



Lecture 12

ADVANCED TERMS AND TOPICS

PB007 Software Engineering I
Faculty of Informatics, Masaryk University
Fall 2020

Topics covered



- ✧ Summary of covered topics
- ✧ Outline of additional topics
- ✧ Languages and frameworks
- ✧ Tool support
- ✧ Course follow-up



Summary of Covered Topics

Lecture 12/Part 1

Covered topics



1. **Software development**, UML Use Case diagram.
2. **Requirements specification**, UML Activity diagram.
3. System analysis and design, structured vs. object-oriented A&D.
4. **Object oriented analysis**, UML Class, Object and State diagram.
5. **Data modelling and management**, ERD.
6. **High-level design**, UML Class diagram in design.
7. **Low-level design and implementation**, UML Interaction diagrams
8. **Architecture design**, UML Package, Component and Deployment diagram.
9. **Testing**, verification and validation.
10. **Operation**, maintenance and system evolution.



Software development **management**.



Outline of Additional Topics

Lecture 12/Part 2

Distributed systems



✧ Virtually all large computer-based systems are now distributed systems.

“... a collection of independent computers that appears to the user as a single coherent system.”

✧ Distributed systems issues

- Distributed systems are **more complex** than systems that run on a single processor.
- Complexity arises because different parts of the system are **independently managed** as is the network.
- There is **no single authority** in charge of the system so top-down control is impossible.

Mobile applications



- ✧ A mobile applications include apps designed to run on smartphones, tablet computers and other mobile devices.
- ✧ They are usually available through application distribution platforms, operated by the owner of the mobile operating system, such as the Apple App Store and Google Play.
- ✧ Mobile apps were originally offered for general productivity and information retrieval, including email, calendar, contacts and weather information.
- ✧ However, public demand drove rapid expansion into many other categories, including banking, order-tracking, or medical apps.

Embedded systems



- ✧ Computers are used to control a wide range of systems from **simple domestic machines**, through **games controllers**, to entire **manufacturing plants**.
- ✧ Their software must react to events generated by the hardware and, often, issue control signals in response to these events.
- ✧ The software in these systems is **embedded in system hardware**, often in **read-only memory**, and usually responds, in **real time**, to events from the system's environment.
- ✧ Issues of **safety** and **reliability** may dominate the system design.

Cloud computing

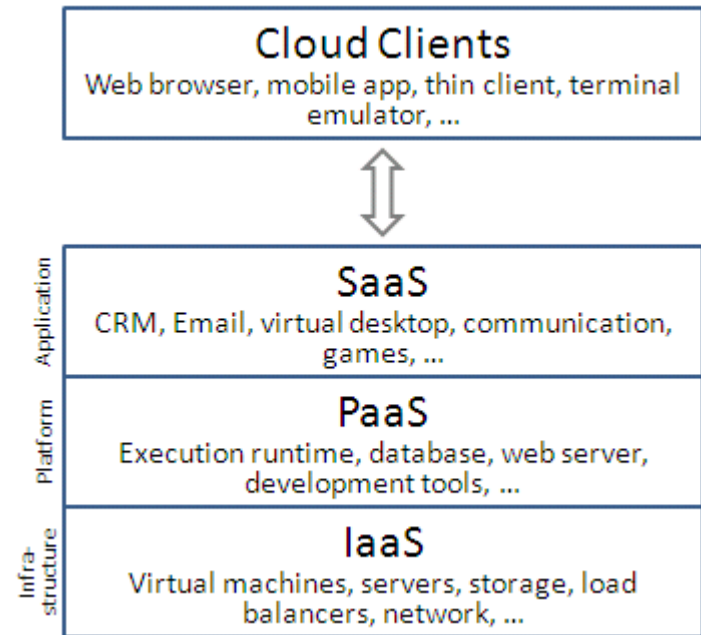


✧ Cloud computing is computing in which large groups of remote servers are networked to allow **centralized data storage** and online **access to computer services or resources**.

✧ Service models

- Infrastructure as a service (IaaS)
- Platform as a service (PaaS)
- Software as a service (SaaS)

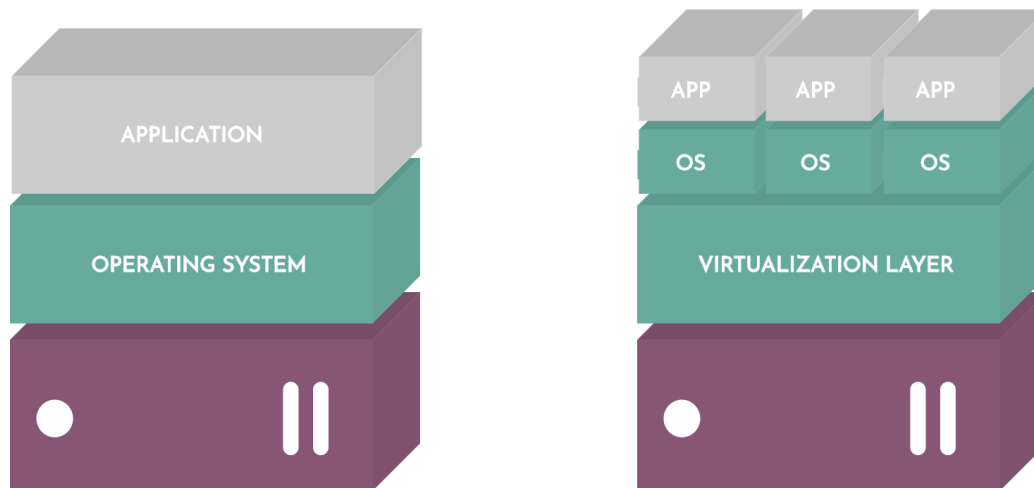
✧ Moreover, **big data and its processing** is a topic on its own



