Question Answering and Chatbots

4th Practical exercise – a simple KGQA-system

Aleksandr Perevalov

aleksandr.perevalov@hs-anhalt.de

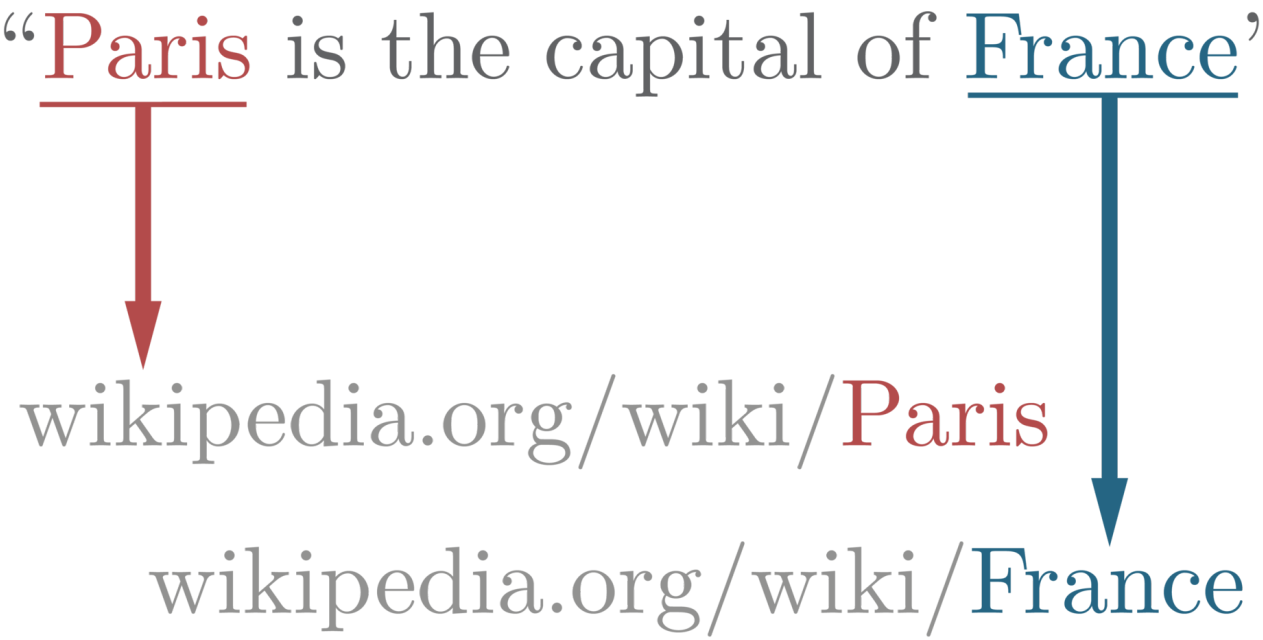
November 3, 2021

Hochschule Anhalt

Anhalt University of Applied Sciences

Named Entity Recognition and Linking

NEL – is the task of determining unique identity to entities (people, locations, songs, etc.) mentioned in text.



Named Entity Recognition and Linking

NEL consists of several steps:

Named Entity Recognition and Linking

NEL consists of several steps:

1. Named Entity Recognition (NER) – spot text spans in that contain Named Entity label;

Named Entity Recognition and Linking

NEL consists of several steps:

1. Named Entity Recognition (NER) – spot text spans in that contain Named Entity label;
2. Named entity search – given the text form, find possible candidates in database;

Named Entity Recognition and Linking

NEL consists of several steps:

1. Named Entity Recognition (NER) – spot text spans in that contain Named Entity label;
2. Named entity search – given the text form, find possible candidates in database;
3. Candidates ranking – order candidates according to a parameter e.g. PageRank, Views per month.

Named Entity Recognition and Linking

NEL consists of several steps:

1. Named Entity Recognition (NER) – spot text spans in that contain Named Entity label;
2. Named entity search – given the text form, find possible candidates in database;
3. Candidates ranking – order candidates according to a parameter e.g. PageRank, Views per month.

Tools NER: spaCy, StanfordNLP etc.

Named Entity Recognition and Linking

NEL consists of several steps:

1. Named Entity Recognition (NER) – spot text spans in that contain Named Entity label;
2. Named entity search – given the text form, find possible candidates in database;
3. Candidates ranking – order candidates according to a parameter e.g. PageRank, Views per month.

