Sindhu Saravanan

ContAINER DIAGRAM

Contents

[1. What is a Container Diagram? 2](#_Toc482205138)

[2. Intent 2](#_Toc482205139)

[3. Structure 2](#_Toc482205140)

[4. Containers 2](#_Toc482205141)

[5. Interactions 2](#_Toc482205142)

[6. System Boundary 3](#_Toc482205143)

[7. Motivation 3](#_Toc482205144)

[8. Audience 3](#_Toc482205145)

[9. Container Diagram 3](#_Toc482205146)

## What is a Container Diagram?

Container diagram provides the high level technology choices with a box and lines diagram

## Intent

A Container diagram gives insights into

* The overall shape of the software system
* The high level technology decisions
* How responsibilities are distributed over the system
* How do containers communicate with each other

## Structure

A simple block diagram with the key technology choices. The diagram typically shows web servers, application servers, standalone applications, databases, file systems and other pieces that make up the system. Also external dependencies can be shown

## Containers

Containers are the logical executables that make up the software system such as

* Web Server ( Apache HTTP Server, Microsoft IIS Server …)
* Enterprise Service Bus
* SQL Databases (Oracle, Sybase, Microsoft SQL Server …)
* NoSQL Databases (MongoDB, CouchDB …)
* Other Storage Systems (Azure Blob storage)
* File Systems
* Windows Services
* Web Services
* Stand Alone Applications
* Web browsers and plugins

For each container drawn on the diagram, please specify

**Name:** The logical name of the container

**Technology:** The technology choice of the container

**Responsibilities:** A high level of the container’s responsibilities

If you have a question as to whether to represent the container in a separate box, simply ask if that box will be deployed in a separate physical or virtual hardware.

## Interactions

Typically the inter-container communication is inter-process communication. It is useful to explicitly identify this and summarise how the interfaces will work. It is important to annotate the interactions

## System Boundary

If you include systems and users outside the scope of the system being built, it would be a good idea to draw lines to demarcate the system boundary.

## Motivation

Context diagram is a high level overview, whereas the container diagram opens the system and shows what’s inside.

* It makes high level technology choices explicit
* It shows where there are relationships between containers and how they communicate

## Audience

Technical people inside and outside of the software development team, software developers, operational and support staff

## Container Diagram

Given below is the Container Diagram

