Diagrams

Component Diagram

Component Diagram

- Komponentendiagramm
- Stellt die Beziehungen zwischen den einzelnen Systemkomponenten dar. Kann physische und logische Komponenten beinhalten.
- **Zeigt...**
  - alle verwendeten Systeme
    - Web-, backend-server
    - Datenbanken
    - Browser
    - REST-clients
  - Schnittstellen

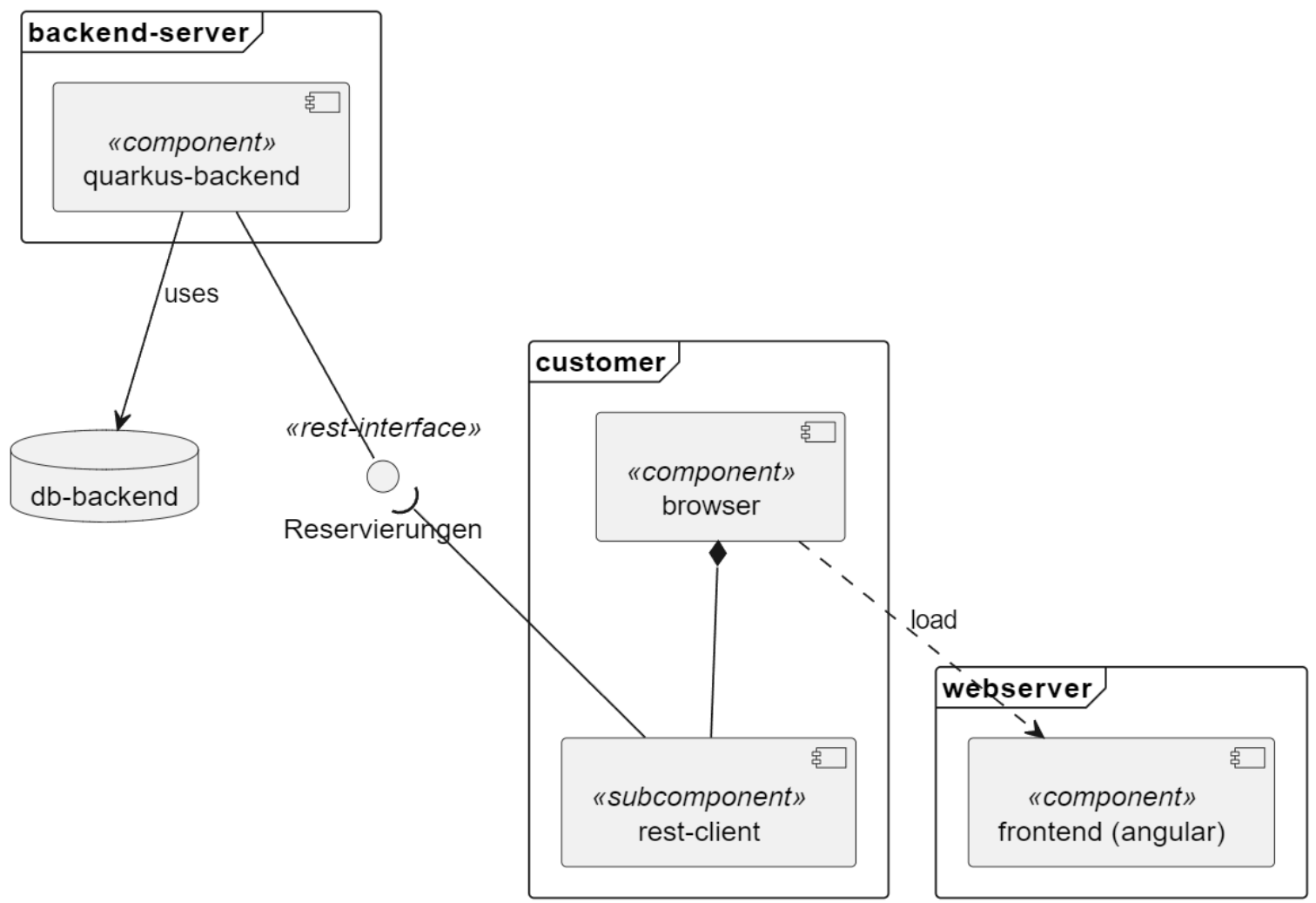
Component Diagram

- **Zeigt NICHT (unbedingt)...**
  - Den Standort
  - Die Anzahl
  - Das Datenmodell
  - ...

