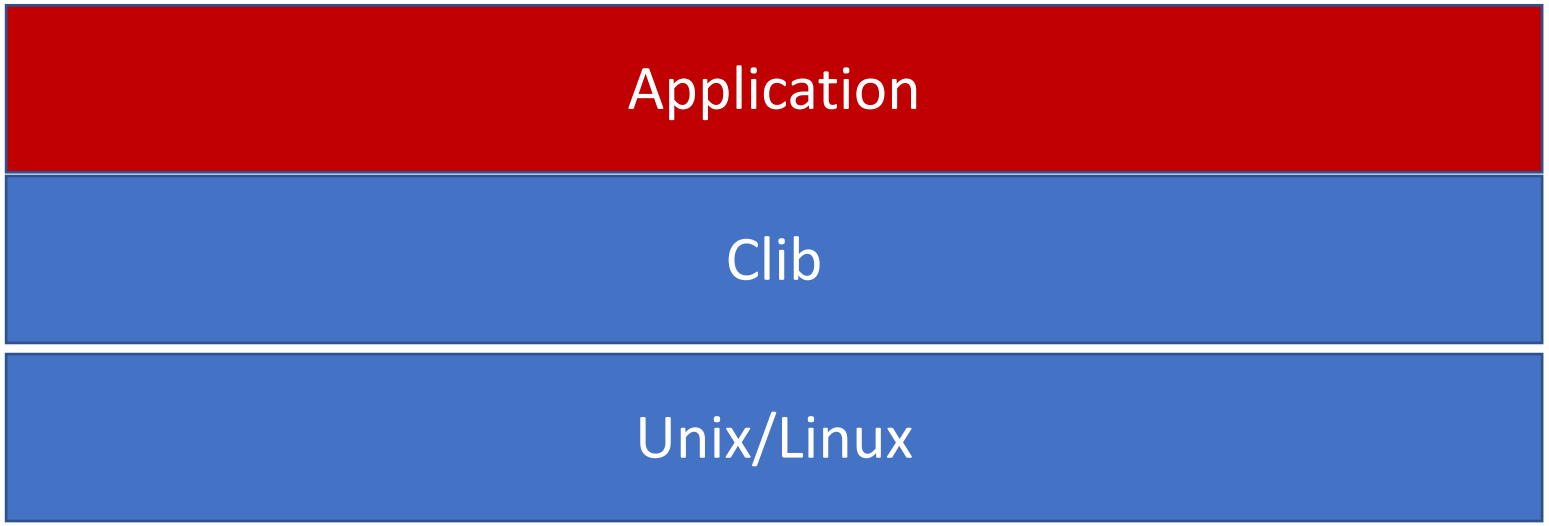
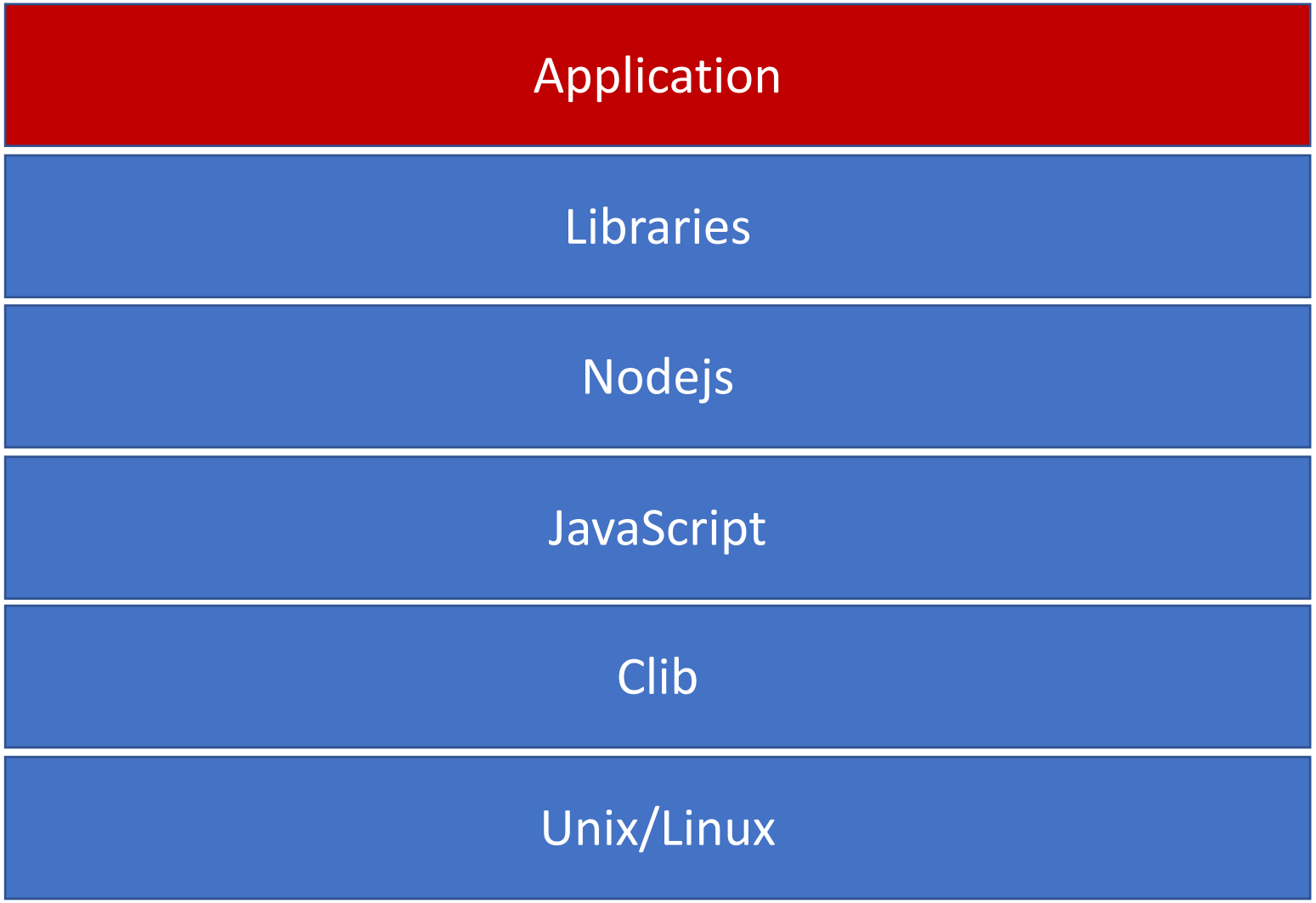
COMP.SE.140