Virtualization



- ✧ A process of running a virtual instance of a computer system in a layer abstracted from the actual hardware
- ✧ A virtual machine (VM) is an isolated software container with an OS and application inside
- ✧ Each VM is completely independent



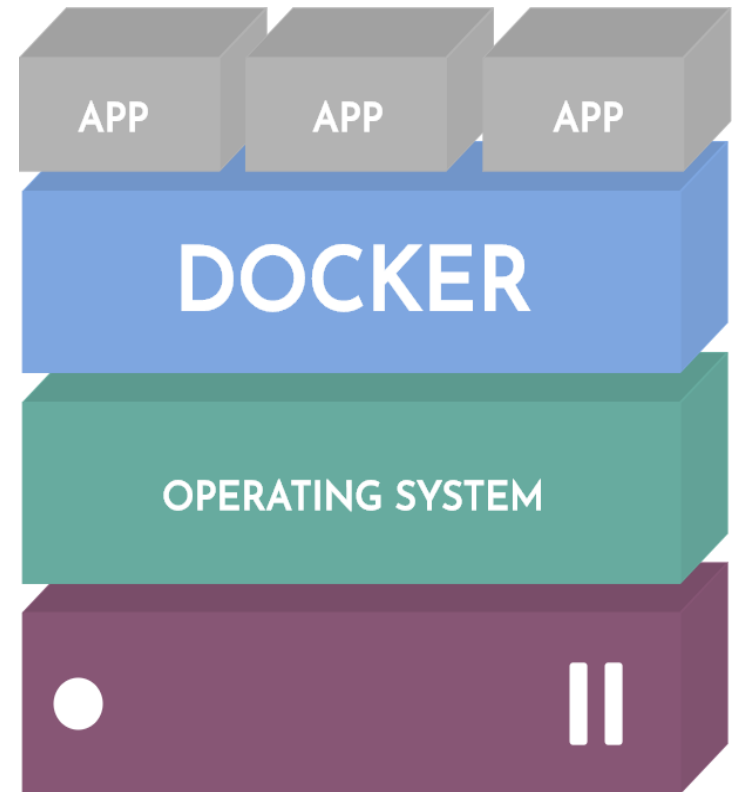
TRADITIONAL ARCHITECTURE

VIRTUAL ARCHITECTURE

Docker



- ✧ An open source platform for building, deploying, and managing containerized applications
- ✧ A docker container is a standalone executable package of software that includes everything needed to run an application
- ✧ Containers share one OS unlike VMs





