ABB Technikerschule

Technik. Informatik. Wirtschaft. Management →

Dominik Meyer
Software Engineering

Container Diagramm

Abweichung Zeitplan

Nach den ersten 9 Lektionen, Beschluss Dozent

- Speed etwas verringern
- Mehr auf Tasks eingehen
- Übungen im Unterricht vertiefen / verlängern

- *PlantUML* anstatt als Übung als «Idee» abgeben

- C4 Model Schritt für Schritt aneignen

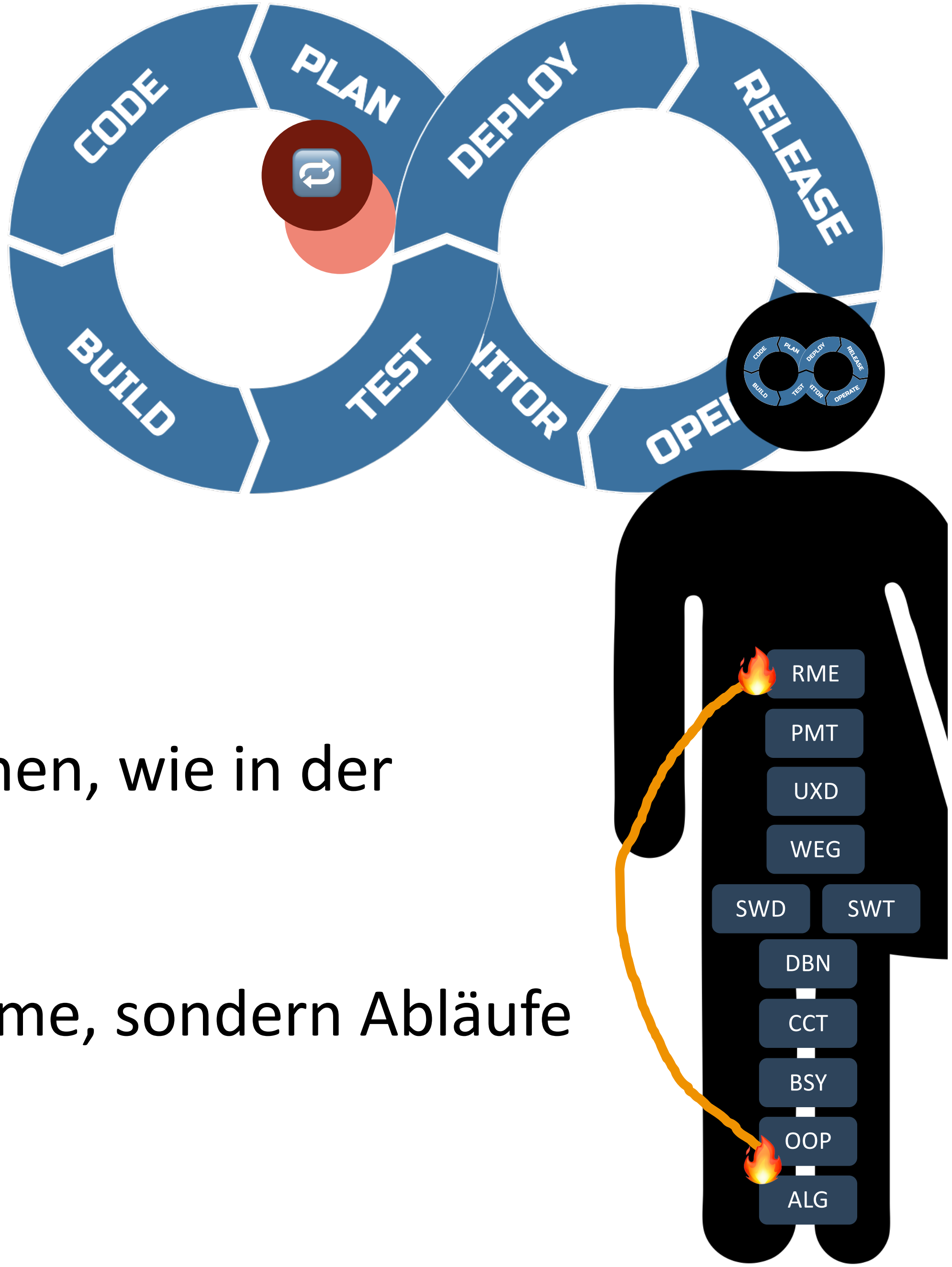
Ziel

Nach der Lektion haben die Studierenden aus dem Kontextdiagramm ein Containerdiagramm abgeleitet.

