LIFERAY®

CDI and OSGi

So happy together!

Raymond Augé
raymond.auge@liferay.com

Disclaimer

Everything in this presentation is subject to change at any time as it consists of in-progress specification work.

CDI - Key Goals

Develop a simple and intuitive model for existing CDI developers

Simplify OSGi challenges for new adopters

Take advantage of CDI's SPI as much as possible to ensure compatibility

Enable the simplest migration path for existing applications

LIFERAY

OSGi - Key Goals

Provide services (of all service scopes)

Optional, greedy/reluctant, static/dynamic, unary/n-ary, ordered service dependencies

Integration with Configuration Admin (& factory configurations)

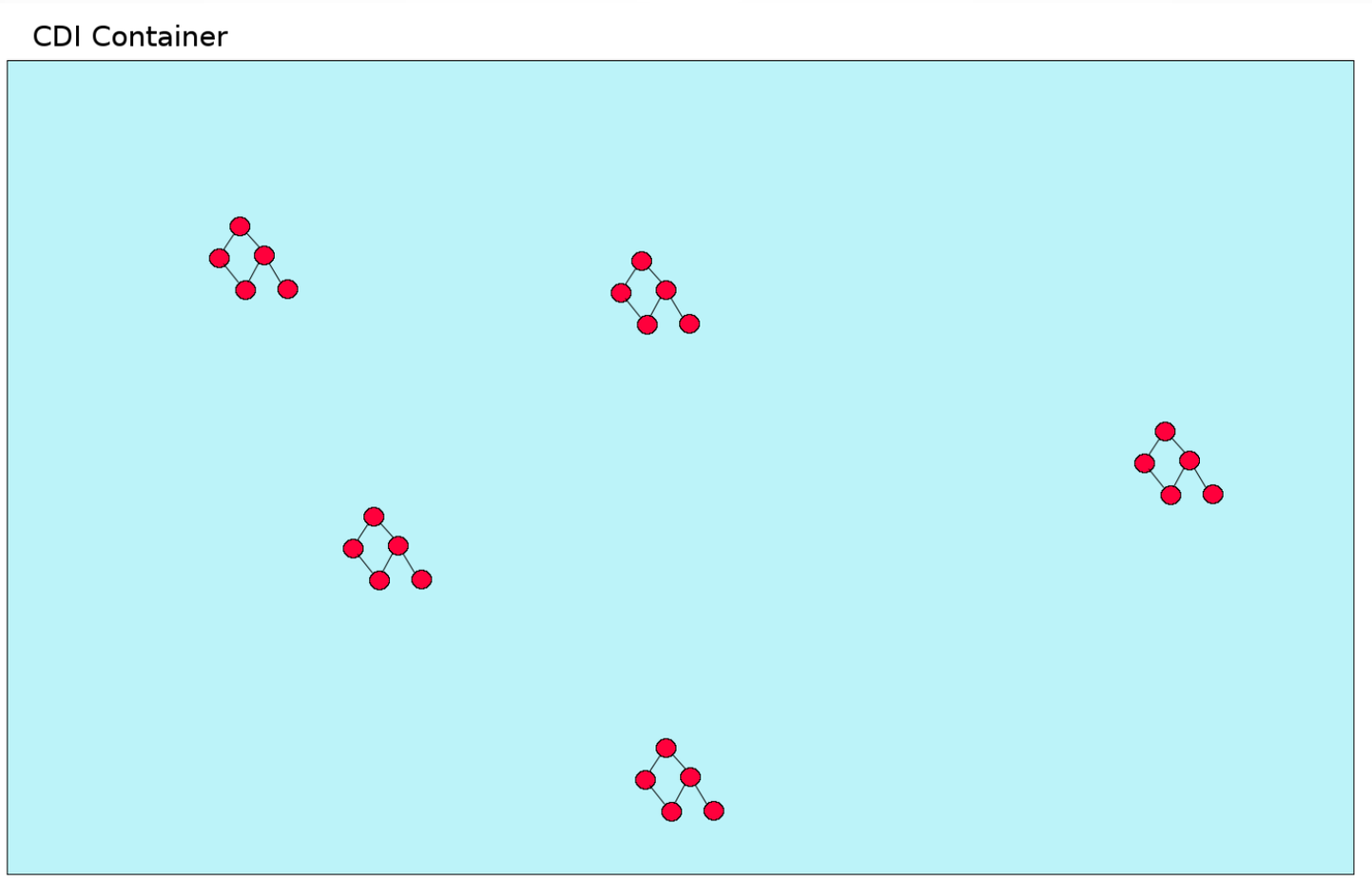
Lazy dependency management (**no damping!**)

