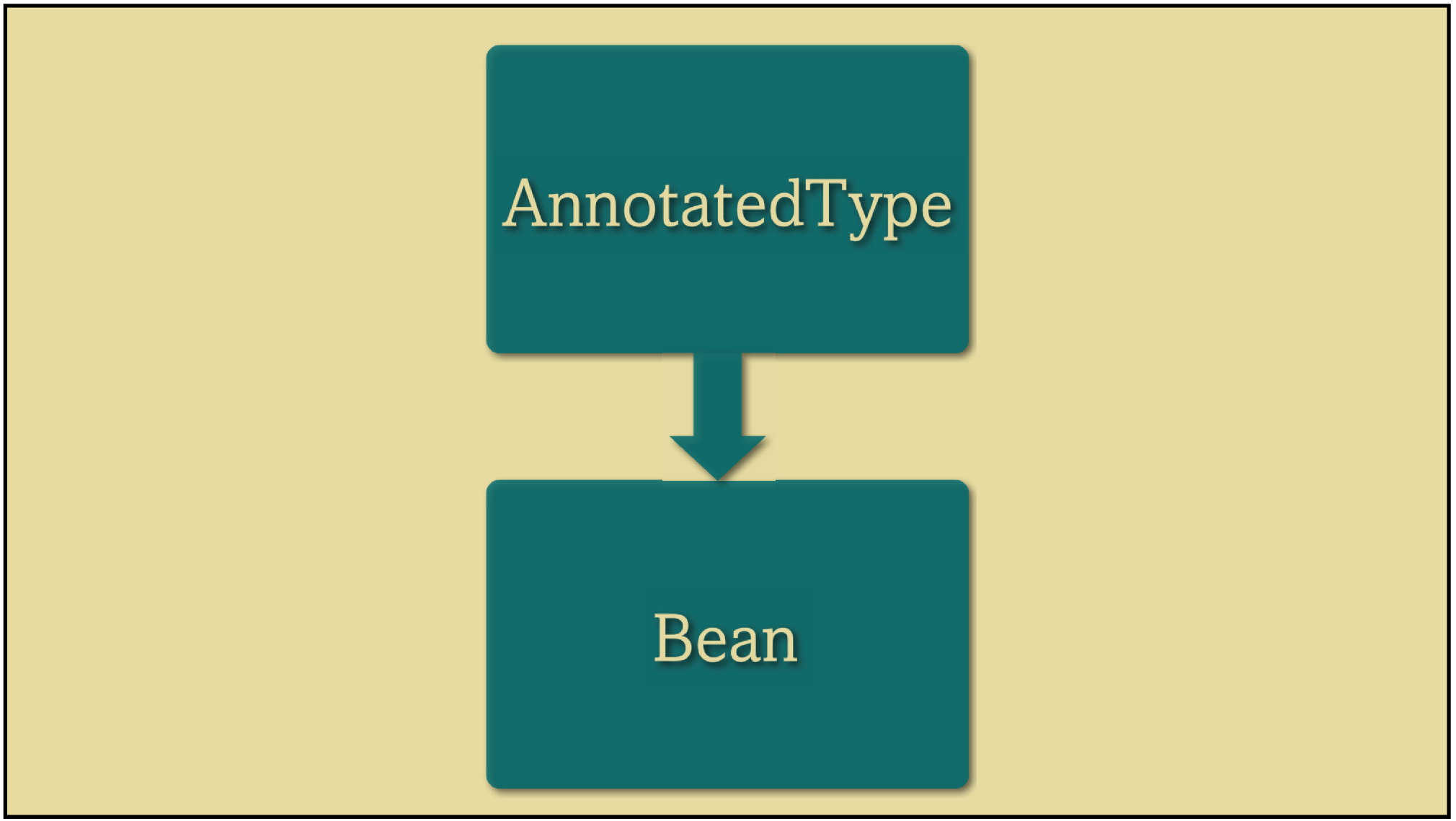
"When is an extension recognized?"