Dependencies

Tampereen yliopisto
Tampere University


Application
Clib
Unix/Linux


Application
Libraries
Nodejs
JavaScript
Clib
Unix/Linux

Tampereen yliopisto
Tampere University

Remember container use case example

- Your application needs
  - Certain version of nodejs
  - Set of libraries (certain versions)
  - Mongo database

- Your system has
  - Wrong version of nodejs
  - Mongo serving another application

- Solution
  - Create a docker image (container)
  - Install the image
  - Run the image

Base images used in exercise 4

- node:10 20
- node:10-alpine 3
- node:10.15.3-stretch 1
- node:10.16.3-alpine 1
- node:11 1
- node:11-alpine 1
- node:12 1
- node:12.2-alpine 1
- node:8.16.1-alpine 1
- node:8.16.1-jessie-slim 1
- node:alpine 1
- node:latest 2
- golang:alpine AS builder 1
- golang:latest 1
- python 1
- python:3 2
- python:3.6 2
- python:3.7-alpine 2
- python:latest 2
- ubuntu:latest 1

Base images used in exercise 4

- node:10 20
- node:10-alpi 3
- node:10.15.3-stretch 1
- node:10.16.3-alpine
- node:11
- node:11-alpine
- node:12
- node:12.2-alpine
- node:8.16.1-alpine
- node:8.16.1-jessie-sli
- node:alpine
- node:latest

- golang:alpine AS builder 1
- golang:latest 1
- python 1
- python:3 2

2
2
2
1

node:<version>
This is the defacto image. If you are unsure about what your needs are, you probably want to use this one. It is designed to be used both as a throw away container (mount your source code and start the container to start your app), as well as the base to build other images off of.

Base images used in exercise 4

• node:10	20
• node:10-alpine	3
• node:10.15.3-stretch	1
• node:10.16.3-alpine	
• node:11	
• node:11-alpine	
• node:12	
• node:12.2-alpine	
• node:8.16.1-alpine	
• node:8.16.1-jessie-slim	
• node:alpine	
• node:latest	

• golang:alpine AS builder	1
• golang:latest	1
• python	1

node:<version>-slim
This image does not contain the common packages contained in the default tag and only contains the minimal packages needed to run node. Unless you are working in an environment where *only* the node image will be deployed and you have space constraints, we highly recommend using the default image of this repository.

Base images used in exercise 4

- node:10
- node:10-alpine
- node:10.15.3
- node:10.16.3-alpine
- node:11
- node:11-alpine
- node:12
- node:12.2-alpine
- node:8.16.1-alpine
- node:8.16.1-jessie-slim
- node:alpine
- node:latest