Languages and Frameworks

Lecture 12/Part 3

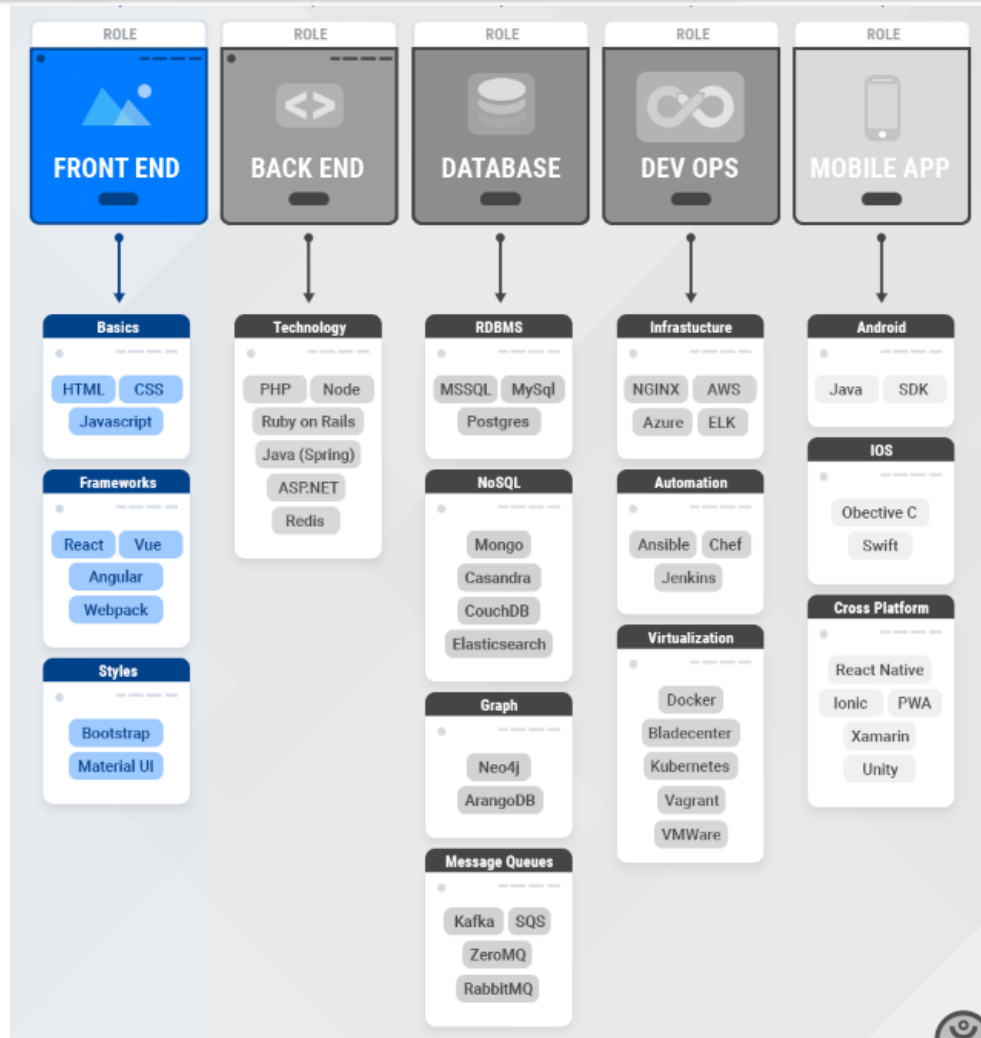
Languages and frameworks



https://res.cloudinary.com/cybercoders/image/upload/c_scale,g_south_east,l_cc_logo_bug_wenazs.png,w_40/v1557870077/Full_Stack_ebv14s.png



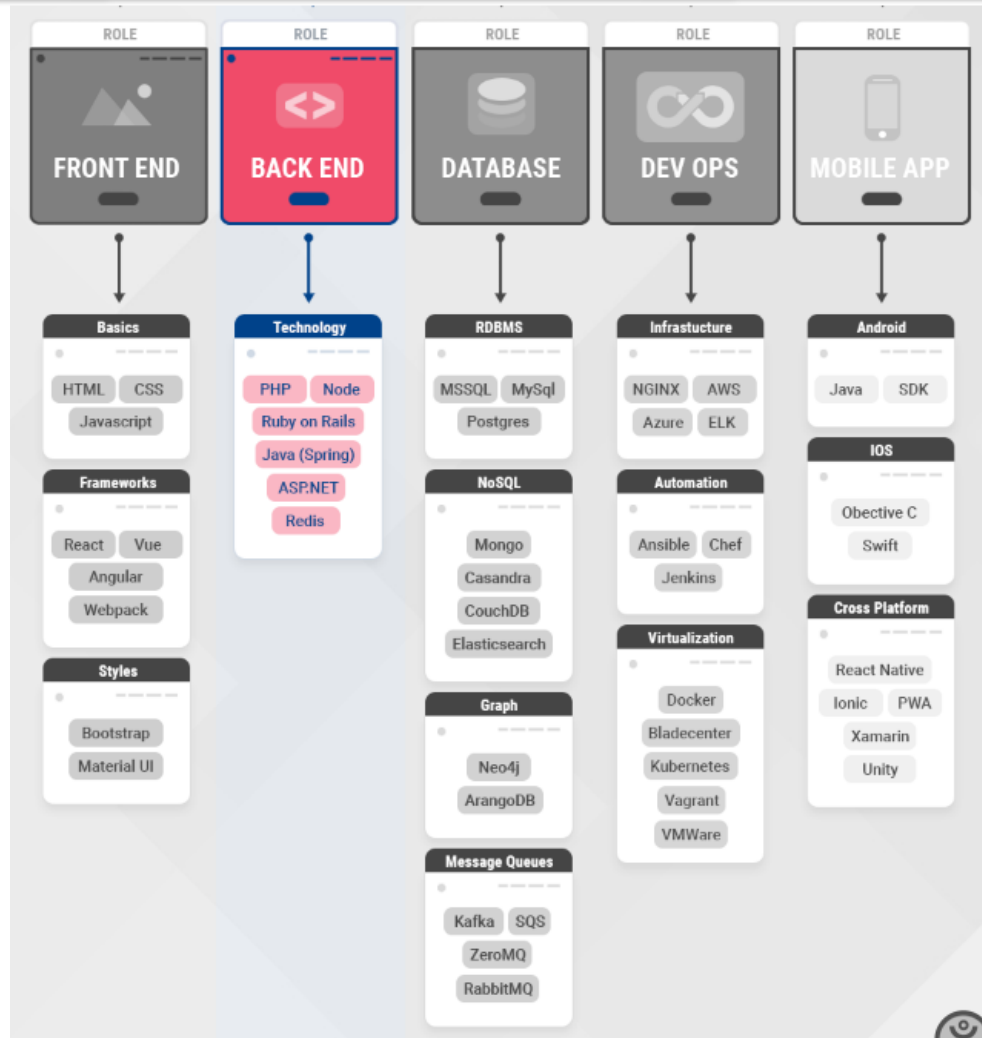
Frontend



https://res.cloudinary.com/cybercoders/image/upload/c_scale,g_south_east_l_1_cc_logo_bug_wenazs.png,w_40/v1557870077/Full_Stack_ebv14s.png