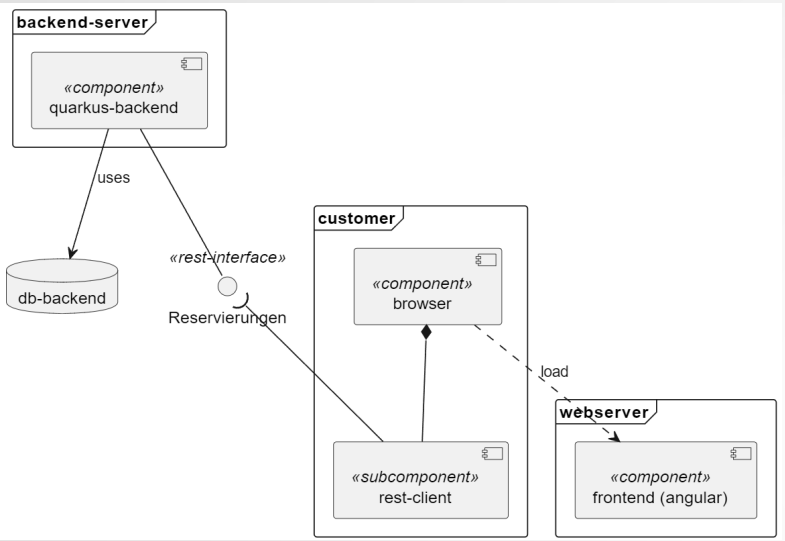
Component Diagram

Example

Reservierungssystem eines Tennisvereins



Reservierungssystem eines Tennisvereins




```
@startuml

interface Reservierungen <<rest-interface>>

frame backend-server {
    component "quarkus-backend" <<component>> as
        backend
}

database "db-backend" as db {
}

backend --> db : uses

backend -- Reservierungen

frame webserver {
    component "frontend (angular)" <<component>> as
        frontend
}

frame customer {
    component browser <<component>>
    component [rest-client] <<subcomponent>>
}

Reservierungen )-- [rest-client]
browser ..> frontend : load
browser *-- [rest-client]
@enduml
```

Component Diagram

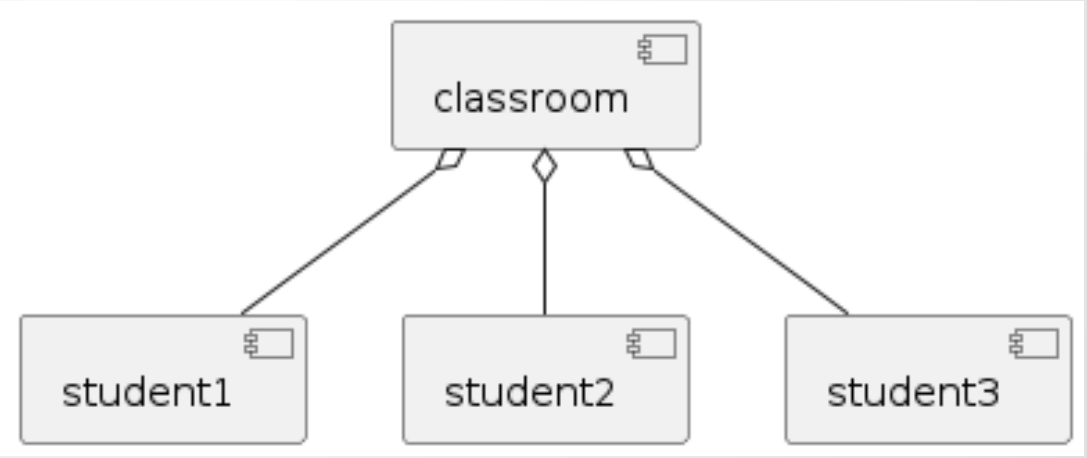
Composition
Aggregation

Aggregation

"has a" relationship.