node:<version>-alpine

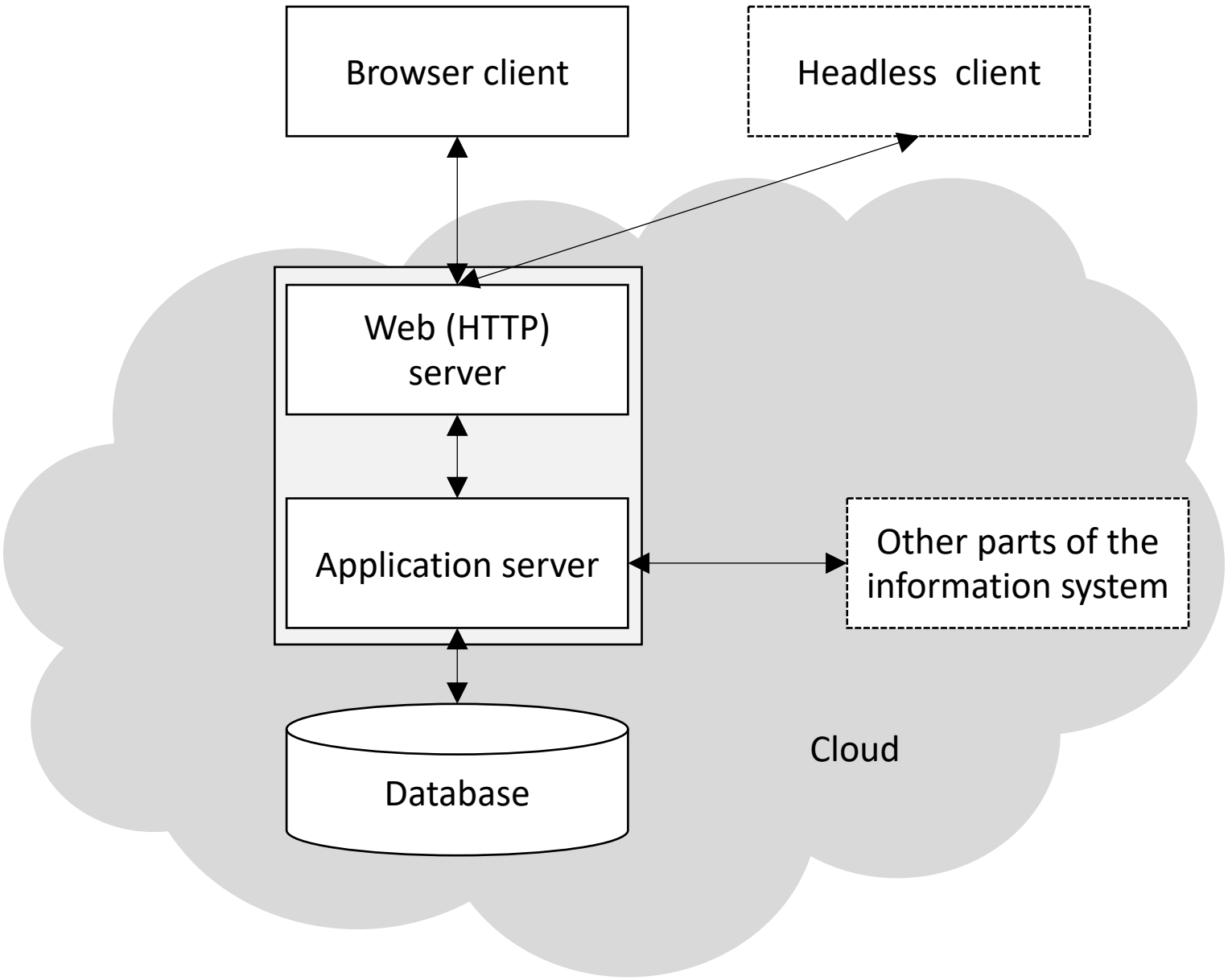
This image is based on the popular Alpine Linux project, available in the alpine official image. Alpine Linux is much smaller than most distribution base images (~5MB), and thus leads to much slimmer images in general.