Tools NER: spaCy, StanfordNLP etc.
Tools NEL: DBpedia Spotlight, TagMe, AGDISTIS, etc.

Relation Prediction

Text classification has many applications in Natural Language Processing. Specifically, in Question Answering & Chatbots, it can be used as a **Relation (Predicate) Prediction** component.

Relation Prediction

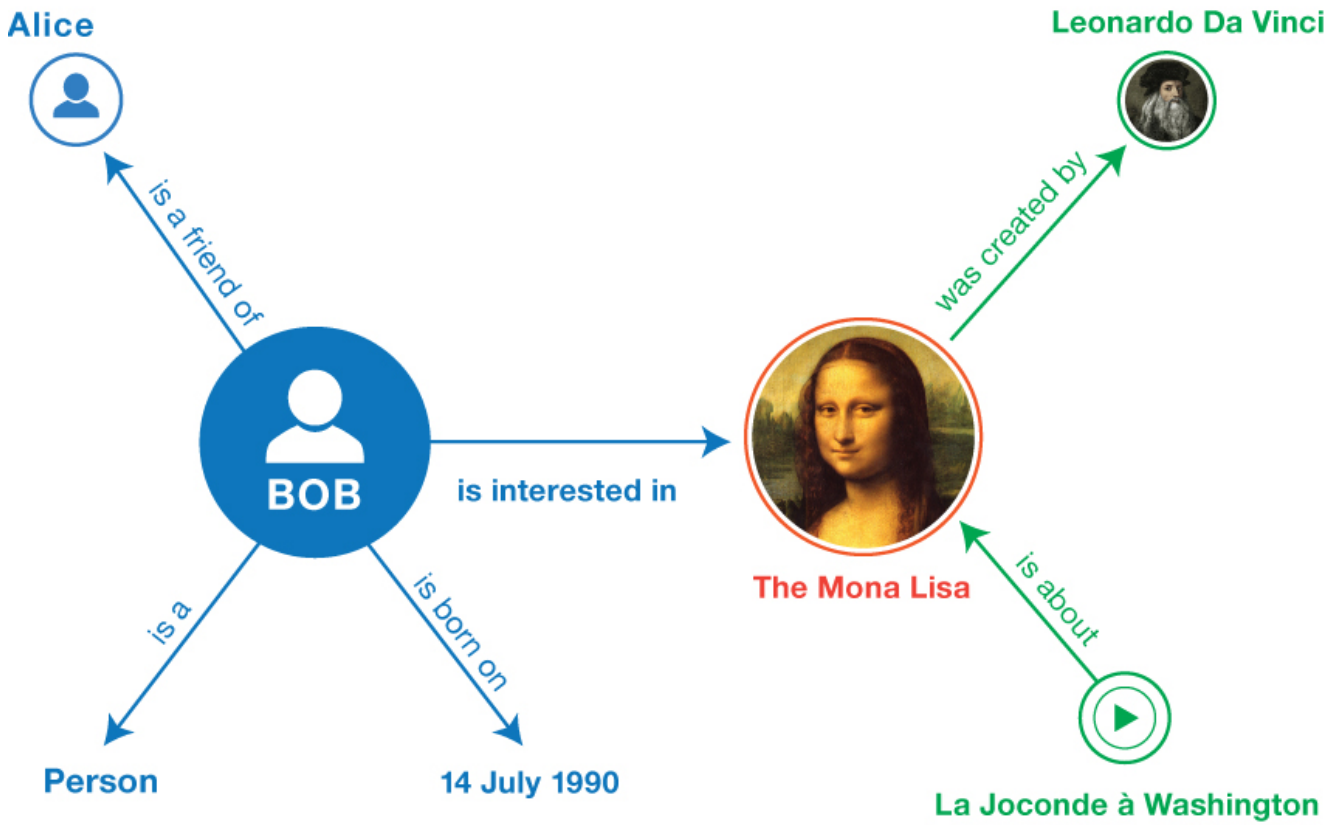
Relation (or Predicate) in terms of knowledge graphs is an **edge that is connecting two nodes (or entities)**. For example, having a triple: <Mona_Lisa> <?> <Leonardo_da_Vinci> the relation <?> is <Author> (or e.g. <Was_Created_By>).