LIFERAY

CDI **Container**

LIFERAY

CDI Container holds **Managed Beans** define graphs instances

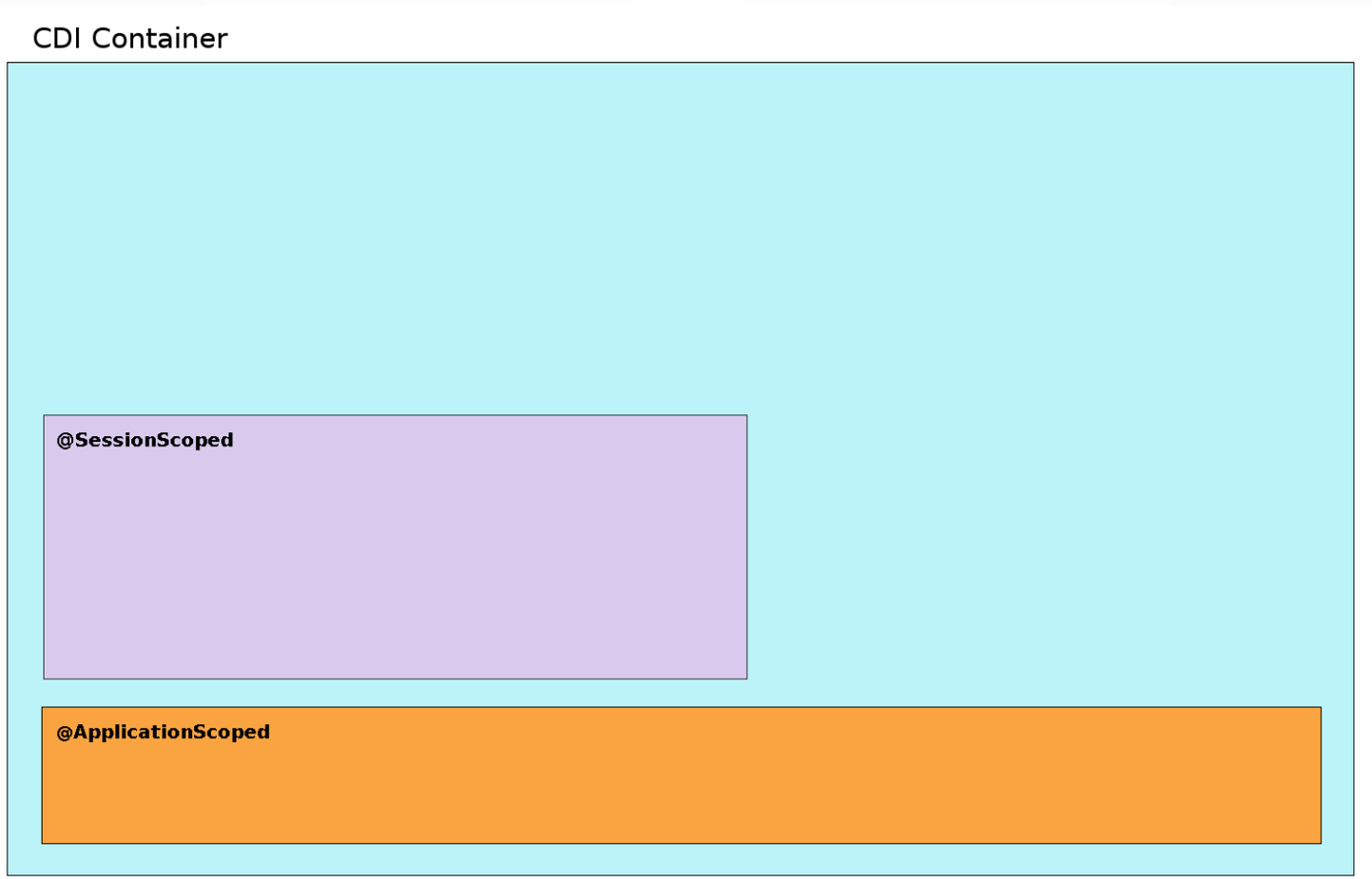


LIFERAY

Managed Beans have a **Scope**

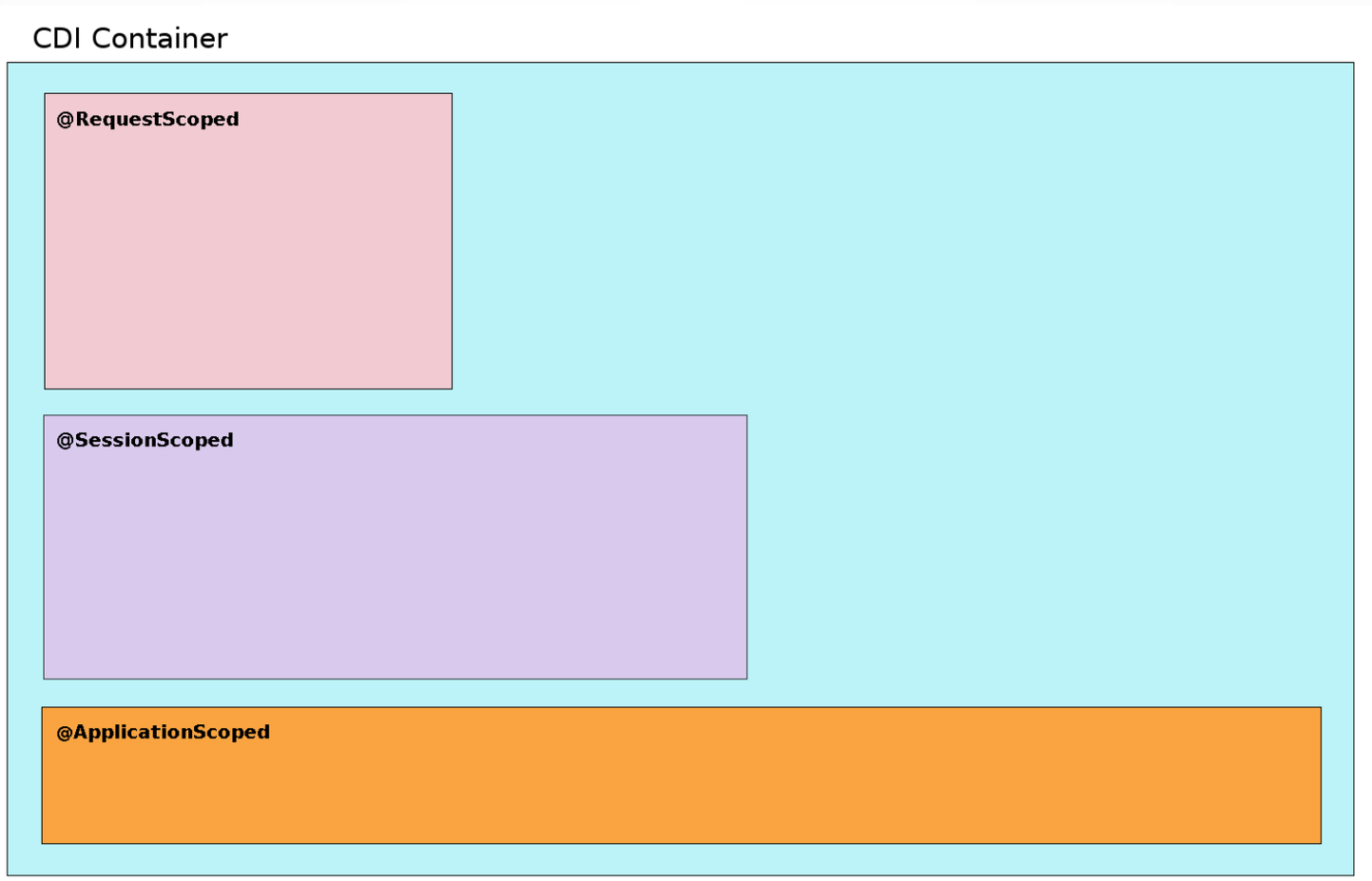
LIFERAY

Scopes define **Lifecycle** of managed beans



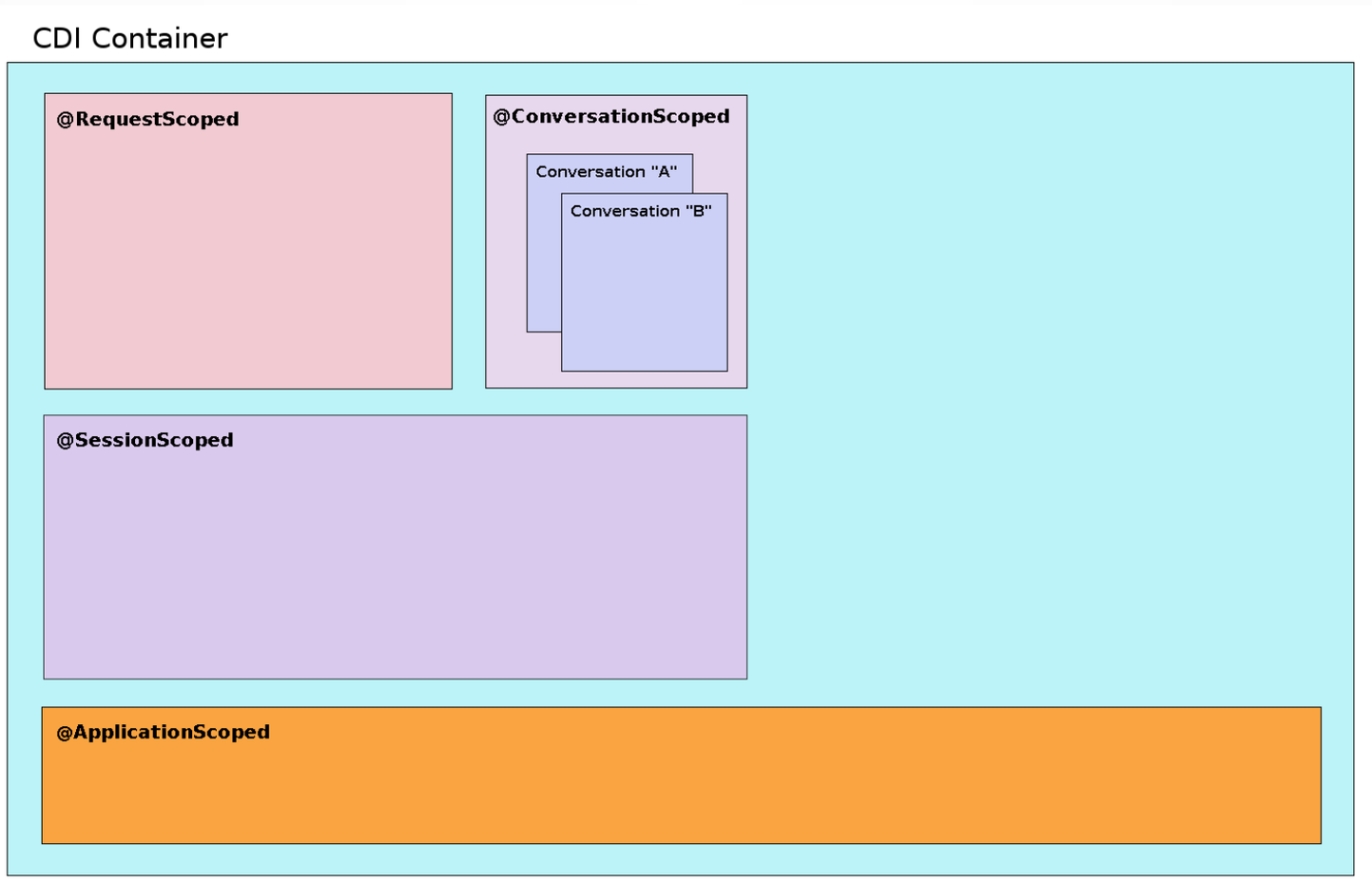
LIFERAY

Lifecycle follow edges of business logic resulting in **Contexts** which hold *contextual instances*