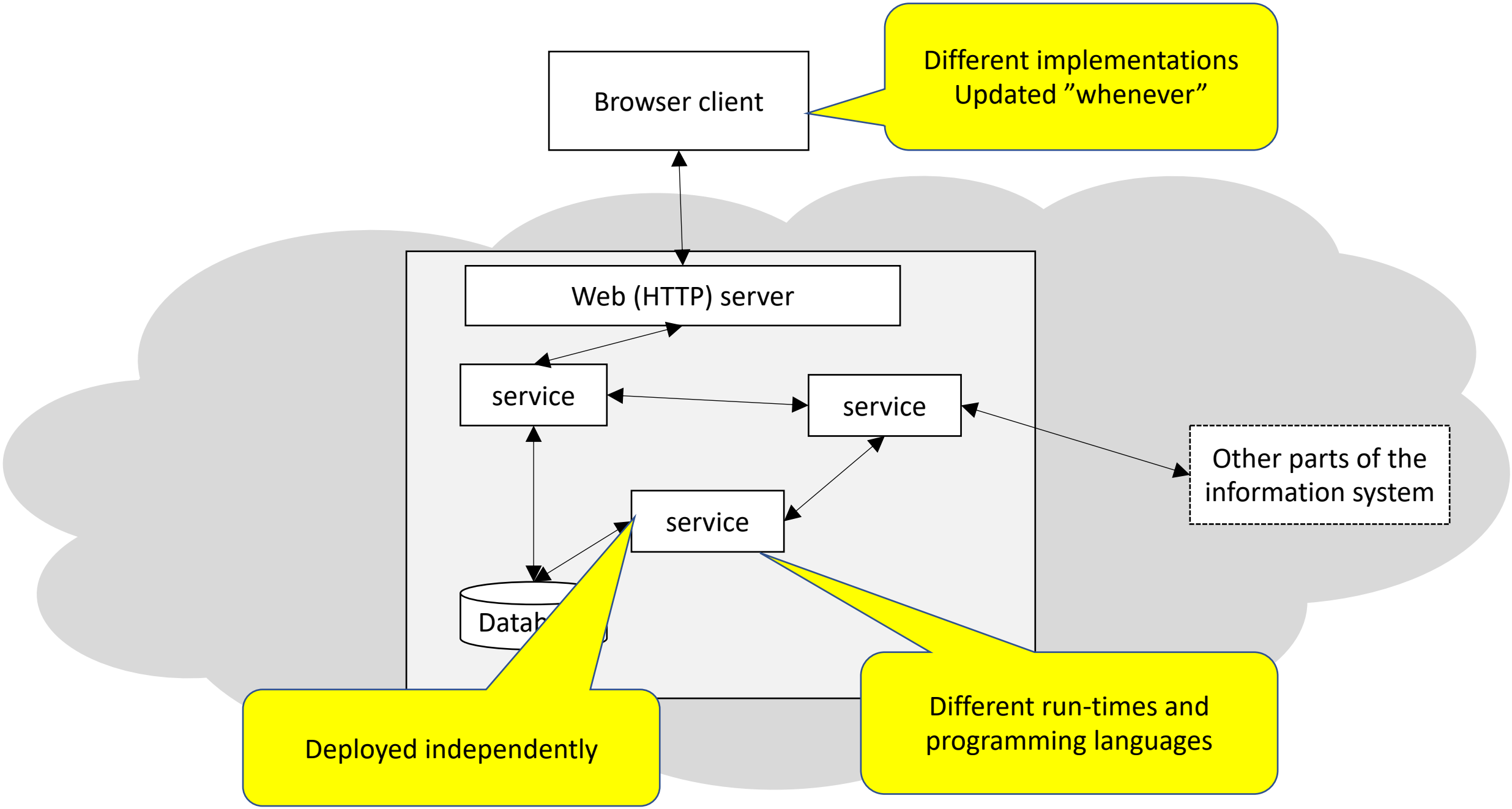
This variant is highly recommended when final image size being as small as possible is desired. The main caveat to note is that it does use musl libc instead of glibc and friends, so certain software might run into issues depending on the depth of their libc requirements. However, most software doesn't have an issue with this, so this variant is usually a very safe choice. See this Hacker News comment thread for more discussion of the issues that might arise and some pro/con comparisons of using Alpine-based images.