Relation Prediction

In this regard, Relation Prediction is the task of **recognizing a relation, based on a textual question**. In this case, question: "Who is the author of Mona Lisa?" has relation "Author" (or e.g. "Was created by").

Relation Prediction

In this regard, Relation Prediction is the task of **recognizing a relation, based on a textual question**. In this case, question: "Who is the author of Mona Lisa?" has relation "Author" (or e.g. "Was created by").



Simple SPARQL queries

Subject-Predicate-**Object** – **Forward Query:**

```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>

SELECT ?uri WHERE {
  dbr:Michael_Jordan dbo:birthPlace ?uri .
  # we ask for an object (?uri)
}
```

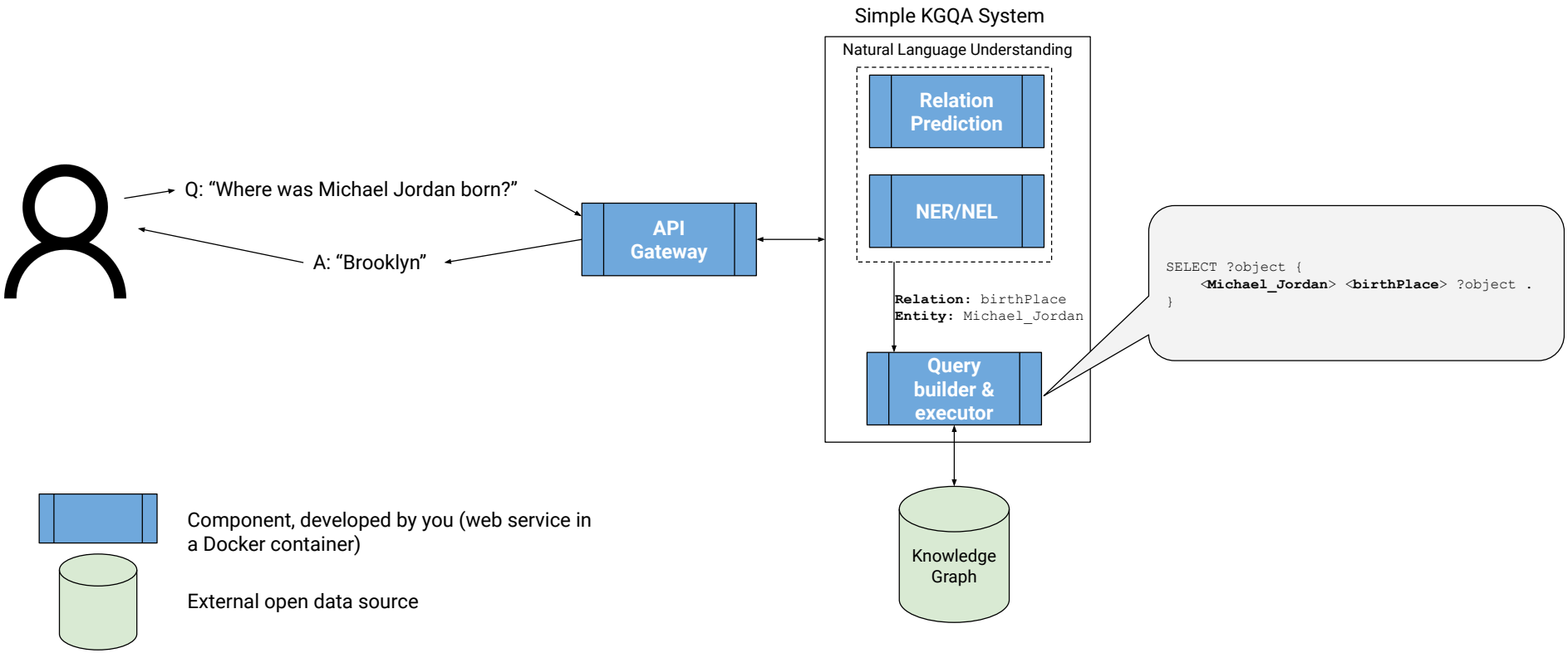
Simple SPARQL queries

Subject-Predicate-Object – **Backward Query:**

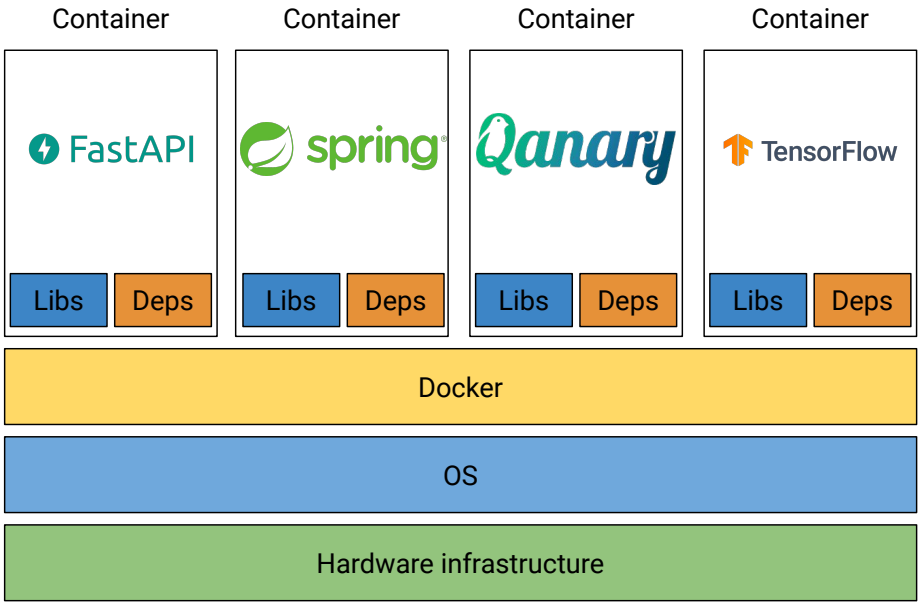
```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>

SELECT ?uri WHERE {
  ?uri dbo:birthPlace dbr:Berlin .
  # we ask for the subject (?uri)
}
```