Ressourcen aktivieren

C4 Model - Container

- Lektion 10
  - Kontextdiagram
- Anforderungen verstehen und in Container aufteilen
  - Zoom in 🔍
- Wir arbeiten hier mit RME zusammen, wie in der Praxis!
- Ziel heute ist nicht fertige Diagramme, sondern Abläufe



Agenda

- C4 Model Intro Artikel
- Vom **Kontext** zum **Container**
- Vorgemachtes Beispiel
- «Bite» Jinder Container Diagramm
- Zielkontrolle

Task

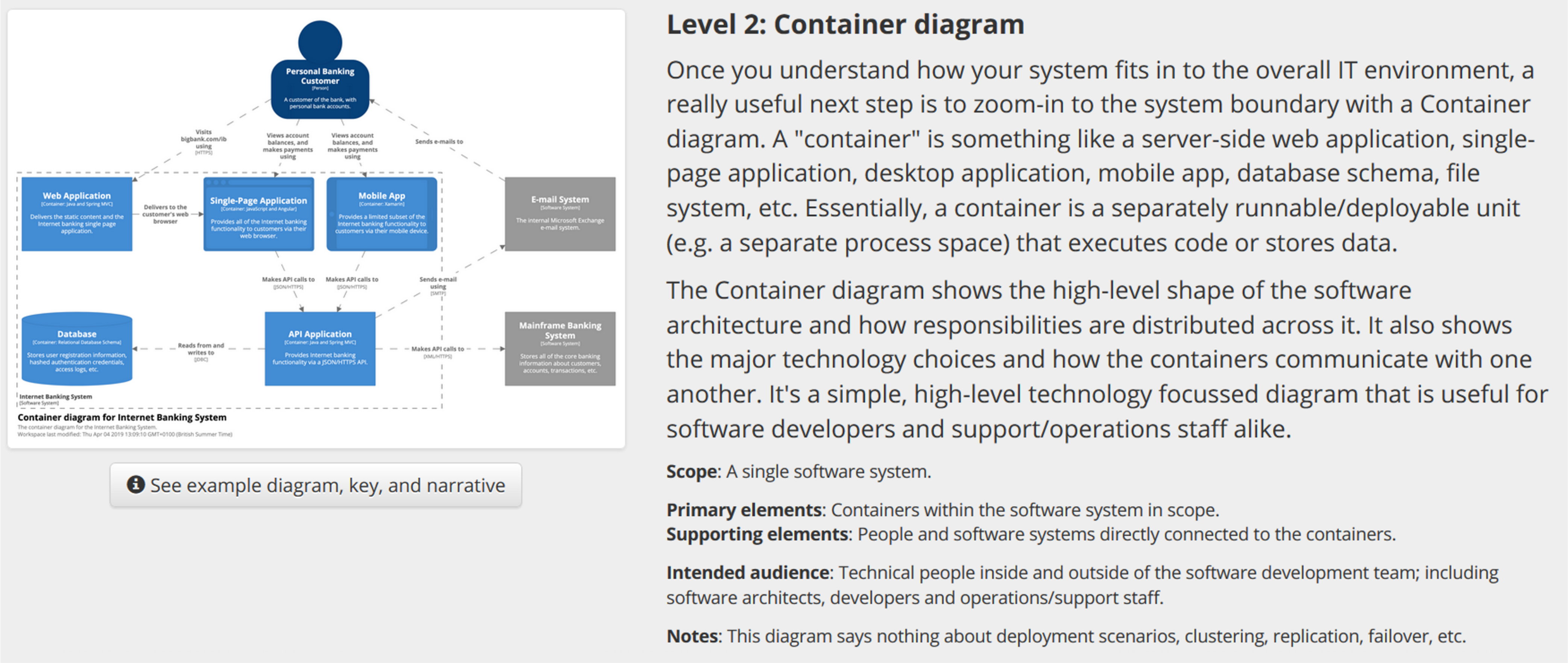
Lesen

AUFTRAG

⏰ 10min

infoq.com/articles/C4-architecture-model/

Lesen Sie diesen Artikel für sich.