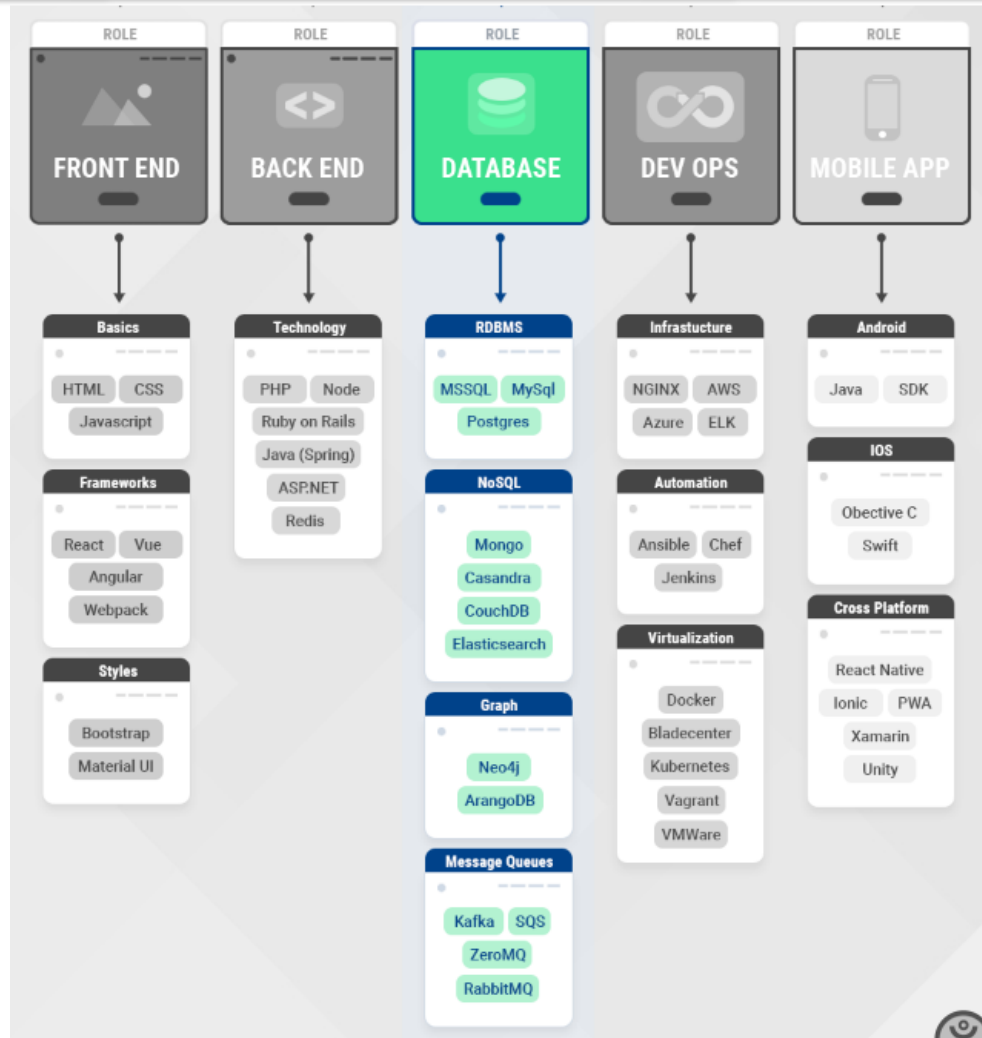
Backend



https://res.cloudinary.com/cybercoders/image/upload/c_scale,g_south_east_l_1_cc_logo_bug_wenazs.png,w_40/v1557870077/Full_Stack_ebv14s.png



