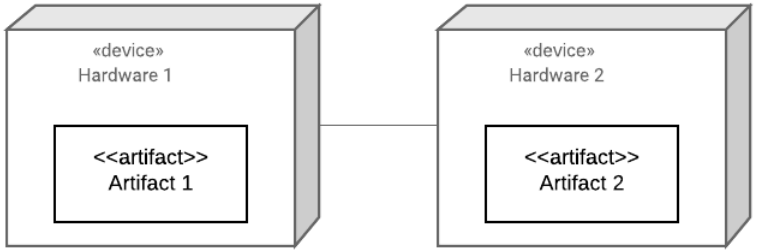
Mu.SA
museum sector alliance

Deployment Diagrams

Co-funded by the
Erasmus+ Programme
of the European Union

Deployment Diagrams

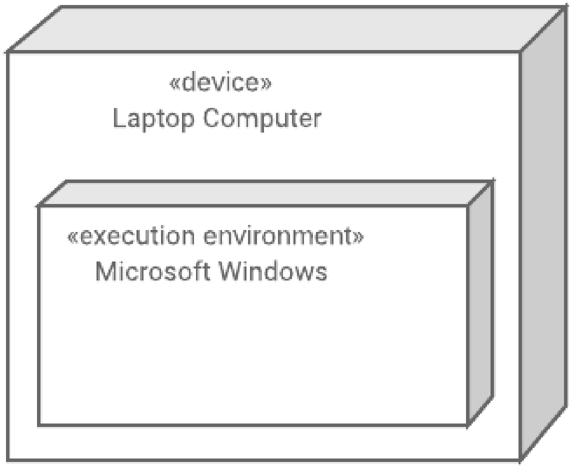
- There are many informal graphical notations to represent infrastructure and deployment;
- UML Deployment Diagrams are a ***standard*** for expressing **infrastructure** and **execution environments** in which software systems run;
- They also allow to represent how **software artifacts** are associated with the hardware infrastructure (nodes) where they should be installed;
- The figure below exemplifies a simple UML deployment diagram;



Co-funded by the Erasmus+ Programme of the European Union

Deployment diagrams

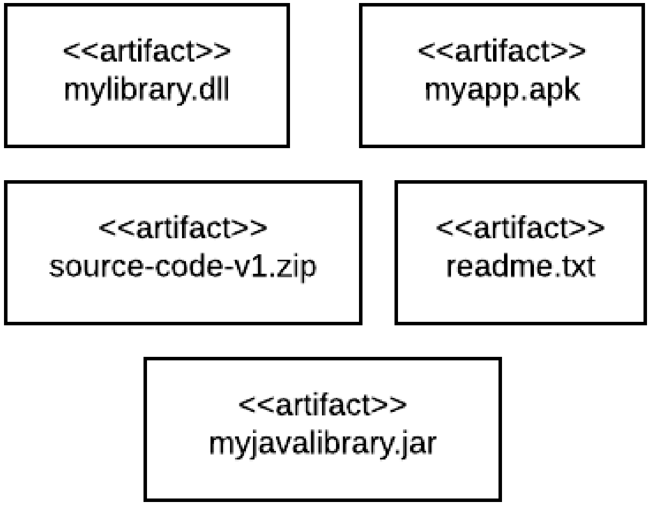
- ***Hardware nodes*** and ***execution environments*** are represented as boxes, as exemplified below.



Co-funded by the Erasmus+ Programme of the European Union

Deployment diagrams

- ***Software artifacts*** are all the files that make the software system and that play a part in the executon of the system.



