

Pong EJSCS Scene Container Server

Version 0.1 - Reference Implementation

Contents

Project Overview	2
Overview of EJSCS Features, planned and implemented	2
Application Overview	2
What is EJSCS	3
What standard level of EJSC/JSC is supported in Pong?	3
Developer's Guide	4
Okay, seriously? What is this?	4

Project Overview

Note

This document is available in multiple formats. Some may be better formatted than others. If your copy appears distorted, please obtain either of these versions:

- PDF Document: [whitepaper.pdf](#)
- HTML Document: [whitepaper.html](#)

Knifa's Pong is an open source, partial EJSCS scene container server implementation that combines the functionality of a full SOA and EDA to some unquantifiable, subjective level, in order to create an agile, enterprise ready framework for fully leveraging none, some or all of JSR 287.

Note

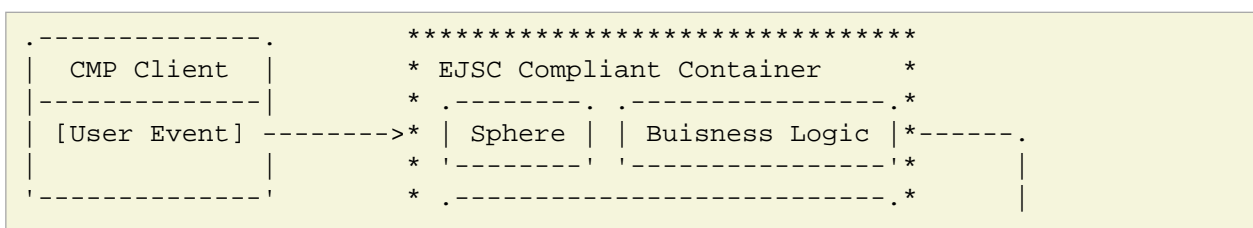
Pong EJSCS is currently under development, and as such, specifications and features below are not final and may change, possibly, but not necessarily, when the [JSR 292](#) specification (currently in 'public review' phase) is finalized.

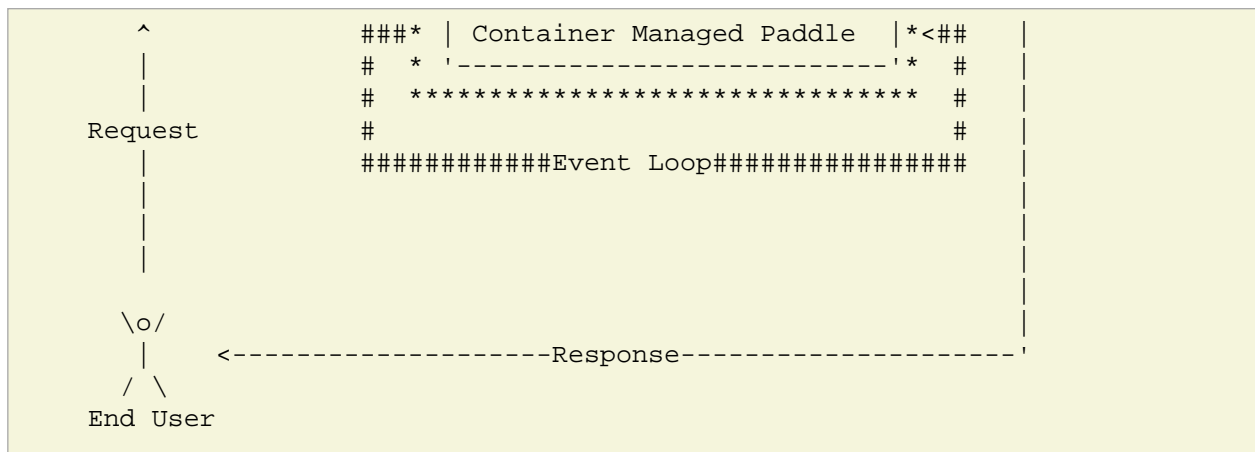
Overview of EJSCS Features, planned and implemented

- Fully Compliant with [JSR 14](#)
- Easy provisioning and hot deployment in an stopped state via XML middleware leveraging the Apache ant build tool
- Stateful and Stateless CMP (Container Managed Paddle) entity operation
- Support for programmatical, source level access to exposed EJB and JMX-based Web Services
- Ability to expose specific compilation units and source code to [JSR 199](#) compliant build tool (directly or indirectly) through XML-based Apache Ant build tools
- Will eventually provide a mechanism for visually observing and interacting with the running state of a simulation.

Application Overview

The EJSCS container provided operates under the following model:





As you may have noticed, it is very straightforward while remaining sufficiently enterprise-ready at all times.

What is EJSCS

EJSCS will eventually be the deployment standard in accordance to a yet to-be-allocated JSR. It will define how to properly design a system suitable for management of (E)JSC.

What standard level of EJSC/JSC is supported in Pong?

The following standards are supported as of the first specification draft:

- Standard /Java Sphere Caroming/ (tm) (JSC)
- /Enterprise Java Sphere Caroming/ (EJSC)
- Anything that passes [JSR 199](#) or alternatively, compiles without generating errors.

In fact, the Pong EJSCS is a full *Enterprise Java Sphere Caroming System Container*.

About EJSC(S):

The Enterprise Java Sphere Caroming specification defines an API and interface that is specifically designed to generate Java Enterprise-ready Entertainment Emotions (JEREE 1.3) from simulating the caroming (or bouncing) of a Sphere-like element between standard Container Managed Paddles.

The EJSCS (*Enterprise Java Sphere Caroming System Container*) specification defines the set of API needed to implement a working EJSC /container/, that allows the simulation to take place, with optional XML-based middleware and configuration layers.

Developer's Guide

Simply run *ant* in the project root. The *dist/* directory will contain executable artifacts generated from the source code.

Okay, seriously? What is this?

Personally, I ([Alexandre Gauthier](#)) really hate the enterprise bullshit that comes with Java, which is sadly, a rather decent programming language.

So I wrote this rather dead on pastiche of java enterprise components, XML abuse and no obvious "What is this project about and what does it /do/" section included.

Why? Mostly because:

1. Knifa's Pong game needed a readme, and I decided to step up to provide one.
2. I was bored and frustrated
3. It needed to be made Enterprise Ready.
4. I really enjoy writing reStructuredText documents.

So yeah, it's all in good fun, and really isn't anything more than a shitty Pong clone in Java.