Personal Banking Customer
[Person]
A customer of the bank, with personal bank accounts.
Visits bigbank.com/ib using
[HTTPS]
Views account balances, and makes payments using
Views account balances, and makes payments using
Sends e-mails to
Web Application
[Container: Java and Spring MVC]
Delivers the static content and the Internet banking single page application.
Delivers to the customer's web browser
Single-Page Application
[Container: JavaScript and Angular]
Provides all of the Internet banking functionality to customers via their web browser.
Mobile App
[Container: Xamarin]
Provides a limited subset of the Internet banking functionality to customers via their mobile device.
E-mail System
[Software System]
The internal Microsoft Exchange e-mail system.
Makes API calls to
[JSON/HTTPS]
Makes API calls to
[JSON/HTTPS]
Sends e-mail using
[SMTP]
Database
[Container: Relational Database Schema]
Stores user registration information, hashed authentication credentials, access logs, etc.
Reads from and writes to
[JDBC]
API Application
[Container: Java and Spring MVC]
Provides Internet banking functionality via a JSON/HTTPS API.
Makes API calls to
[XML/HTTPS]
Mainframe Banking System
[Software System]
Stores all of the core banking information about customers, accounts, transactions, etc.
Internet Banking System
[Software System]

Container diagram for Internet Banking System

The container diagram for the Internet Banking System.
Workspace last modified: Thu Apr 04 2019 13:09:10 GMT+0100 (British Summer Time)

See example diagram, key, and narrative

Level 2: Container diagram

Once you understand how your system fits in to the overall IT environment, a really useful next step is to zoom-in to the system boundary with a Container diagram. A "container" is something like a server-side web application, single-page application, desktop application, mobile app, database schema, file system, etc. Essentially, a container is a separately runnable/deployable unit (e.g. a separate process space) that executes code or stores data.

The Container diagram shows the high-level shape of the software architecture and how responsibilities are distributed across it. It also shows the major technology choices and how the containers communicate with one another. It's a simple, high-level technology focussed diagram that is useful for software developers and support/operations staff alike.

Scope: A single software system.

Primary elements: Containers within the software system in scope.
Supporting elements: People and software systems directly connected to the containers.

Intended audience: Technical people inside and outside of the software development team; including software architects, developers and operations/support staff.

Notes: This diagram says nothing about deployment scenarios, clustering, replication, failover, etc.