<<artifact>>
mylibrary.dll

<<artifact>>
myapp.apk

<<artifact>>
source-code-v1.zip

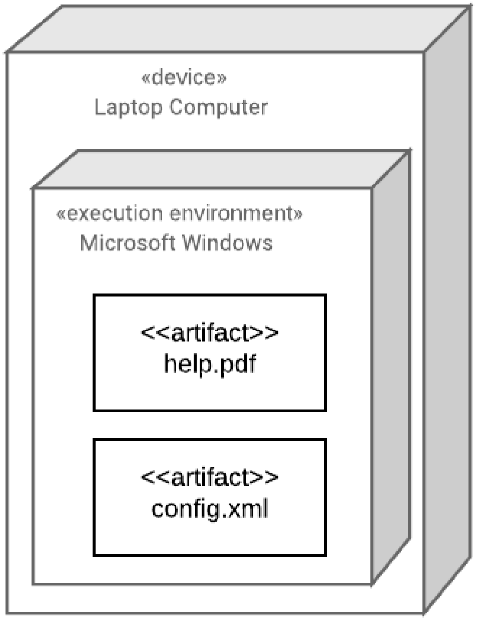
<<artifact>>
readme.txt

<<artifact>>
myjavalibrary.jar

...

Deployment diagrams

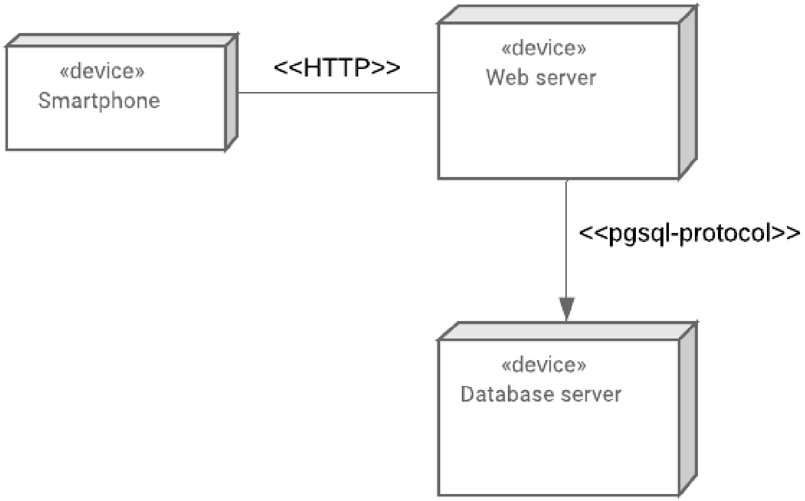
- Software **artifacts** can be associated to hardware **nodes**, as exemplified below.



Co-funded by the Erasmus+ Programme of the European Union

Deployment diagrams

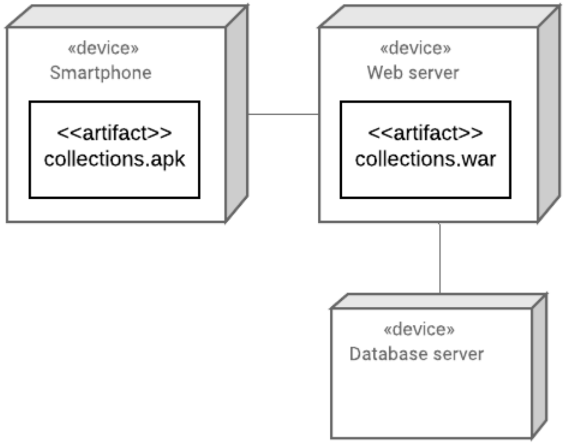
- **Lines** represent the communication between the various nodes during execution.
- The **communication protocol** (i.e., the language used to communicate) is represented using the notation for UML stereotypes, between << and >>.



Co-funded by the Erasmus+ Programme of the European Union

Example #1

- The example below illustrates the infrastructure required for the deployment of a software system for a museum;
- The infrastructure can be described as composed by three different hardware elements: the server named *Web Server*, the sever named *Database Server* and *Smartphones* of the end-users;



(continues)

Co-funded by the
Erasmus+ Programme
of the European Union

Example #1

- The installation of the system in this infrastructure comprises two files:
  - `collections.apk` is a file that is installed on smartphones and that provides an app to allow accessing a museum's online catalog;
  - `collections.war` is a file to be installed on the server named *Web Server* and that will provide data to the smartphone app;

Co-funded by the Erasmus+ Programme of the European Union

Example #2

Consider a software system to monitor the temperature in the exhibition room of a museum. Temperature data is obtained through a sensor attached to a small computer (RaspberryPi) where it is installed a service responsible for reading data from the sensor and sending it to a central server. This service consists of a single file *temperatureCollector.jar*.