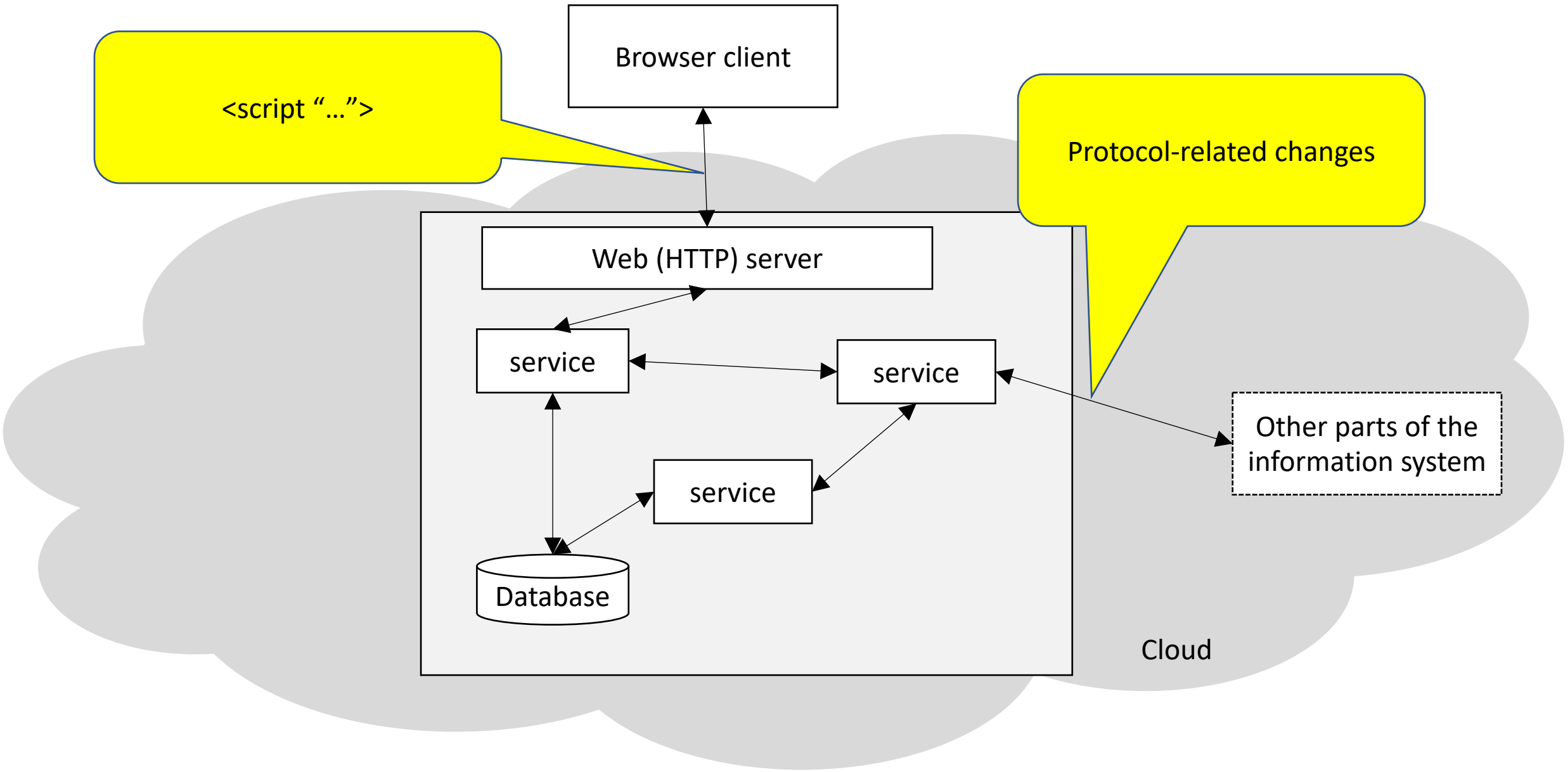
Base images used in exercise 4

node:10	20
node:10-alpine	3
node:10.15.3-stretch	1
node:10.16.3-alpine	1
node:11	2
node:11-alpine	2
node:12	2
node:12.2-alpine	1
node:8.16.1-alpine	
node:8.16.1-jessie-slim	
node:alpine	
node:latest	
golang:alpine AS builder	1
golang:latest	1
python	1
python:3	2

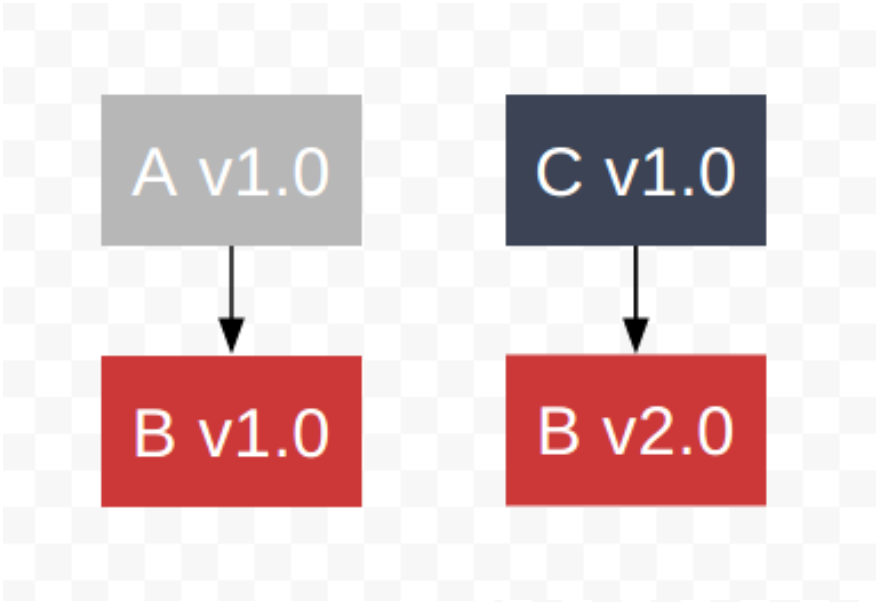
Some of these tags may have names like buster, jessie, or stretch in them. These are the suite code names for releases of Debian and indicate which release the image is based on. If your image needs to install any additional packages beyond what comes with the image, you'll likely want to specify one of these explicitly to minimize breakage when there are new releases of Debian.