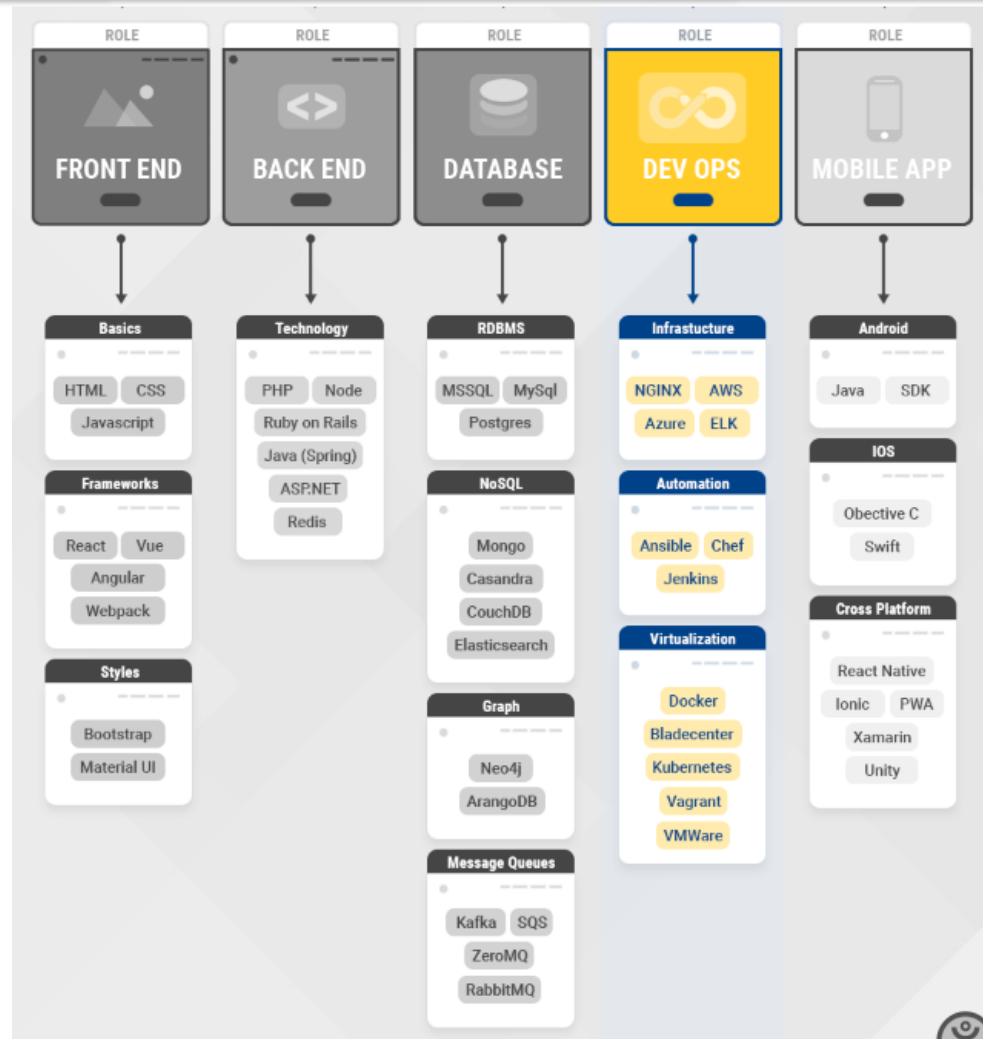
Database



https://res.cloudinary.com/cybercoders/image/upload/c_scale,g_south_east,t_l_cc_logo_bug_wenazs.png,w_40/v1557870077/Full_Stack_ebv14s.png