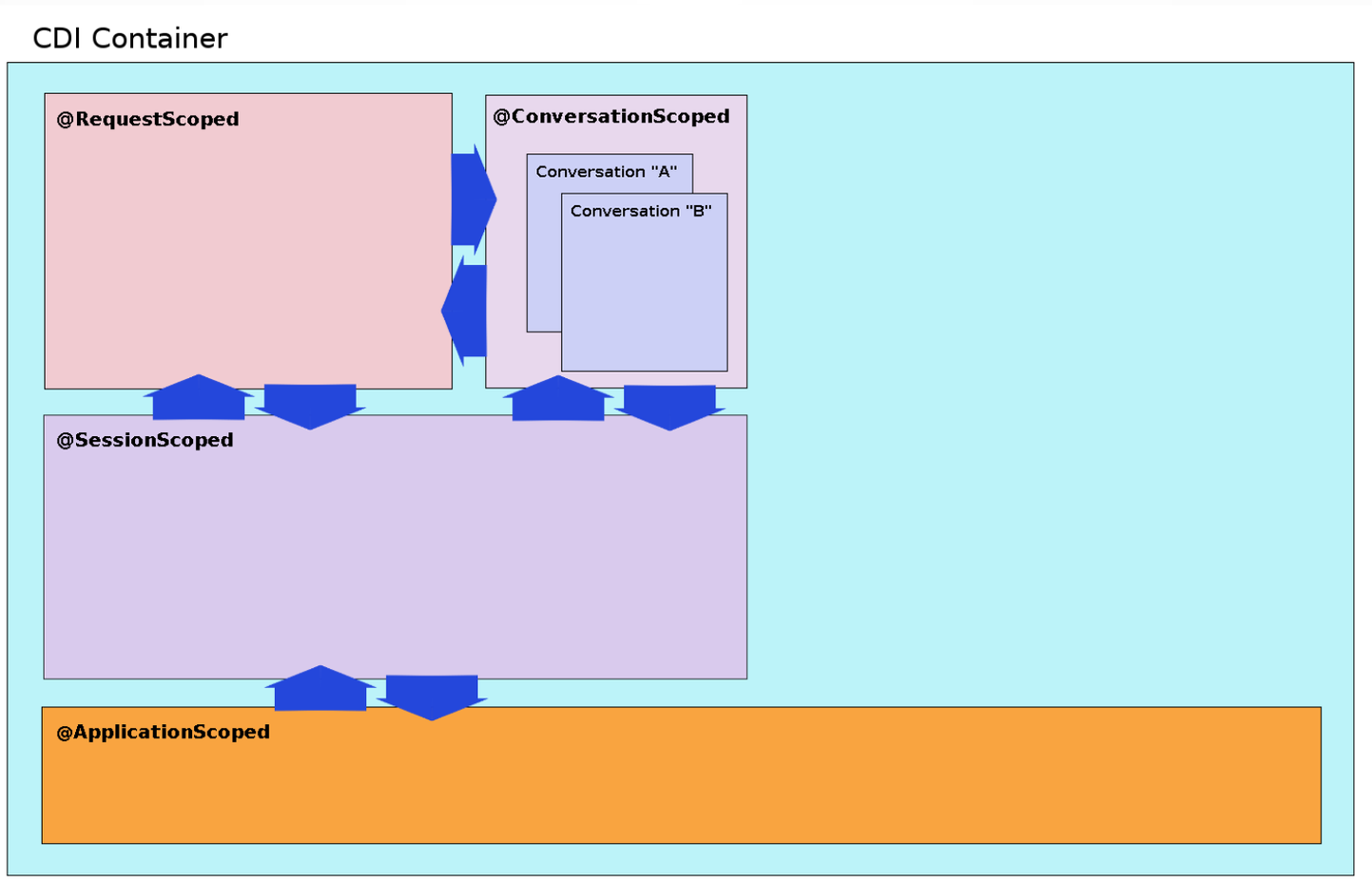
LIFERAY

Scope's lifecycle is guided by business **rules,** may include aspects like **identity**