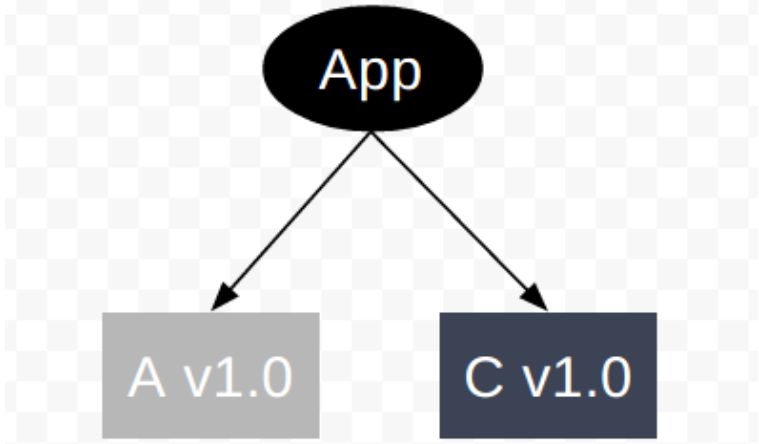

Browser client
Headless client
Web (HTTP) server
Application server
Other parts of the information system
Database
Cloud

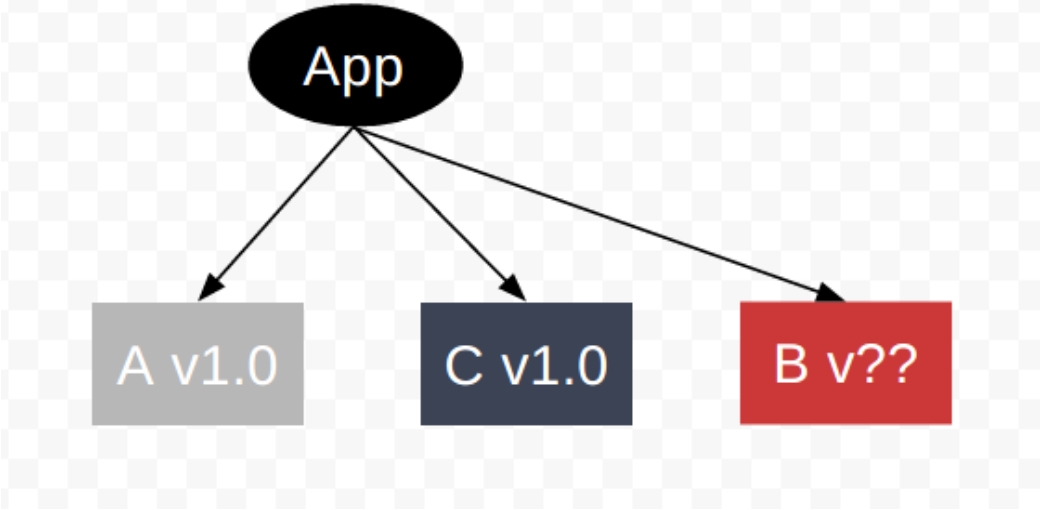

Browser client
Different implementations
Updated ”whenever”
Web (HTTP) server
service
service
service
Other parts of the
information system
Deployed independently
Different run-times and
programming languages


<script "...">
Browser client
Protocol-related changes
Web (HTTP) server
service
service
service
Database
Other parts of the information system
Cloud

https://npm.github.io/how-npm-works-docs/theory-and-design/dependency-hell.html







The old way

Static approach

- Libraries come with the compiler, or are intalled to the development tool
- Compiler integrates application with libraries
- The integrated system is deployed to users

The Web & Cloud way

Dynamic approach

- Libraries are downloaded in a dynamic manner
- Huge number of libraries available, use each other, and are frequently updates (continuous delivery)