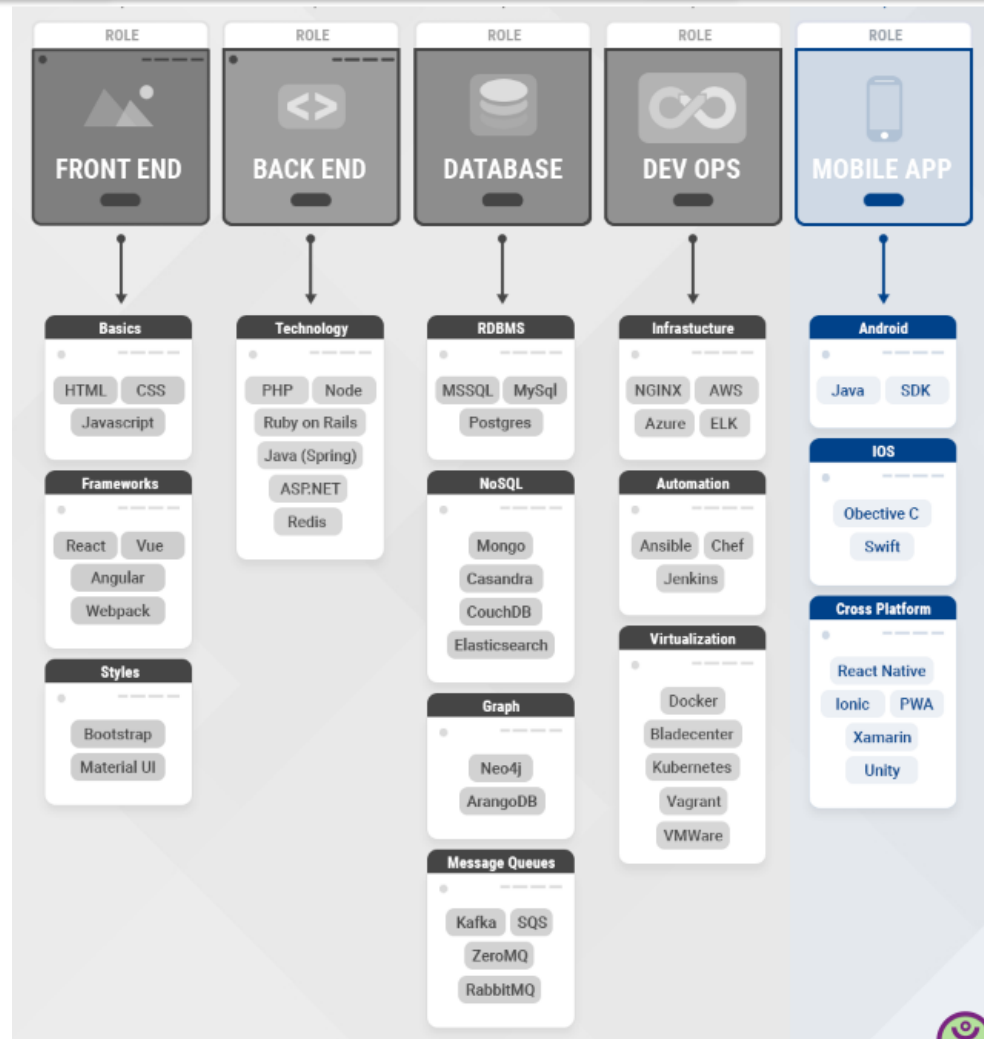
DevOps



https://res.cloudinary.com/cybercoders/image/upload/c_scale,g_south_east,t_l_cc_logo_bug_wenazs.png,w_40/v1557870077/Full_Stack_ebv14s.png



Mobile



https://res.cloudinary.com/cybercoders/image/upload/c_scale,g_south_east_l_cc_logo_bug_wenazs.png,w_40/v1557870077/Full_Stack_ebv14s.png





Tool Support

Lecture 12/Part 4

SE tasks commonly supported by tools



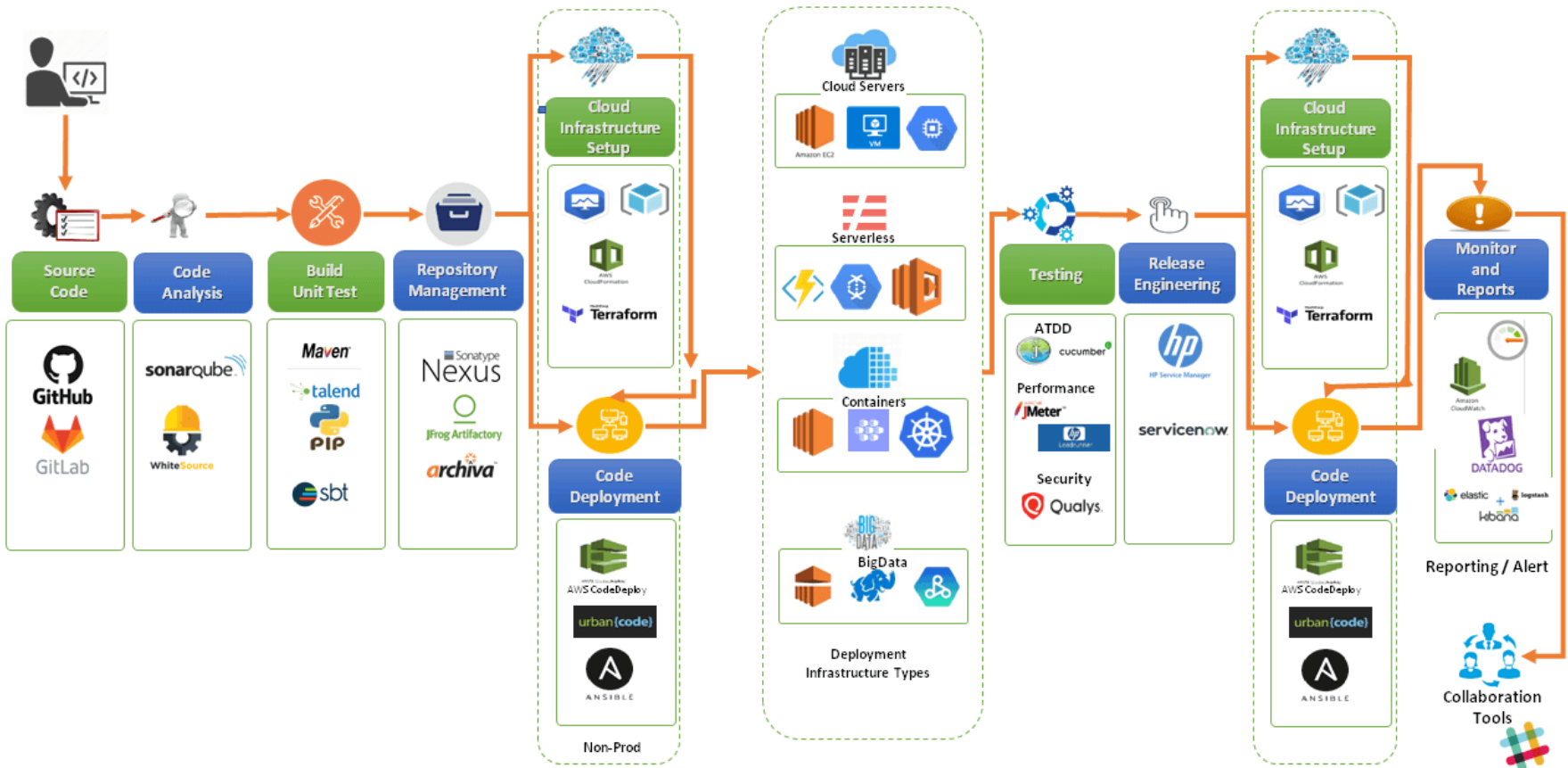
- ✧ Plan and schedule software development project
- ✧ Specify, manage and trace requirements
- ✧ Model and analyze business processes
- ✧ Create design and deployment models
- ✧ Create, edit, compile and debug code in different languages
- ✧ Generate and import database schema
- ✧ Track changes
- ✧ Manage tests
- ✧ Document software development
- ✧ Communicate and develop team based projects