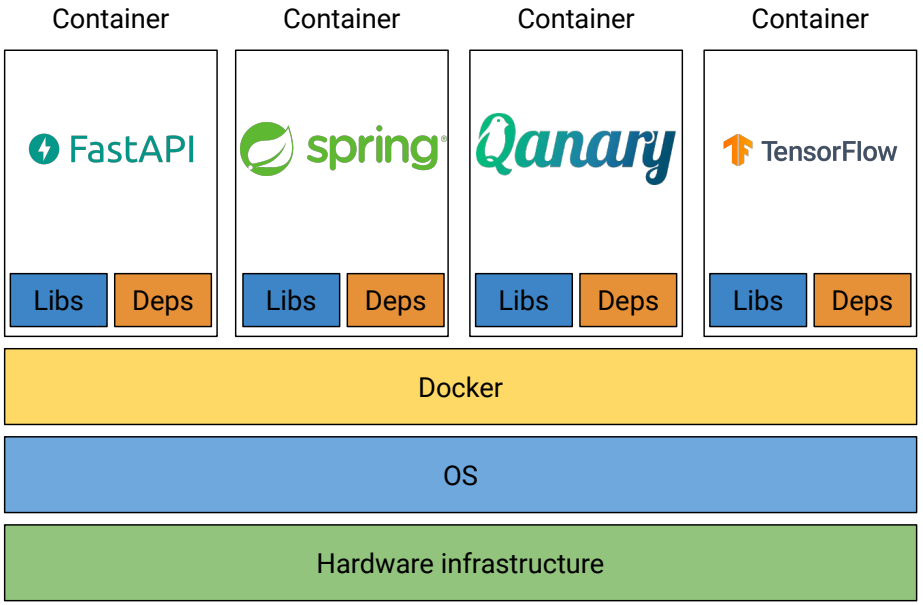
Simple KGQA system



Docker

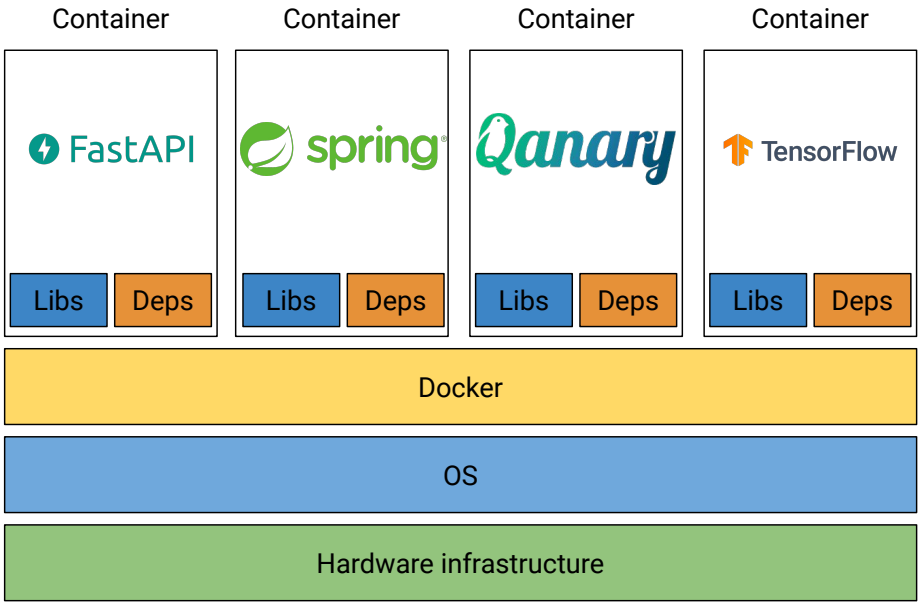

Container
Container
Container
Container
FastAPI
spring
Qanary
TensorFlow
Libs
Deps
Libs
Deps
Libs
Deps
Libs
Deps
Docker
OS
Hardware infrastructure

Docker