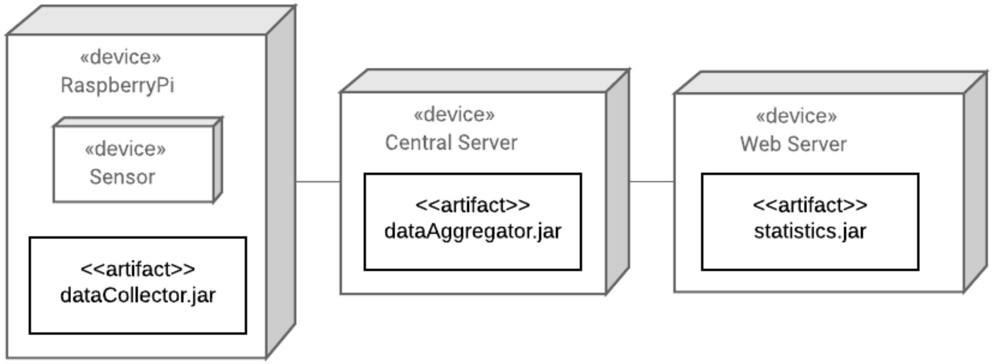
In the central server, data is aggregated and stored in a database. This task is performed by a piece of software packaged in the file *dataAggregator.jar*. The aggregated data is used by another service that computes some temperature statistics. This is a Web application and is installed in a dedicated Web server as the file *statistics.jar*.

How would you design the deployment diagram for such a software system?

Co-funded by the
Erasmus+ Programme
of the European Union

Example #2

- The figure below illustrates a possible representation of the infrastructure required by the temperature monitoring system and the respective deployment;



Co-funded by the
Erasmus+ Programme
of the European Union

Synopsis

- This presentation was about the main concepts about software deployment, including its connection with DevOps principles and practices, the associated packaging and distribution methods, the relation between deployment and software architecture, and the activities and technologies used for deploying.
- Now that you finished watching it, you should be able to:
  - Identify software packaging and distribution methods;
  - Explain the relationship between deployment and software architecture;
  - Identify technologies and standards used for deploying software;
  - Design the deployment environment of a given system

Co-funded by the
Erasmus+ Programme
of the European Union

List of references

- Kim, Gene, et al. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. IT Revolution, 2016.
- Newman, Sam. Building microservices: designing fine-grained systems. " O'Reilly Media, Inc.", 2015.
- Miles, Russ, and Kim Hamilton. Learning UML 2.0. " O'Reilly Media, Inc.", 2006.

Co-funded by the Erasmus+ Programme of the European Union

Presenter's bio page

Filipe Correia
filipe.correia@fe.up.pt

I'm an Assistant Professor at the University of Porto, where I teach and research on Software Engineering topics such as Development Processes, Software Design and Architecture, Object-Oriented Programming and Cloud Computing.

Co-funded by the Erasmus+ Programme of the European Union

Mu.SA
museum sector alliance

Thank you for your attention!

Credits

Author(s): Filipe Correia

Technical reviewers: Christos Pierrakeas and Panagiota Polymeropoulou, HOU

Scientific reviewer: Panos Fitsilis, HOU

www.project-musa.eu

musa@daissy.eap.gr

@MuseumSectorAlliance

#MuseumSectorAlliance

Project Number: 575907-EEP-1-2016-1-EL-EPPKA2-SSA

Co-funded by the Erasmus+ Programme of the European Union

This project has been funded with support from the European Commission. This presentation reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.