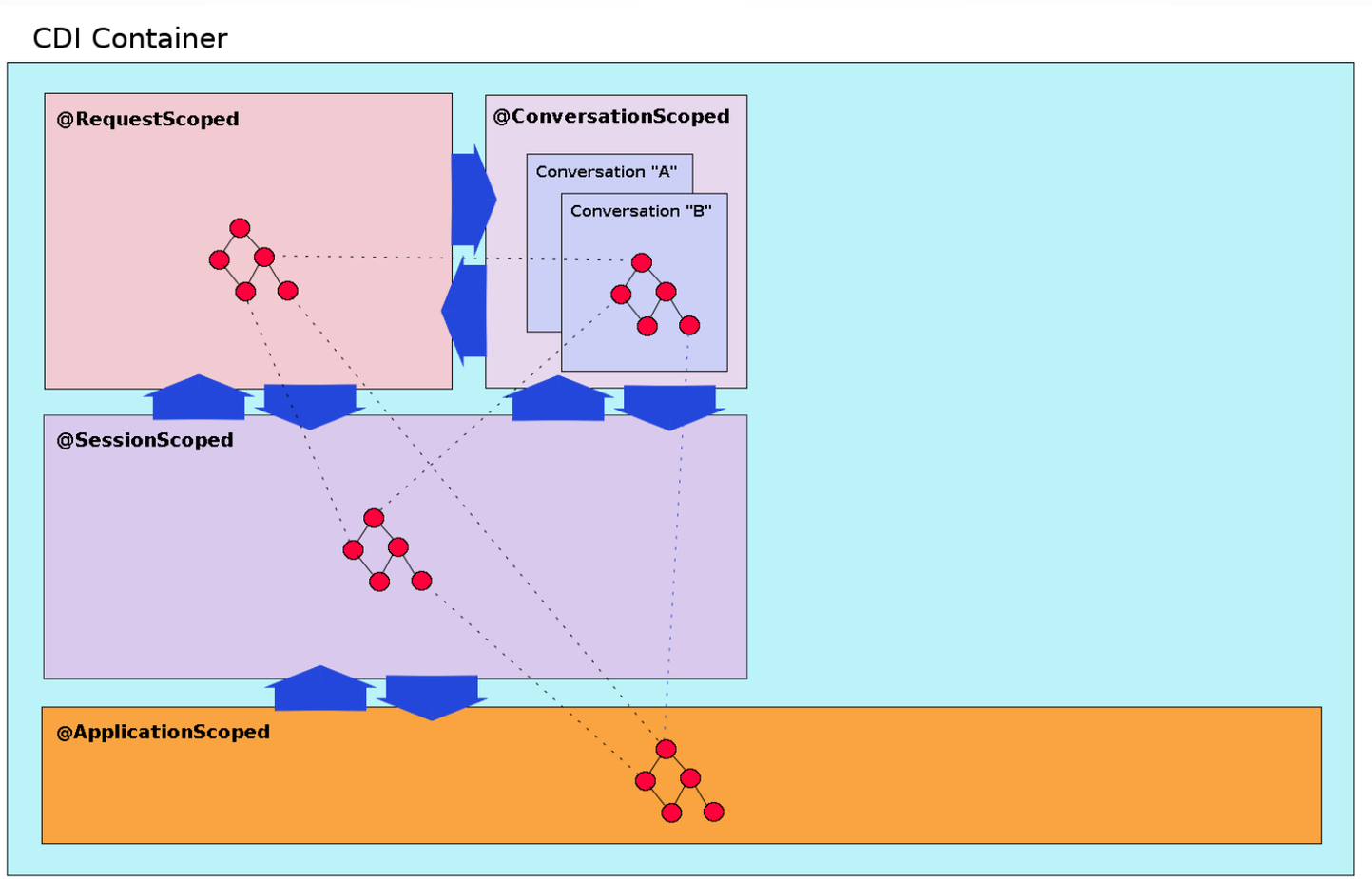
LIFERAY

Interactions defined by which scopes are **Active** as well as business rules



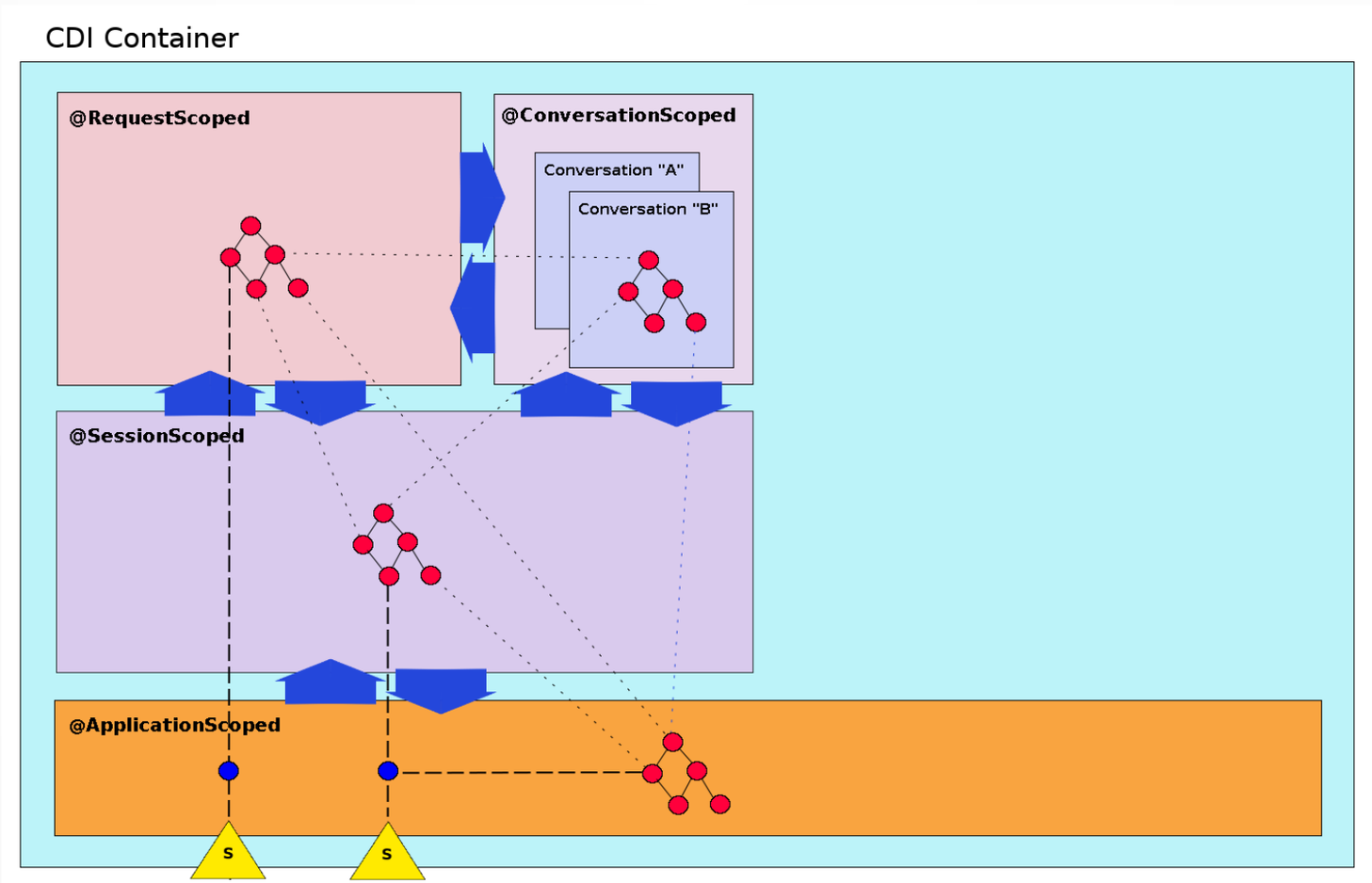
LIFERAY

Injection across scopes requires visibility



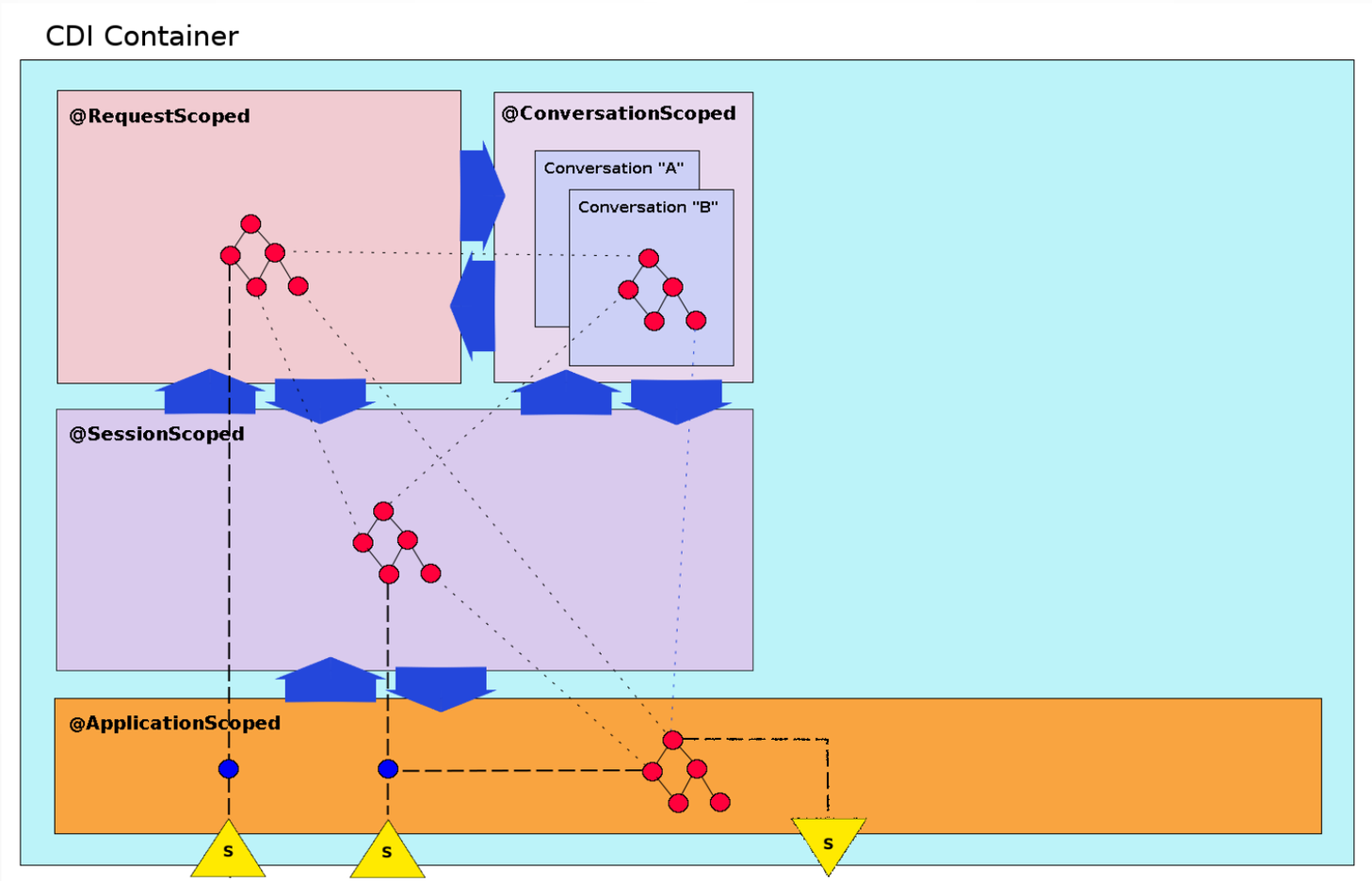
LIFERAY