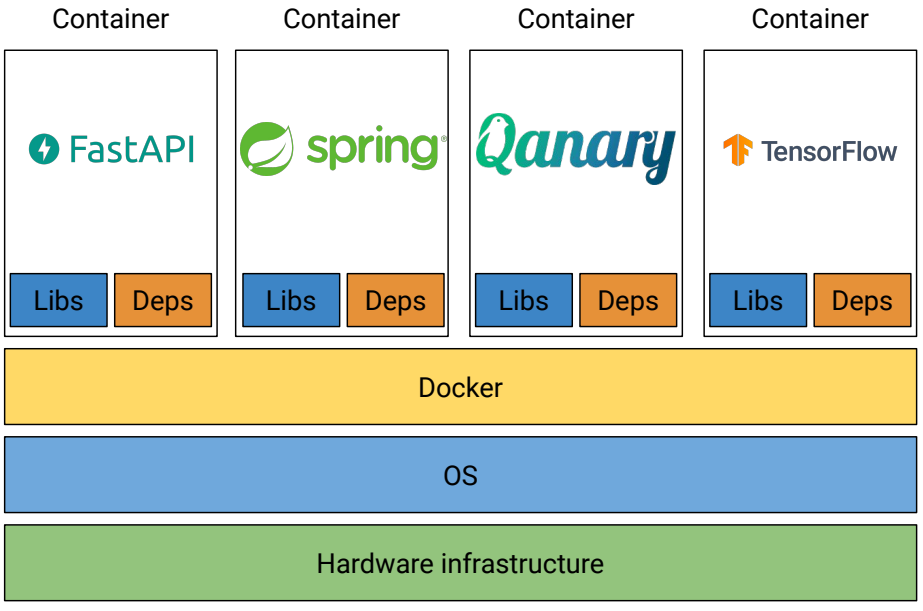
- A tool to pack your application into “containers”;

Docker



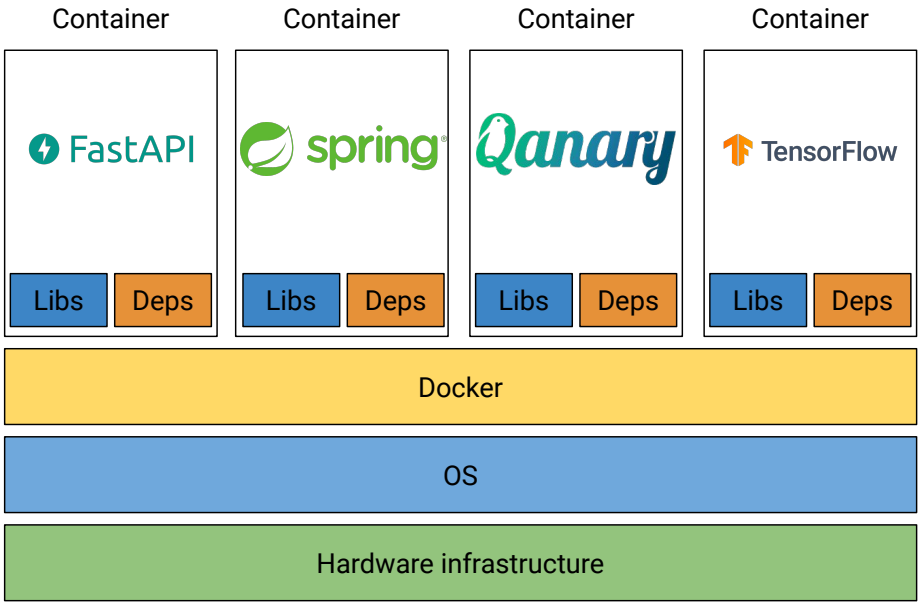
- A tool to pack your application into “containers”;
- Container is a run-time environment with libraries/dependencies/OS;