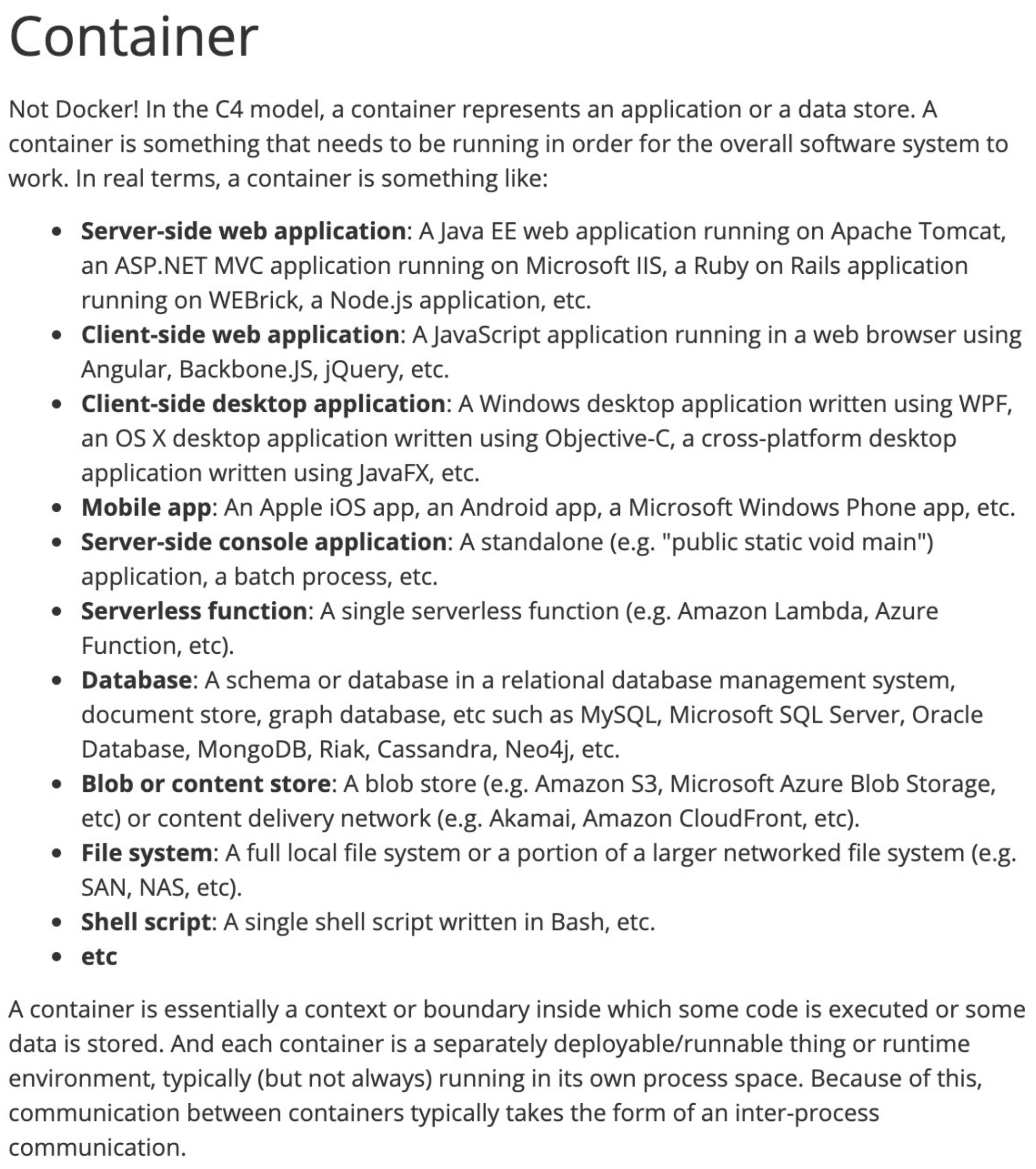
C4 model

Container aus Kontext

- Beschreibung lesen (hier in der Aufgabe)
  - Transfer, MA, DA in den Unterlagen aus RME
- Anforderungen und Use Cases analysieren
- Welche Container werden benötigt für diese «Funktionalität»?
- Was sagen die RME Spezifikationen zu den Schnittstellen?

Analogie:

- Kontext: SBB Online Fahrplan
  - Mögliche Container:
    - Mobile App
    - Fahrplan DB
    - Fahrplan System Deutsche Bahn, Italien, Österreich
    - Website
    - Schalter-Applikation usw.



Container

Not Docker! In the C4 model, a container represents an application or a data store. A container is something that needs to be running in order for the overall software system to work. In real terms, a container is something like:

- **Server-side web application**: A Java EE web application running on Apache Tomcat, an ASP.NET MVC application running on Microsoft IIS, a Ruby on Rails application running on WEBrick, a Node.js application, etc.
- **Client-side web application**: A JavaScript application running in a web browser using Angular, Backbone.JS, jQuery, etc.
- **Client-side desktop application**: A Windows desktop application written using WPF, an OS X desktop application written using Objective-C, a cross-platform desktop application written using JavaFX, etc.
- **Mobile app**: An Apple iOS app, an Android app, a Microsoft Windows Phone app, etc.
- **Server-side console application**: A standalone (e.g. "public static void main") application, a batch process, etc.
- **Serverless function**: A single serverless function (e.g. Amazon Lambda, Azure Function, etc).
- **Database**: A schema or database in a relational database management system, document store, graph database, etc such as MySQL, Microsoft SQL Server, Oracle Database, MongoDB, Riak, Cassandra, Neo4j, etc.
- **Blob or content store**: A blob store (e.g. Amazon S3, Microsoft Azure Blob Storage, etc) or content delivery network (e.g. Akamai, Amazon CloudFront, etc).
- **File system**: A full local file system or a portion of a larger networked file system (e.g. SAN, NAS, etc).
- **Shell script**: A single shell script written in Bash, etc.
- **etc**

A container is essentially a context or boundary inside which some code is executed or some data is stored. And each container is a separately deployable/runnable thing or runtime environment, typically (but not always) running in its own process space. Because of this, communication between containers typically takes the form of an inter-process communication.

Task

Herunterbrechen

AUFTRAG

⏰ 15min

/docs/tasks/bites/c4-model - container-diagram

«Stormen» Sie ein Containerdiagram

Die Aufgabenstellung ist auf den Docs

Das Tool ist frei, schliessen Sie am Kontextdiagramm an

Zielkontrolle

Wer ist die Zielgruppe des Container Diagramm?

Was ist der Unterschied zum Context Diagramm?

Welches Tool sollte für das Container Diagramm verwendet werden?

WEITER WISSEN.

Wir begleiten Sie!