Services are available to traditional scopes via *synthetic reference bean*



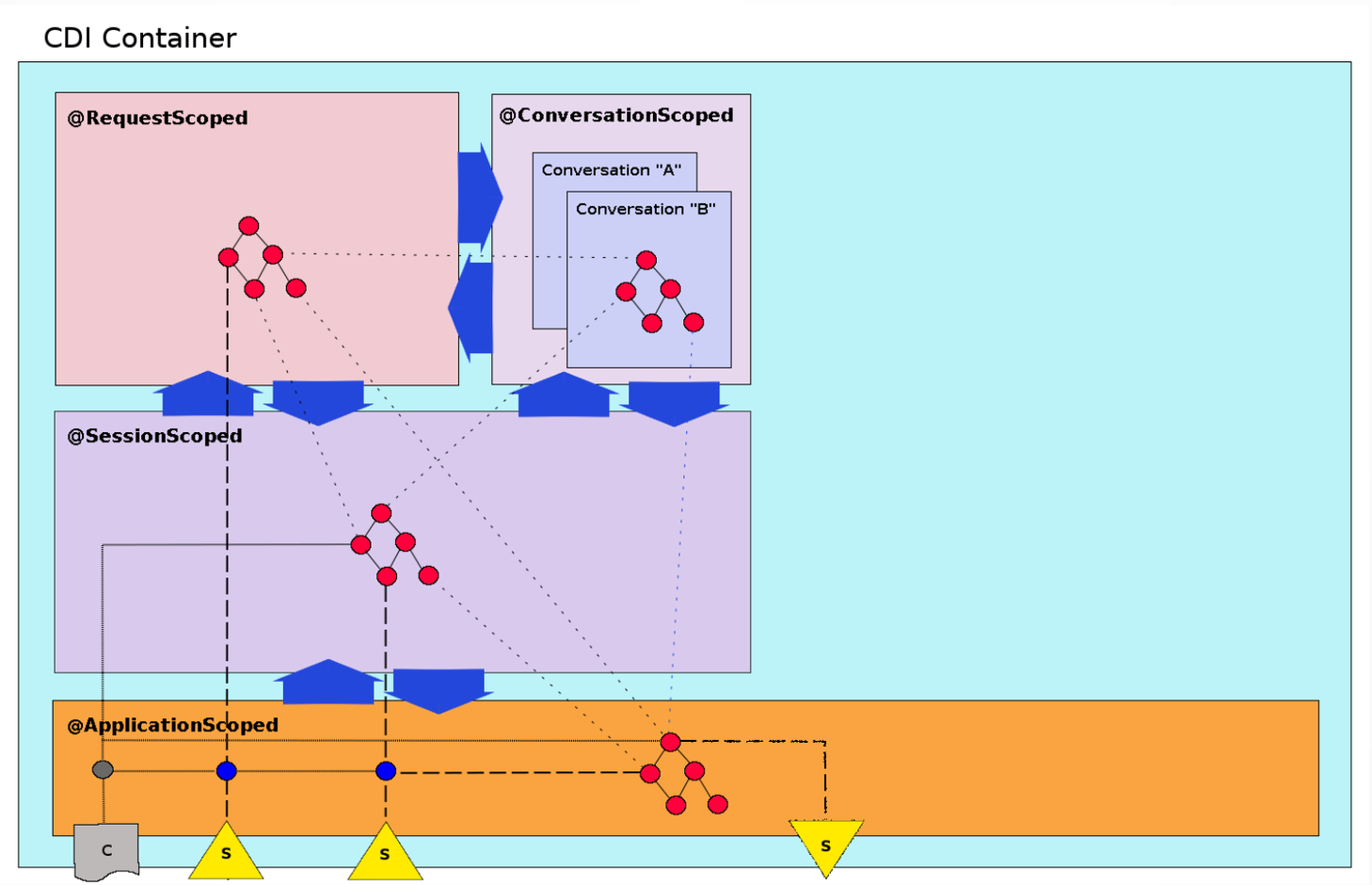
LIFERAY

@ApplicationScoped beans can **provide services**



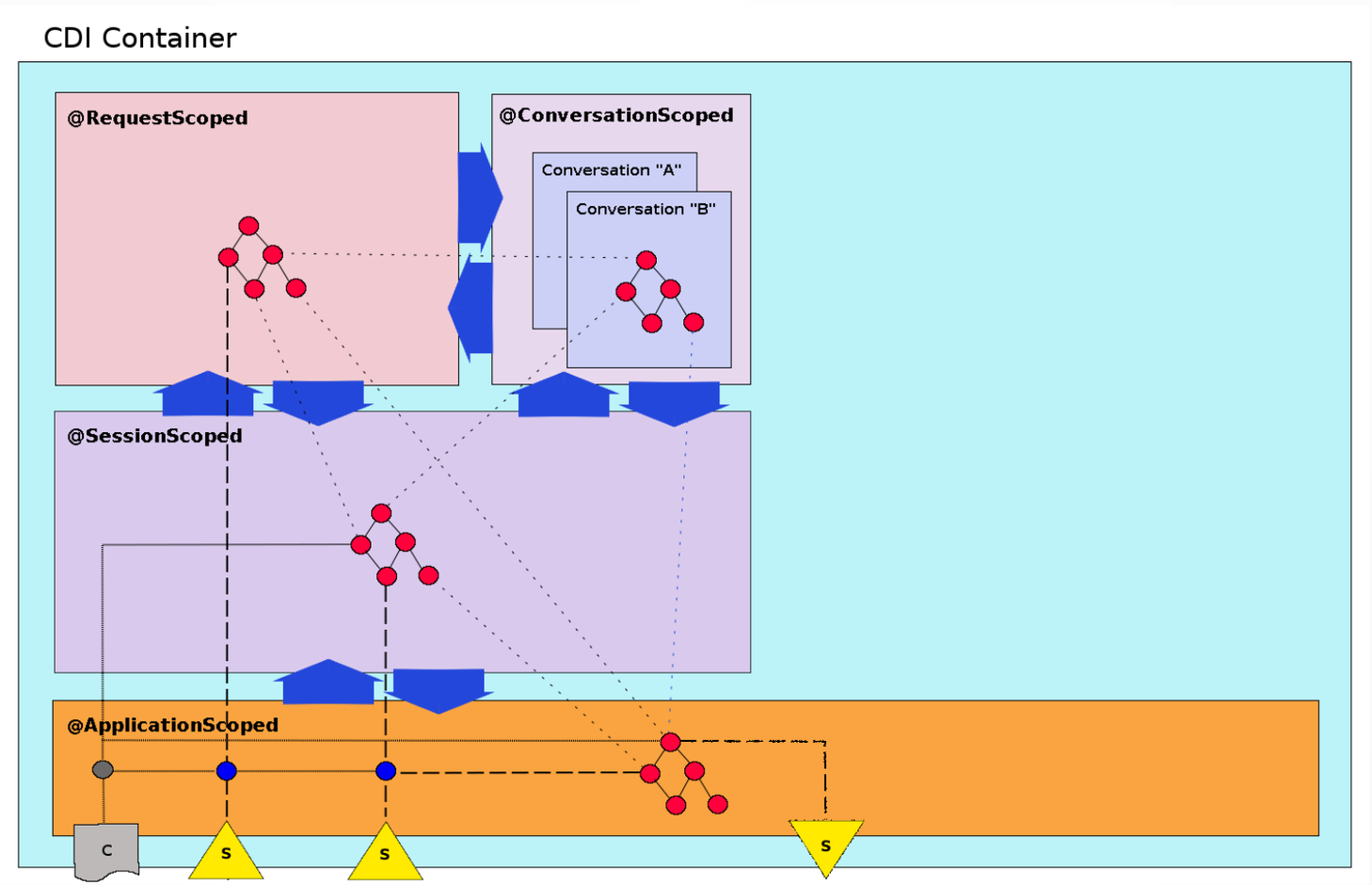
LIFERAY

Configuration available for direct injection, to configure references and published services