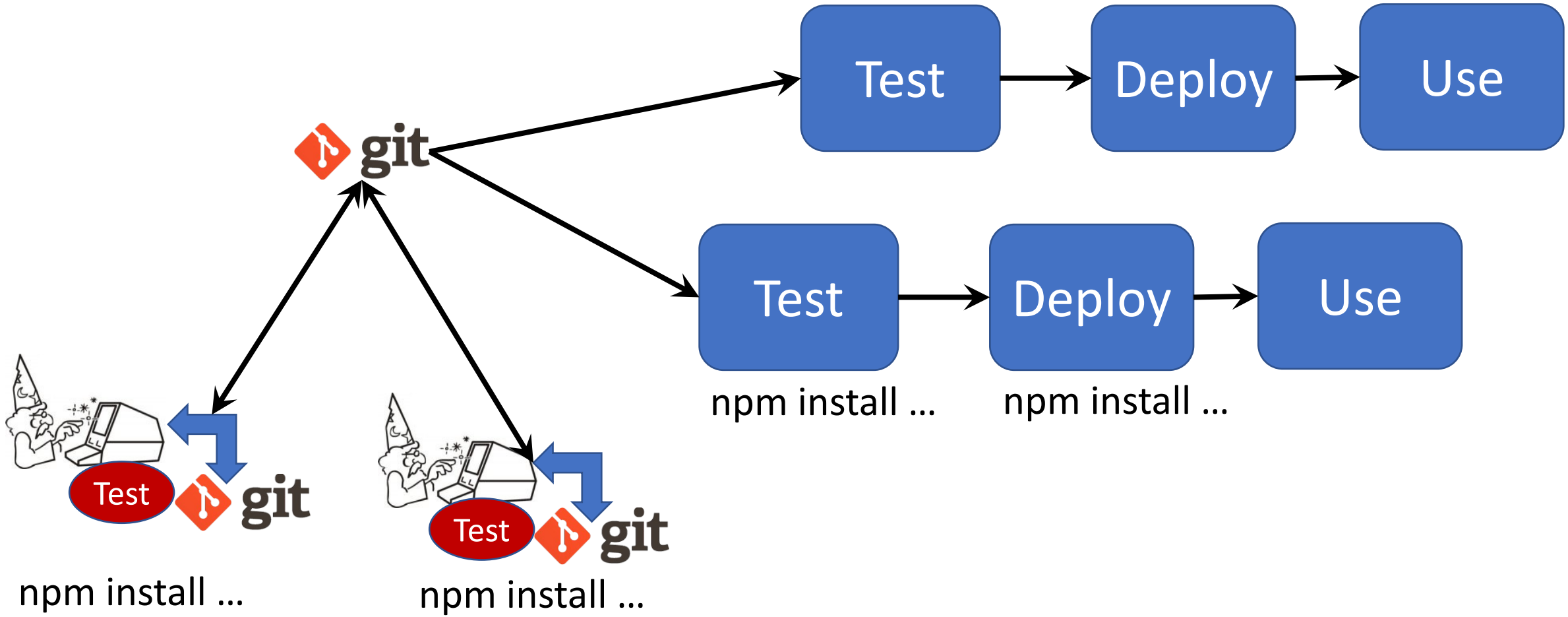
- npm
- pip

Tampereen yliopisto
Tampere University

Package.json

```
{
  "name": "service1",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "express": "4.17.1",
    "request": "2.88.0"
  }
}
```

Development vs use



package-lock.json is automatically generated for any operations where npm modifies either the node_modules tree, or package.json. It describes the exact tree that was generated, such that subsequent installs are able to generate identical trees, regardless of intermediate dependency updates.

This file is intended to be committed into source repositories, and serves various purposes:

- Describe a single representation of a dependency tree such that teammates, deployments, and continuous integration are guaranteed to install exactly the same dependencies.
- Provide a facility for users to “time-travel” to previous states of node_modules without having to commit the directory itself.
- To facilitate greater visibility of tree changes through readable source control diffs.
- And optimize the installation process by allowing npm to skip repeated metadata resolutions for previously-installed packages.

Tampereen yliopisto
Tampere University

Package.json

```
{
  "name": "service1",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "express": "4.17.1",
    "request": "2.88.0"
  }
}
```

- Does not say anything about versions of packages used by express and request
- Allows use of later versions

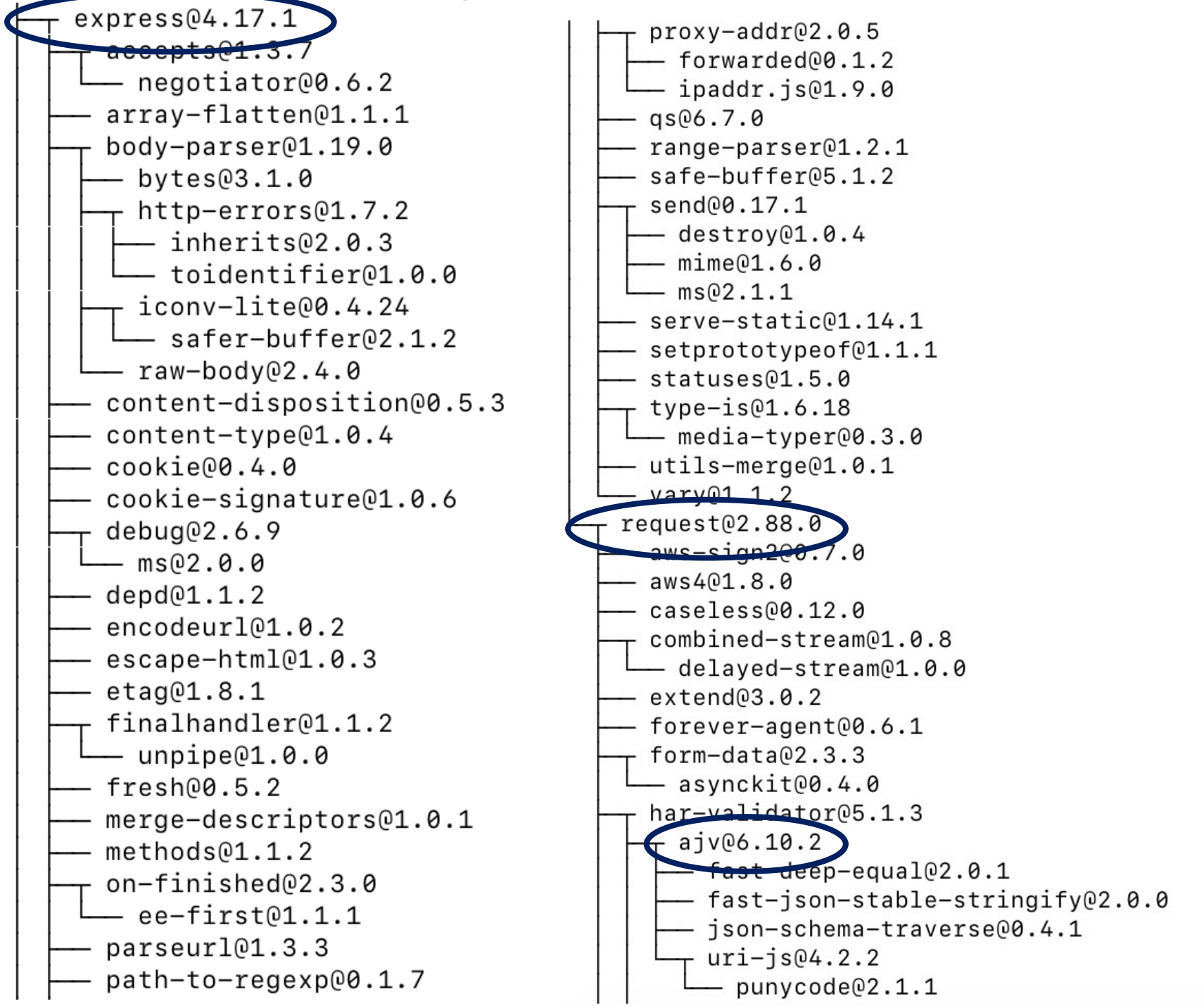
- **What is a possible problem with this?**