Destroying the container does not affect these elements.

Elements may live without a container.

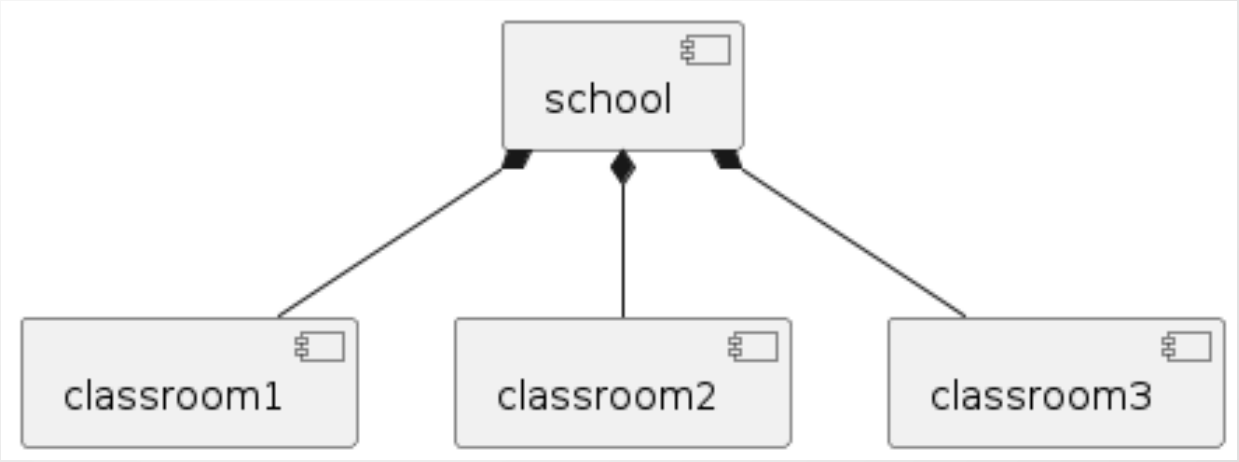


Composition

"owns" relationsship (a special, tighter form of aggregation).

Destroying the container affects these elements.

Elements cannot live without a container.



Data Model (ERD)

(REST-interface)

Data Model (ERD)

- Datenmodell
- (**E**ntity **R**elationship **D**iagram)
- Stellt die Daten und deren Beziehungen dar (Entity-Relationship).
- **Zeigt...**
  - Daten-Entitäten
    - Tabellen
    - REST-Ressourcen
  - Deren Beziehungen
    - 1 zu 1, 1 zu n, m zu n
    - 0 || 1, genau 1, null || n, 1 || n
  - Wie sie verwendet werden
    - uses, contains, appears in, ...