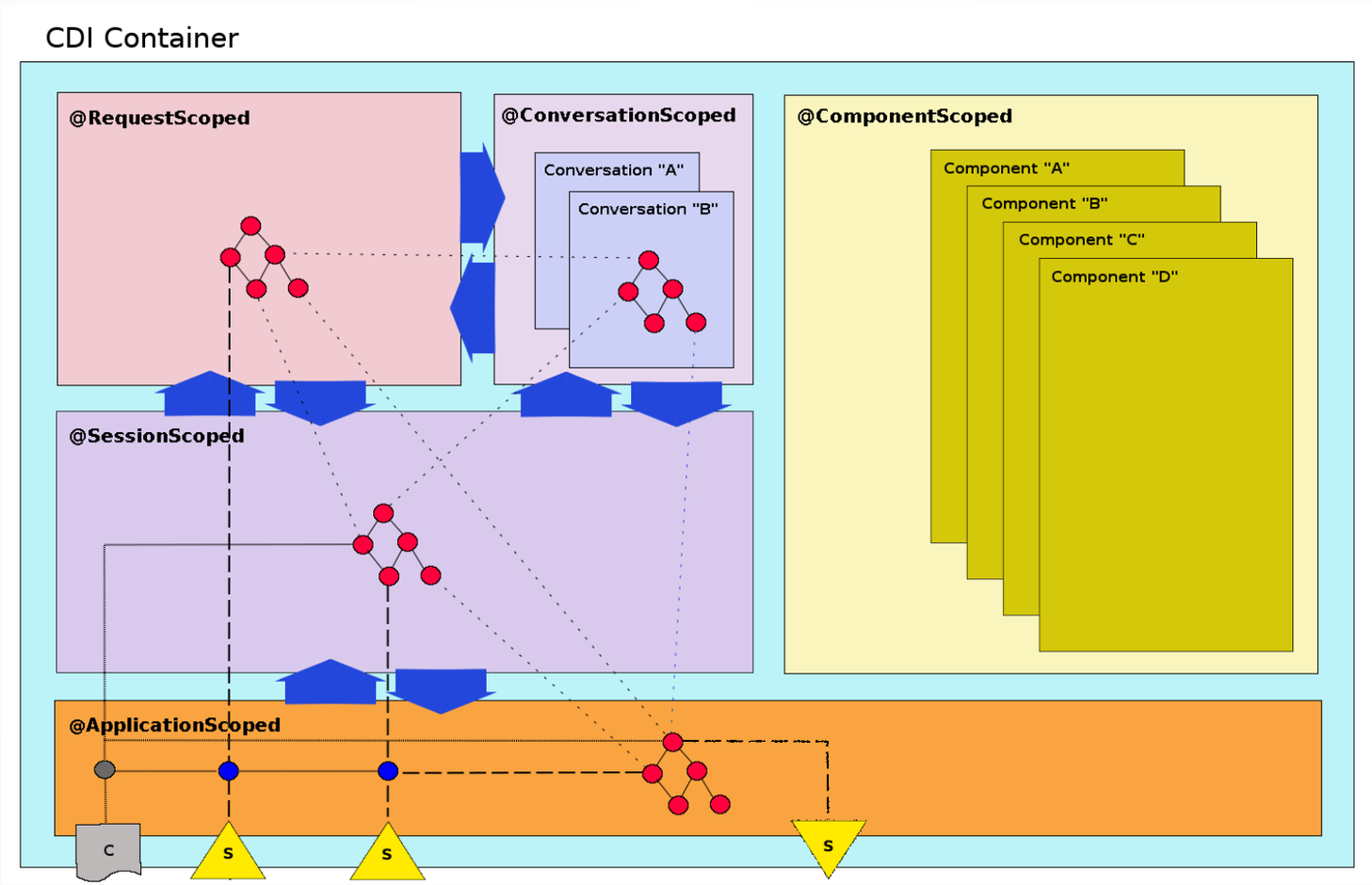
LIFERAY

Dynamism of dependencies affect this whole region of the container as a single component: **Application Component**