Docker



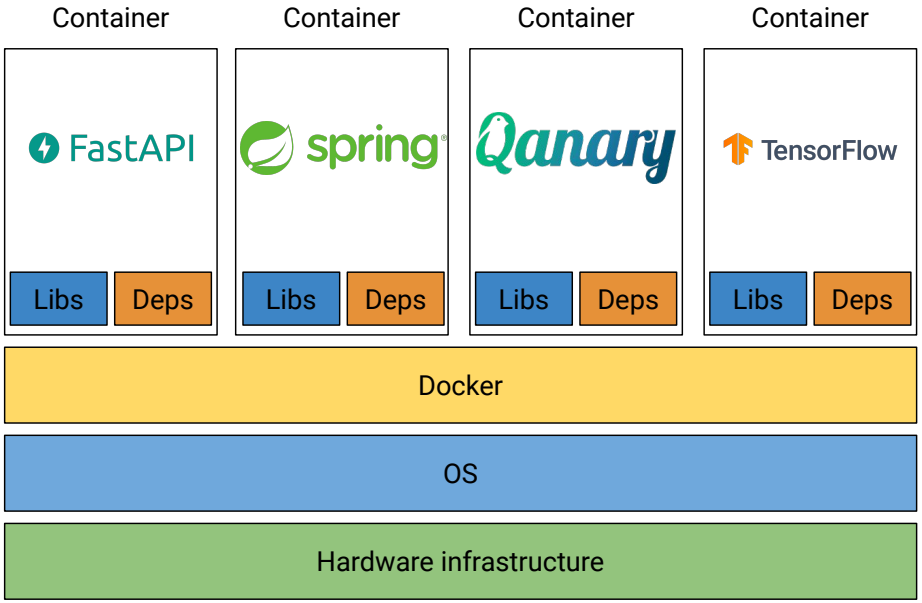
- A tool to pack your application into “containers”;
- Container is a run-time environment with libraries/dependencies/OS;
- Image is an immutable file with libraries/dependencies/OS;

Docker



- A tool to pack your application into “containers”;
- Container is a run-time environment with libraries/dependencies/OS;
- Image is an immutable file with libraries/dependencies/OS;
- Image is created with a script – “Dockerfile” – over another image;

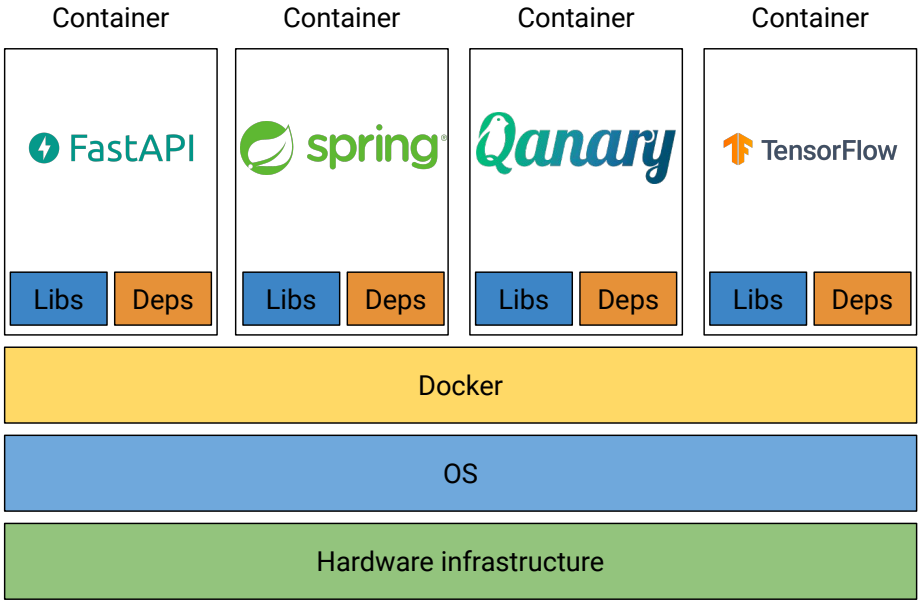
Docker



- Linux containers are running on Linux, and Windows containers are running on Windows[1];