Data Model

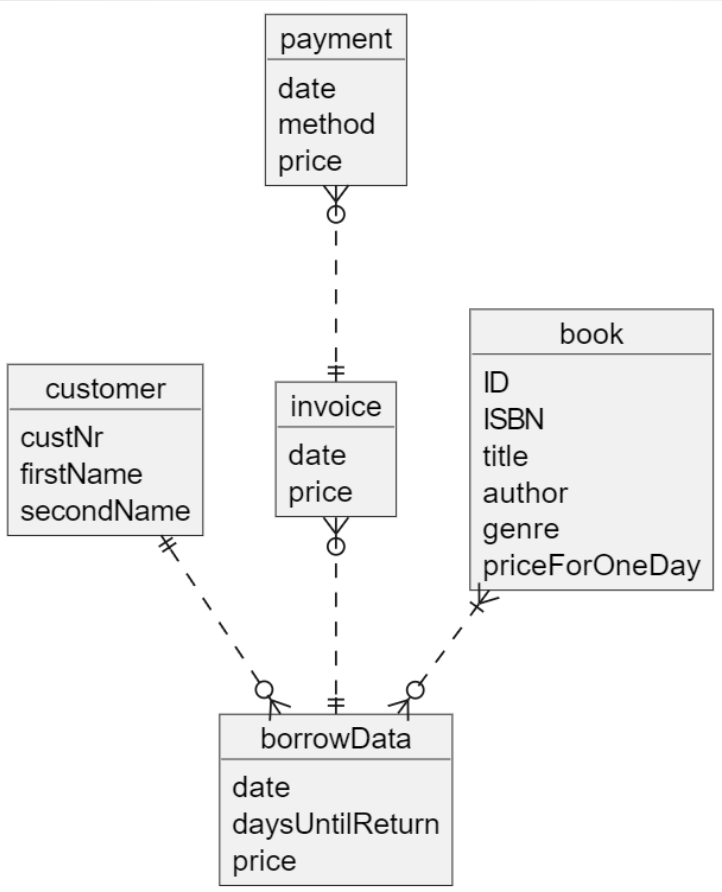
- **Zeigt NICHT...**
  - Die Art der Datenbank
  - Das verwendete Clustering
  - Die Server
  - Den Standort der Applikation
  - ...

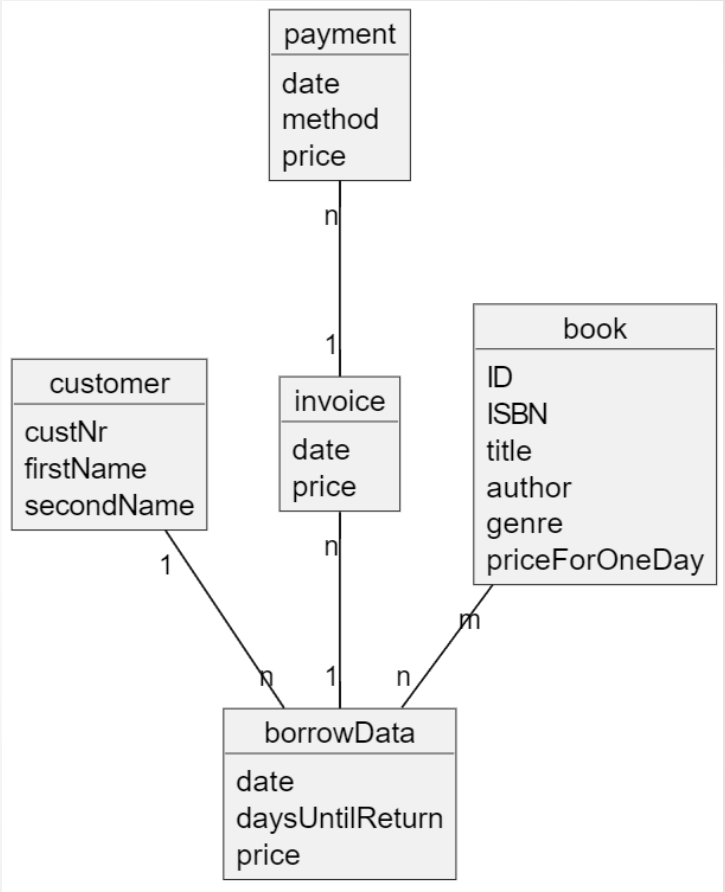
Data Model (ERD)

(REST-interface)

