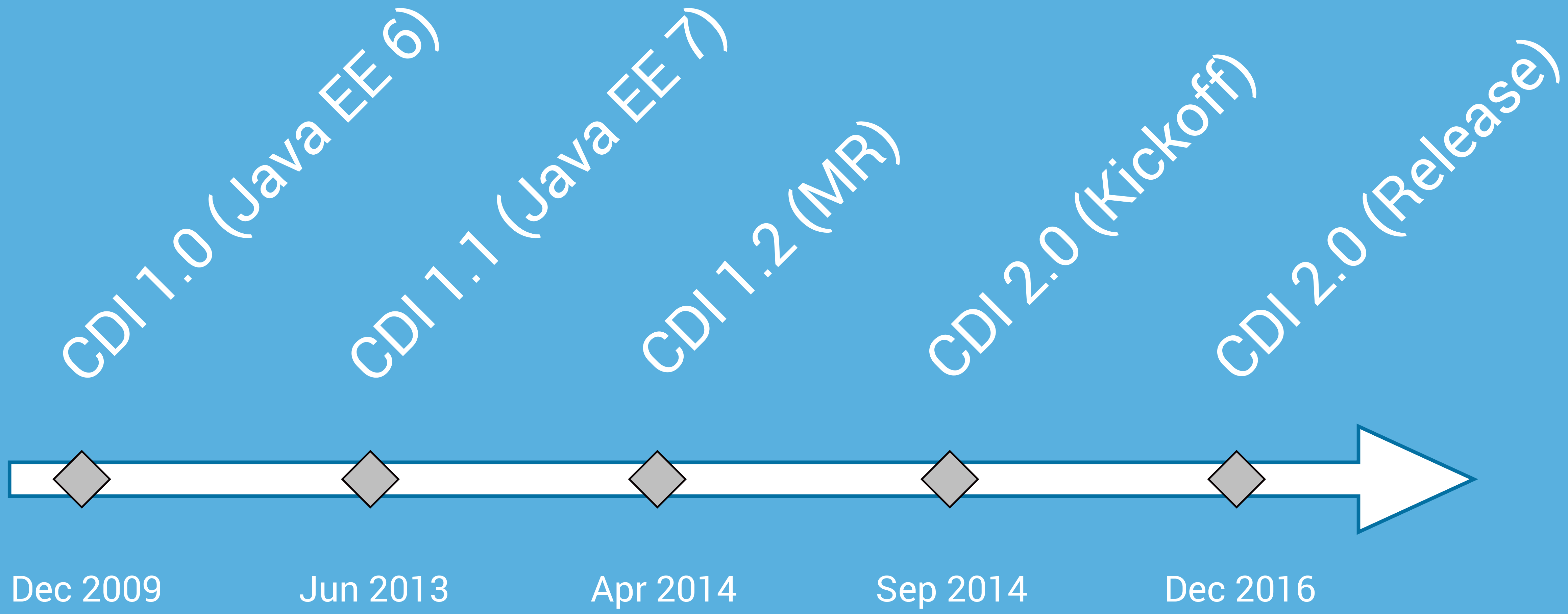
What's New in CDI 2.0

JSR 365

Mark Paluch

- Software Craftsman
- Spring Data Engineer @ Pivotal
- EG for CDI 2.0 (JSR365)
- Open Source committer

CDI Timeline



Parts

CDI 2.0 Parts

- CDI core
- Java SE
- Java EE Integration
- Support implementations for two TCK modes

Asynchronous Events

Asynchronous Events

- Notify event observers asynchronously
- One or more different threads
- Decoupled from synchronous events

Firing Async Events

```
@Inject
private Event<MyEventPayload> event;


public void triggerEvent() {
    event.fireAsync(new MyEventPayload());
}
```

Observing Async Events

```
public void observer(@ObservesAsync MyEvent payload) {
    System.out.println("Yay, I'm called async!");
}
```

Asynchronous Events

- Exceptions and Synchronization with CompletionStage

- Active scopes: Request, Application

- Custom scopes depend on the implementation

Ordered Events

Ordered Events

- Add @Priority to event observers
- Aids observer ordering
- Total global order determined when the event is fired

Observer Ordering

```java
public void observer(@Observes @Priority(2500) MyEvent event) {
    System.out.println("Default priority");
}
```

```java
public void earlier(@Observes @Priority(2499) MyEvent event) {
    // yay! I'm first
    System.out.println("Notified before all other observers");
}


public void observer(@Observes @Priority(2500) MyEvent event) {
    System.out.println("Default priority");
}


public void later(@Observes @Priority(2501) MyEvent event) {
    System.out.println("Notified after all other observers");
}
```

Meta-Data Builder API

- Standardized API
- Beans, BeanAttributes, InjectionPoints, and ObserverMethods
- Builder and Configurator-style

Meta-Data Builder API

```java
public class MyExtension {

  public void afterBeanDiscovery(
                     @Observes AfterBeanDiscovery event) {

    event.addBean()
               .beanClass(DummyCDIProvider.class)
               .produceWith(() -> DummyCDIProvider::new)
               .addQualifier(new TransactionalLiteral());

  }
}
```

Java SE

- Bootstrap API
- Start/stop contexts

Bootstrap API

```java
public static void main(String[] args) {

    try(CDI<Object> cdi = CDI.current()) {
        cdi.select(MyApp.class).get().runMyApplication();
    }
}


public class MyApp{

    public void runMyApplication(){
        // ...
    }
}
```

Validation-Error Lifecycle Event

- Handle validation problems
- Suppression of problems
- Allow Proxying of classes with final methods

What else is in work?

- Java 8: Repeating annotations
- AOP on produced and custom beans

Get in touch

- Web: http://www.cdi-spec.org/
- Mailing list: cdi-dev@lists.jboss.org
- IRC: irc://freenode.net/#cdi-dev
- Twitter: @cdispec
- Github: https://github.com/cdi-spec
- CDI 2.0 JCP page: http://jcp.org/en/jsr/summary?id=365

Q&A