[1] you can run Linux container on Windows, but then Linux VM is created

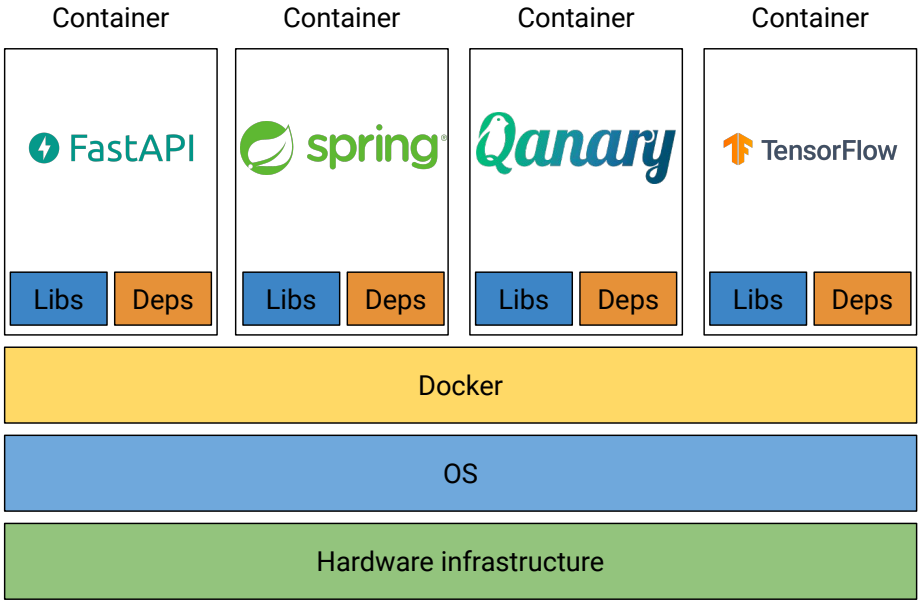
Docker



- Linux containers are running on Linux, and Windows containers are running on Windows[1];
- Same image can be reused for running multiple containers;

[1] you can run Linux container on Windows, but then Linux VM is created

Docker



- Linux containers are running on Linux, and Windows containers are running on Windows[1];
- Same image can be reused for running multiple containers;
- Docker Compose will help you to run multiple-container apps.

[1] you can run Linux container on Windows, but then Linux VM is created

Task for Today – Let's Synchronize!

Task for Today – Let's Synchronize!

- Exercise leaders – join those who have not finished 1st, 2nd and 3rd;

Task for Today – Let's Synchronize!

- Exercise leaders – join those who have not finished 1st, 2nd and 3rd;
- One mini-group works on one exercise – start from the beginning;

Task for Today – Let's Synchronize!

- Exercise leaders – join those who have not finished 1st, 2nd and 3rd;
- One mini-group works on one exercise – start from the beginning;
- Exercise leaders – share your experience and solutions;

Task for Today – Let's Synchronize!

- Exercise leaders – join those who have not finished 1st, 2nd and 3rd;
- One mini-group works on one exercise – start from the beginning;
- Exercise leaders – share your experience and solutions;
- Others – formulate concrete problems and questions;

Task for Today – Let's Synchronize!

- Exercise leaders – join those who have not finished 1st, 2nd and 3rd;
- One mini-group works on one exercise – start from the beginning;
- Exercise leaders – share your experience and solutions;
- Others – formulate concrete problems and questions;
- I will work with each group one by one.

0 Introduction;
1 NER & NEL;
2 Question classification & Web service/API;
3 SPARQL queries over Knowledge Graphs;
4 **Simple KGQA system – based on exercises 0, 1, 2, 3**;
5 Qanary Framework – component oriented approach;
6 Simple ODQA system?;
7 Evaluation of QA systems.