Example

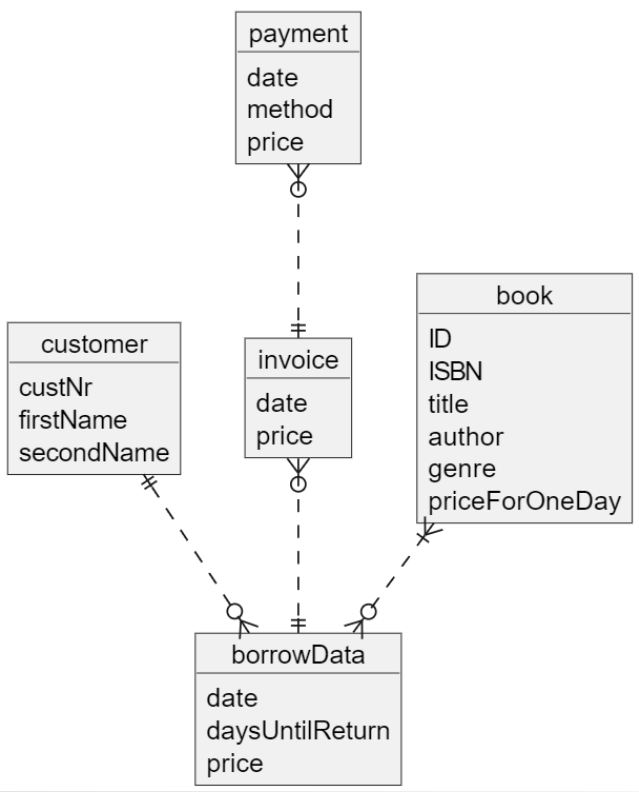
Buchhandlungs - App





- simplified
- loses some context (like zero-or-many to "n")

Buchhandlungs - App




```
@startuml

object customer {
    custNr
    firstName
    secondName
}

object borrowData {
    date
    daysUntilReturn
    price
}

object invoice {
    date
    price
}

object payment {
    date
    method
    price
}
```

```
book }|..o{ borrowData
customer ||..o{ borrowData
invoice }o..|| borrowData
payment }o..|| invoice

object book {
    ID
    ISBN
    title
    author
    genre
    priceForOneDay
}

@enduml
```

Deployment Diagram

Deployment Diagram

- Stellt die verwendete Hardware (Nodes), die installierte Software (Fragments) und deren Interfaces dar.
- **Zeigt...**
  - **WIE** es installiert wird.
  - Nodes
    - Server, Cluster, Cloud, ...
  - Fragments
    - Libraries, Apps, Services, ...
  - Deren Schnittstellen
    - REST, ...