Most popular tools



- ✧ Requirements analysis and design modeling tools
- ✧ Programming environments that automate parts of program construction processes (e.g., automated builds)
- ✧ Software configuration management and version control
- ✧ Testing tools including static and dynamic analysis tools
- ✧ Continuous integration and release management
- ✧ Issue tracking
- ✧ Project management tools

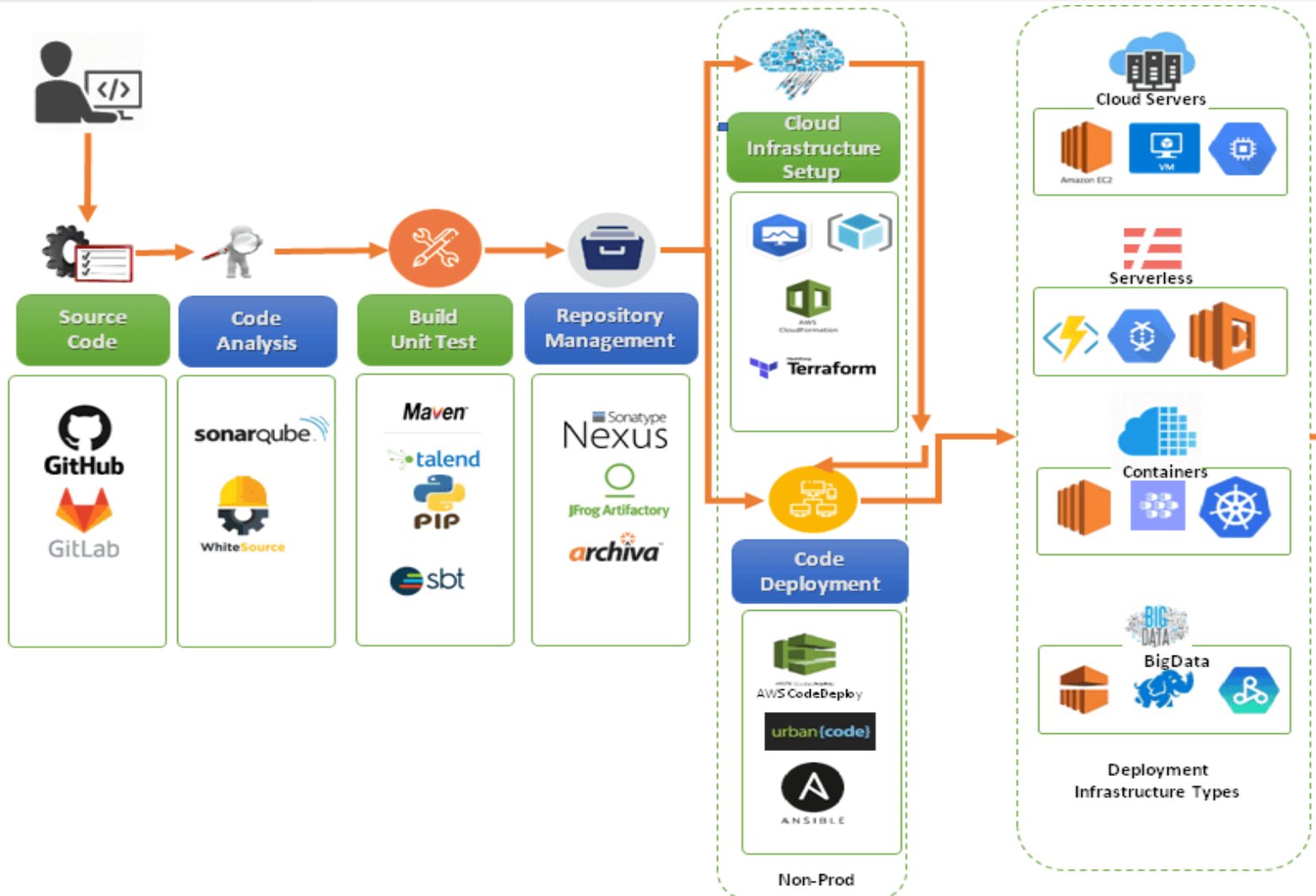
Tools



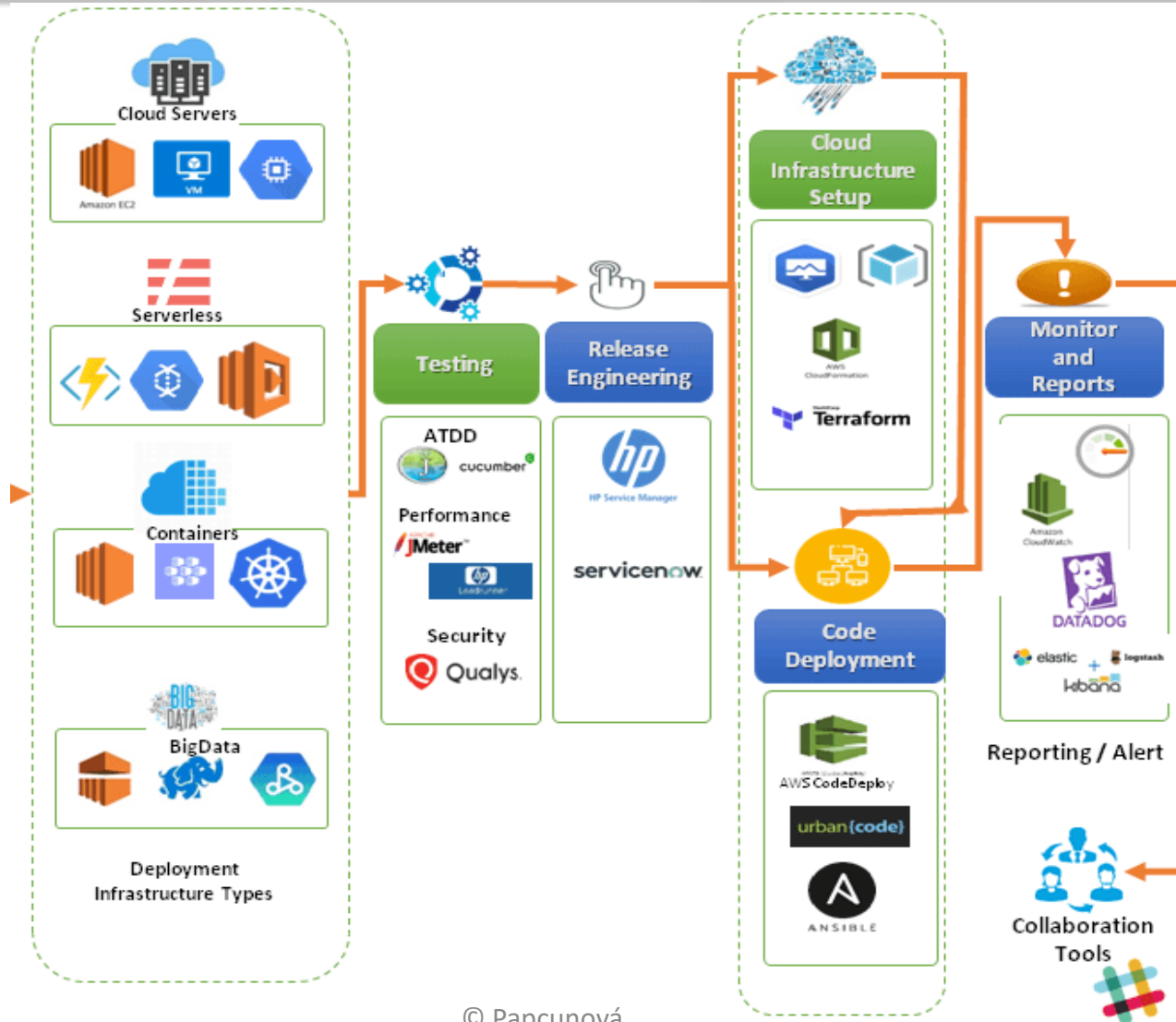
<https://www.wati.com/wp-content/uploads/2018/11/Slide3.png>



Code, infrastructure and deployment



Testing, release and collaboration



Key points



- ✧ Software engineering process can be supported by a large variety of tools.
- ✧ The specific tools are often integrated into a single environment or framework, which assists the developers through integrated support on one place.



Course Follow-up

Lecture 12/Part 5

Course finalization



✧ Seminar projects

- Assessment
- “Seminar completion / Absolvování cvičení” notebook in IS

✧ Exam

- Number of exam dates
- Reservation/cancelation policies
- Length of the exam
- Form of the exam – test part and UML modelling part
- Results and their viewing

✧ Opinion poll

- Do not forget to give us your feedback! 😊

Follow-up and related courses



- ✧ PA017 Softwarové inženýrství II
- ✧ PA103 Objektové metody návrhu informačních systémů
- ✧ PV167 Seminář s návrhových a architektonických vzorů
- ✧ PV260 Software Quality
- ✧ PV258 Software Requirements Engineering

Thanks



Thank you for your attention
and good luck with the exam!