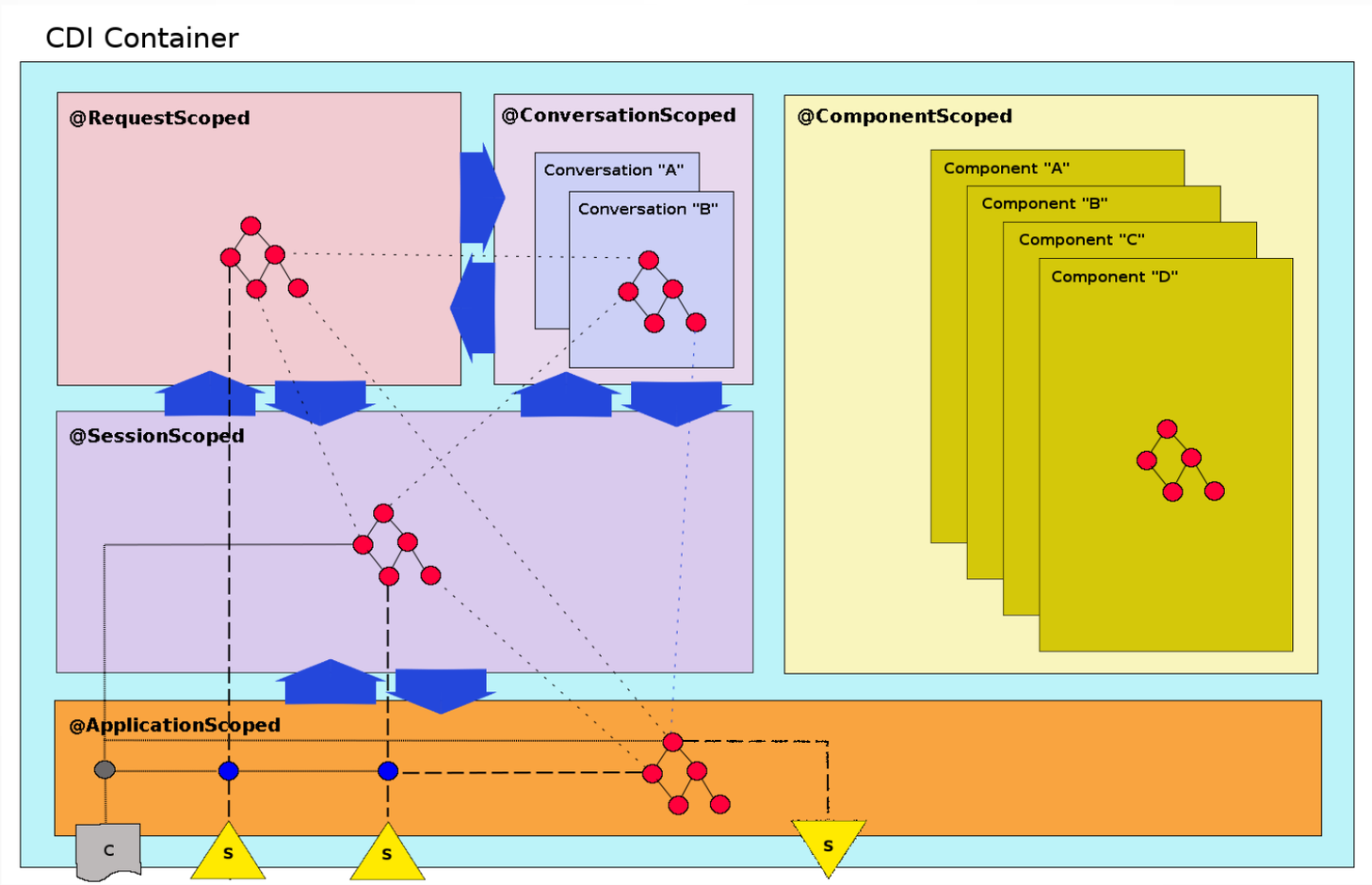
LIFERAY

Pseudo **@ComponentScoped** encapsulates OSGi's business rules



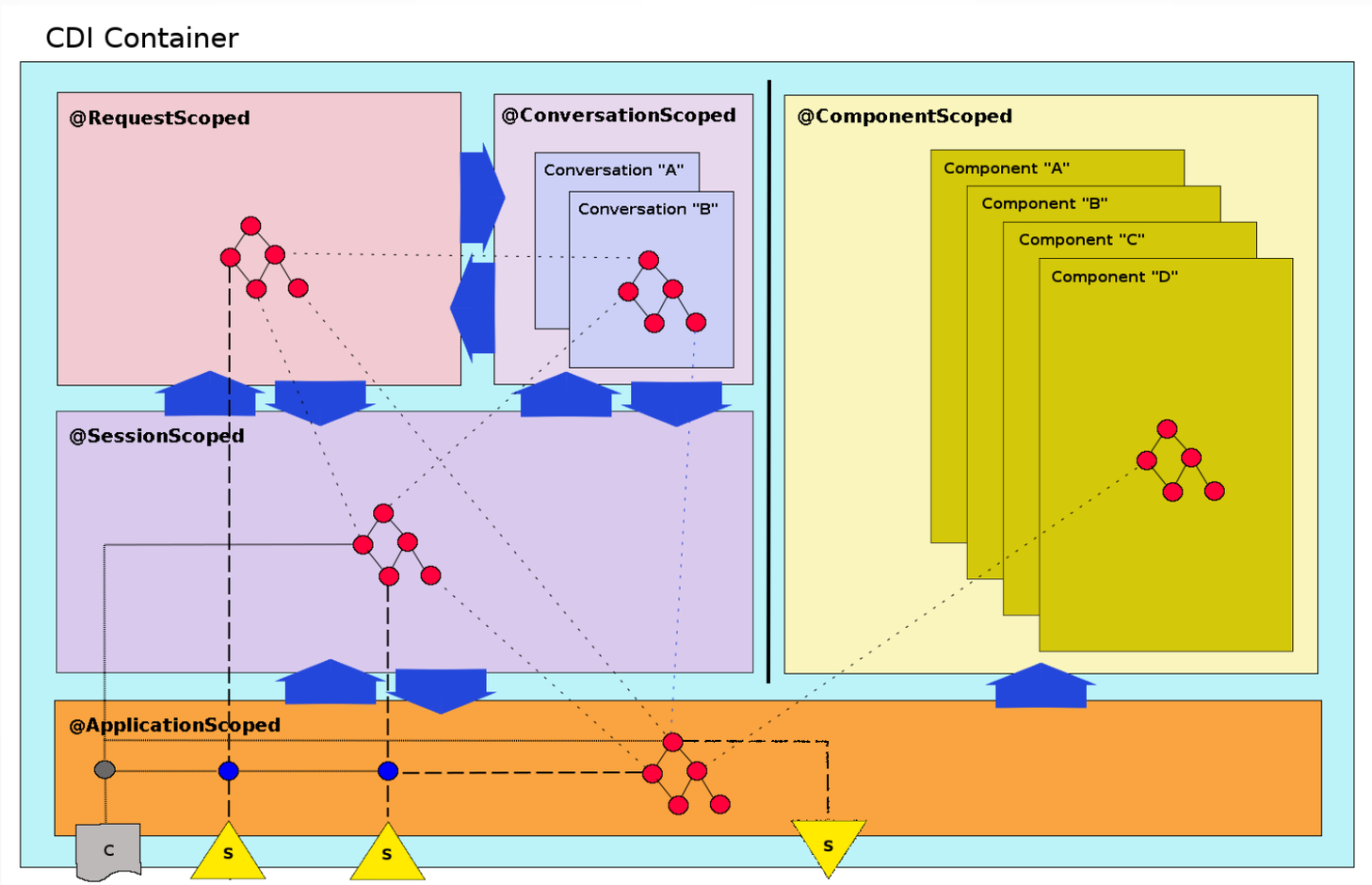
LIFERAY

@ComponentScoped beans form a graph of contextual instances within a context rooted in the **@Component** bean, *identified by name*



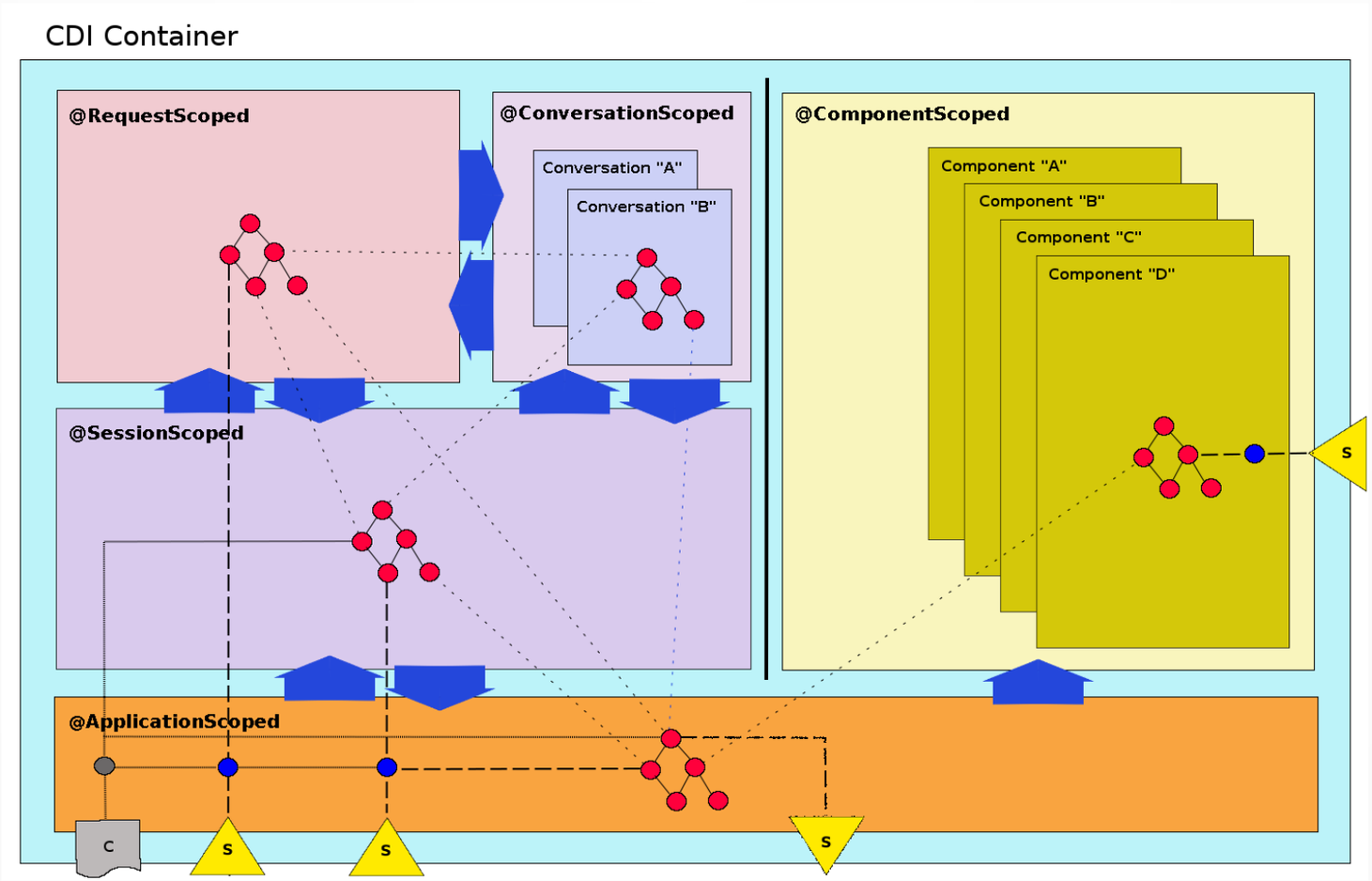
LIFERAY

Controlled visibility allows OSGi's integration to fully support **Dynamic Component Lifecycle**