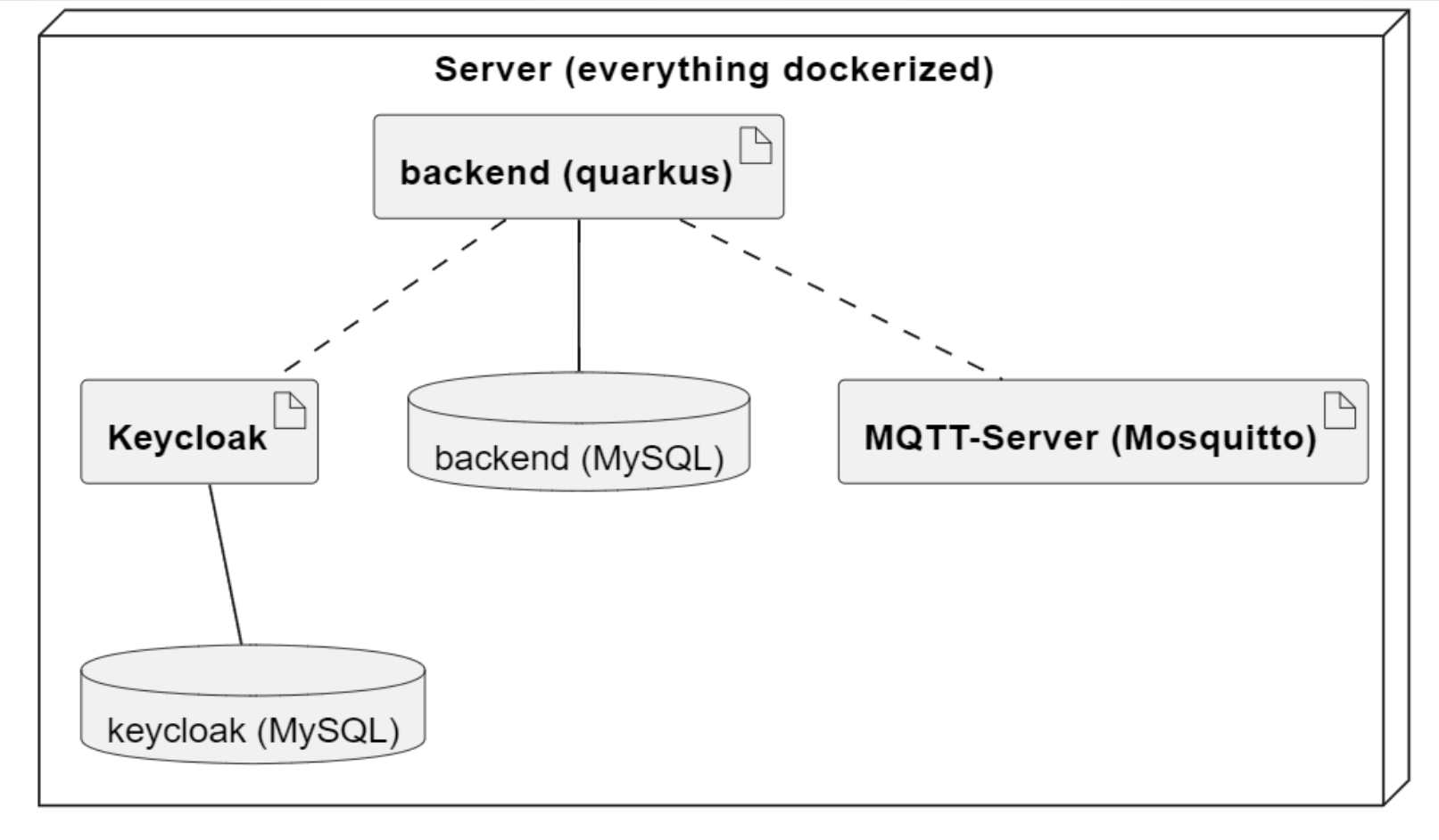
Deployment Diagram

- **Zeigt NICHT…**
  - Das Datenmodell
  - Die Beschaffenheit der Schnittstellen
  - Klienten
  - …

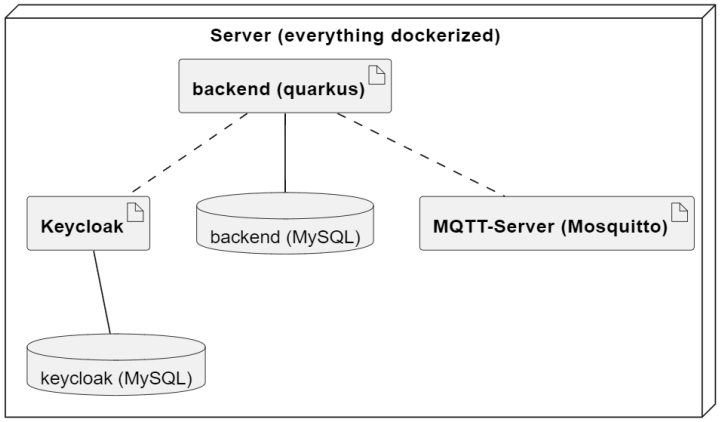
Deployment Diagram

Example

Deployment Diagram



Deployment Diagram




```
@startuml

node server as "Server (everything dockerized)" {
    database dbkeycloak as "keycloak (MySQL)" {
    }
    artifact keycloak as "<b>Keycloak</b>" {
    }

    database dbbackend as "backend (MySQL)" {
    }

    artifact backend as "<b>backend (quarkus)</b>" {
    }

    artifact mqtt as "<b>MQTT-Server (Mosquitto)</b>" {
    }
}

keycloak -- dbkeycloak
backend -- dbbackend
backend .. mqtt
backend .. keycloak

@enduml
```

References

- https://en.wikipedia.org/wiki/Deployment_diagram
- https://en.wikipedia.org/wiki/Component_diagram
- https://en.wikipedia.org/wiki/Data_model