META-INF/services/javax.enterprise.inject.spi.Extension

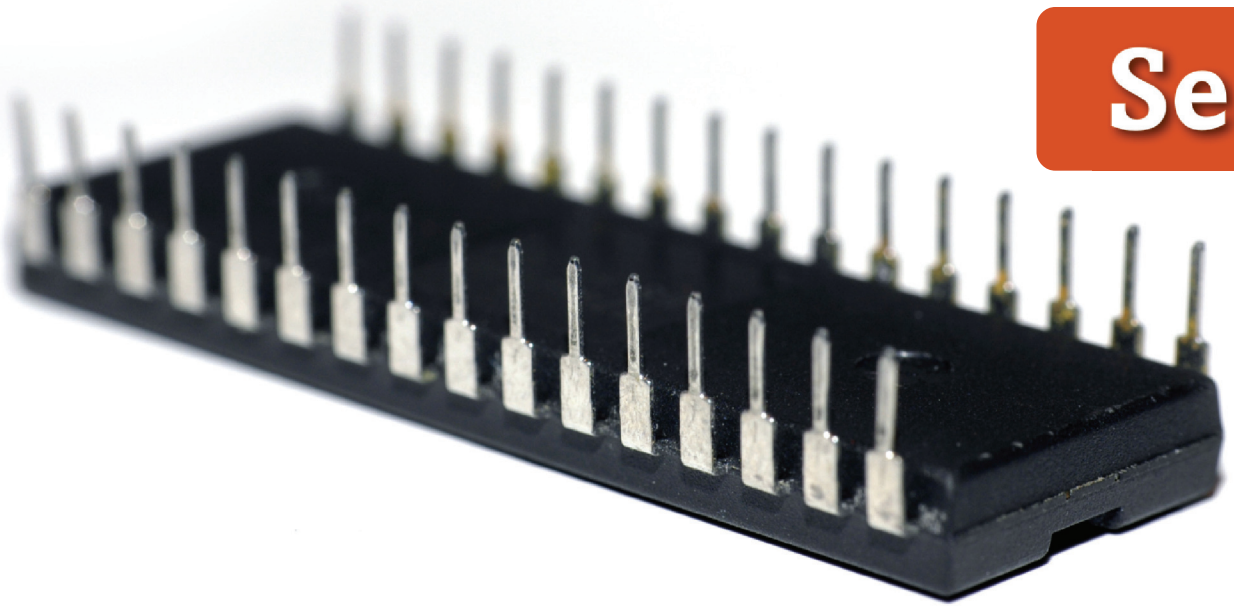
com.acme.vetobean.VetoExtension

BeforeBeanDiscovery
ProcessAnnotatedType
ProcessInjectionTarget
ProcessBean
ProcessObserverMethod
ProcessProducer
AfterBeanDiscovery
AfterDeploymentValidation
BeforeShutdown


AnnotatedType
Bean

Ah, but can we test it?

A testing framework for CDI (and more)



Seam Solder

Generally useful stuff for CDI programming, extension authoring and framework integrators

AnnotatedType builder

Bean builders

Annotation inspectors

Annotation literals

Reflection utilities

Method injector

Resource loading

Seam Config: XML-based bean metadata

define

customize

wire

"Type-safe"

- Reference fully qualified types
- Import packages via XML namespaces

Bean registration control

Tweaking an injection point

Annotation aliasing

Implicit object producer

Event bridge

Interceptor bindings

Declarative behavior via interfaces

Additional scopes

Framework integration

Reinventing Logging

Category as annotation
Exception codes
Type-safe loggers
Language support
An abstraction layer with fresh ideas

Addresses real-world scenarios
Developers code in Java
Translators edit message bundles

Who's writing extensions?

Seam 3 project - http://seamframework.org/Seam3

MyFaces CODI - http://www.irian.at/myfaces_codi

SoftwareMill - https://github.com/softwaremill/softwaremill-common

Now you are!

List your extension @

http://tinyurl.com/cdi-extensions

CDI combines loose coupling with strong typing.

CDI makes Java EE flexible, portable and extensible.

CDI extensions make Java EE competitive and awesome!

Contact Information

email: dan.j.allen@gmail.com

twitter: @mojavelinux

website: mojavelinux.com

JBoss blog: community.jboss.org/people/dan.j.allen/blog