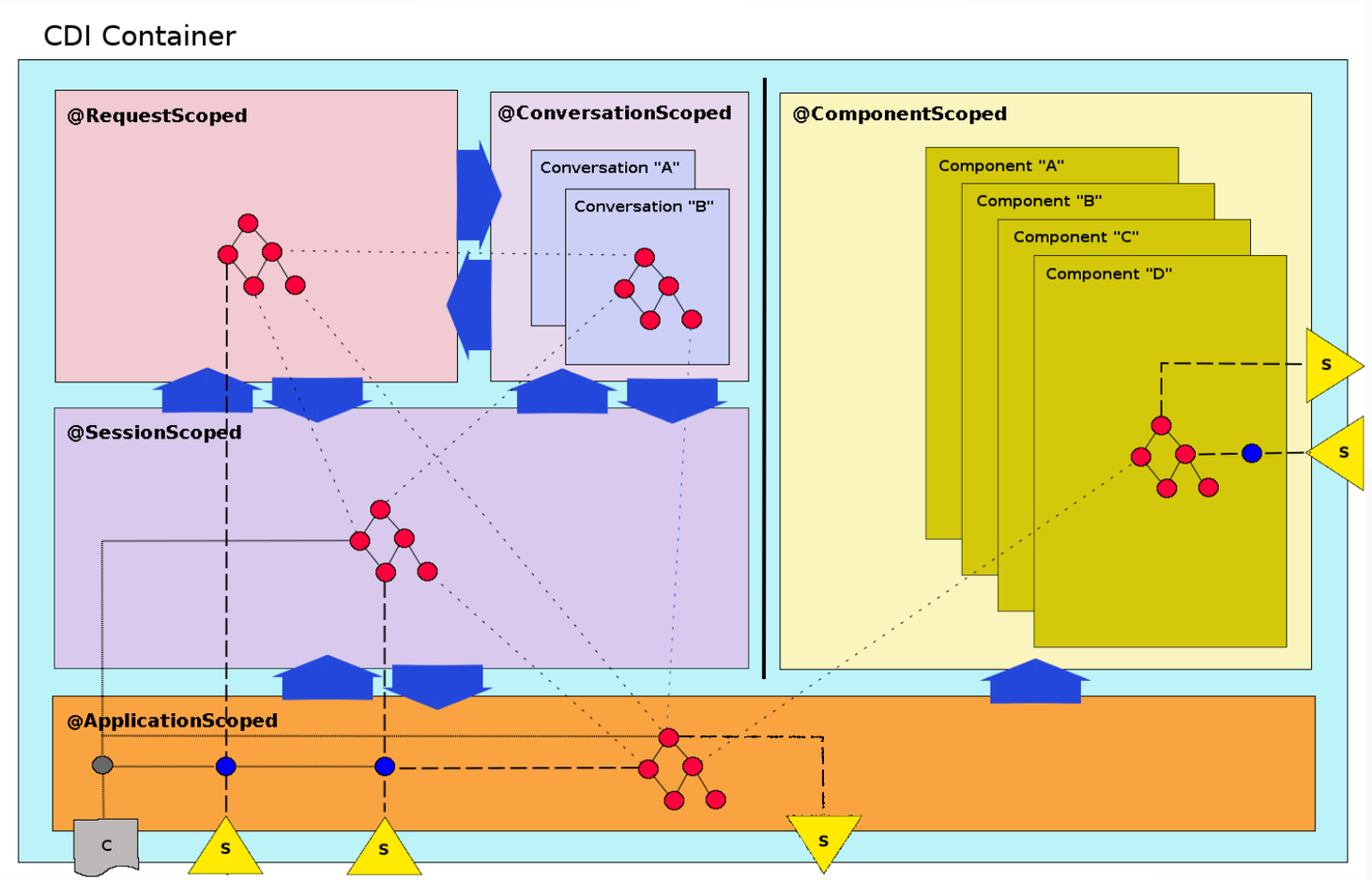
LIFERAY

@Reference to a service provided via synthetic bean scoped to the *component context*



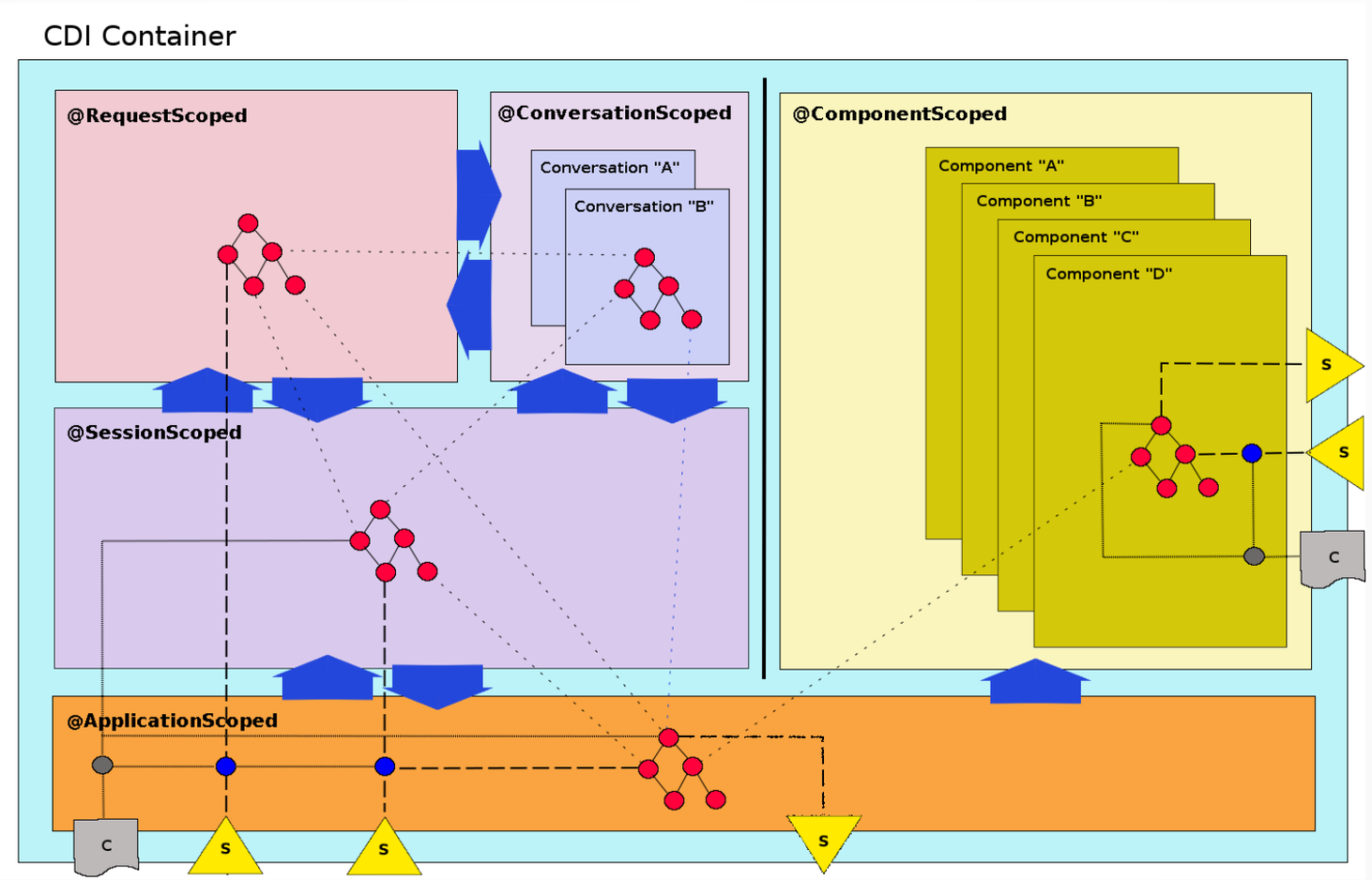
LIFERAY

@Component can provide services



LIFERAY

Configuration available for direct injection, to configure references and published services



LIFERAY

That's **a lot of detail**, but...

What does it look like?

LIFERAY

@Reference a service in a normal bean

```
@ApplicationScoped
public class E4_a {

    @Inject
    @Named("application.function")
    @Reference(target = "(foo=bar)")
    Function<String, Integer> function;

}
```

LIFERAY

Provide a **@Service** from regular bean

```
@ApplicationScoped
@Service
public class E2 {

    @Inject
    E1 e1;

}
```

LIFERAY

@Configuration @Property @Service

```
@ApplicationScoped
@Service
@Property("foo=bar")
@Property("foo=fum")
@ServiceRanking(200)
public class E3 {

    @Inject
    @Configuration
    Config config;

}
```

LIFERAY

Producer method, **@SessionScoped** instance, prototype service

```
public class E6 {

    @Produces
    @SessionScoped
    public Function<String, Integer> getFunction(
        @Named("session.function")
        @Reference(scope = ReferenceScope.PROTOTYPE_REQUIRED)
        ComponentServiceObjects<Function<String, Integer>> function) {

        return function.getService();
    }

}
```

LIFERAY

Optional @Reference

```
@ApplicationScoped
public class E9_a {

    @Inject
    @Named("application.function")
    @Reference
    Optional<Function<String, Integer>> function;

}
```

LIFERAY

@Component with @Reference & multi-configuration interests

```
@Component
@PID("com.foo")
@PID
public class C4 {

    @Inject
    @Reference
    Function<String, Integer> function;

}
```

LIFERAY

Factory component providing a service

```
@Component
@PID
@FactoryPID("com.factory")
@Service
public class C5 {

    @Inject
    @Reference
    Function<String, Integer> function;

}
```

LIFERAY

eclipsecon Europe 2017

OSGi™ Community Event 2017

Evaluate the Sessions

Sign in and vote at **eclipsecon.org**

- 1 0 + 1