Package-lock.json


```
{
  "name": "service1",
  "version": "1.0.0",
  "lockfileVersion": 1,
  "requires": true,
  "dependencies": {
    "accepts": {
      "version": "1.3.7",
      "resolved": "https://registry.npmjs.org/accepts/-/accepts-1.3.7.tgz",
      "integrity": "sha512-Il80Qs2WjYlJIBNzNkK6KYqlVMTbZLXgHx2oT0pU/fjRHyEp+PEfEPY0R3WCwAGVOtauxh1hOxNgIf5bv7dQpA==",
      "requires": {
        "mime-types": "~2.1.24",
        "negotiator": "0.6.2"
      }
    },
    "ajv": {
      "version": "6.10.2",
      "resolved": "https://registry.npmjs.org/ajv/-/ajv-6.10.2.tgz",
      "integrity": "sha512-TXtUUEYHuaTEbLZWIKUr5pmBuhDLy+8KYtPYdcV8qC+pOZL+NKqYwvWSRrVXHn+ZmRRAu8vJTAznH7Oag6RVRw==",
      "requires": {
        "fast-deep-equal": "^2.0.1",
        "fast-json-stable-stringify": "^2.0.0",
        "json-schema-traverse": "^0.4.1",
        "uri-js": "^4.2.2"
      }
    },
```

Tampereen yliopisto
Tampere University

npm ls



Tampereen yliopisto
Tampere University

Nodejs has also changed (for example)

2018-10-23, Version 11.0.0 (Current), @jasnell

Notable Changes

Build

FreeBSD 10 is no longer supported. #22617

child_process

The default value of the windowsHide option has been changed to true. #21316

console

console.countReset() will emit a warning if the timer being reset does not exist. #21649

console.time() will no longer reset a timer if it already exists. #20442

Dependencies

V8 has been updated to 7.0. #22754

fs

The fs.read() method now requires a callback. #22146

The previously deprecated fs.SyncWriteStream utility has been removed.#20735

Tampereen yliopisto
Tampere University

Npm versions

- 6.12.0 15 days ago
- 6.12.0-next.0 a month ago
- 6.11.3 2 months ago
- 6.11.2 2 months ago
- 6.11.1 2 months ago
- 6.11.0 2 months ago
- 6.10.3 3 months ago
- 6.10.2 3 months ago

Tampereen yliopisto
Tampere University

New in NPM 5

2. Lockfiles

With npm@5, lockfiles are the default (package-lock.json). This simply means that whatever files you get when you install a package will be the same every time you install that package after initial install. This eliminates the challenges developers had with having different files on different developer environments after installing the same package.

And the language (ECMAScript / JavaScript)

ES5 (2009)

- This is the baseline version of JS which you can generally assume all run-times (except really old ones!) will support.

ES6 / ES2015

- Standard Modules — import and export
- Standardised Promises
- Classes & Inheritance
- Block-scoped variables — let and const
- Template Literals
- Object destructing into variables
- Generator functions
- Map and Set data structures
- Internationalisation for Strings, Numbers and Dates via Intl API

ES7 / ES2016

- Array.includes()
- Numeric exponent (power of) operator **

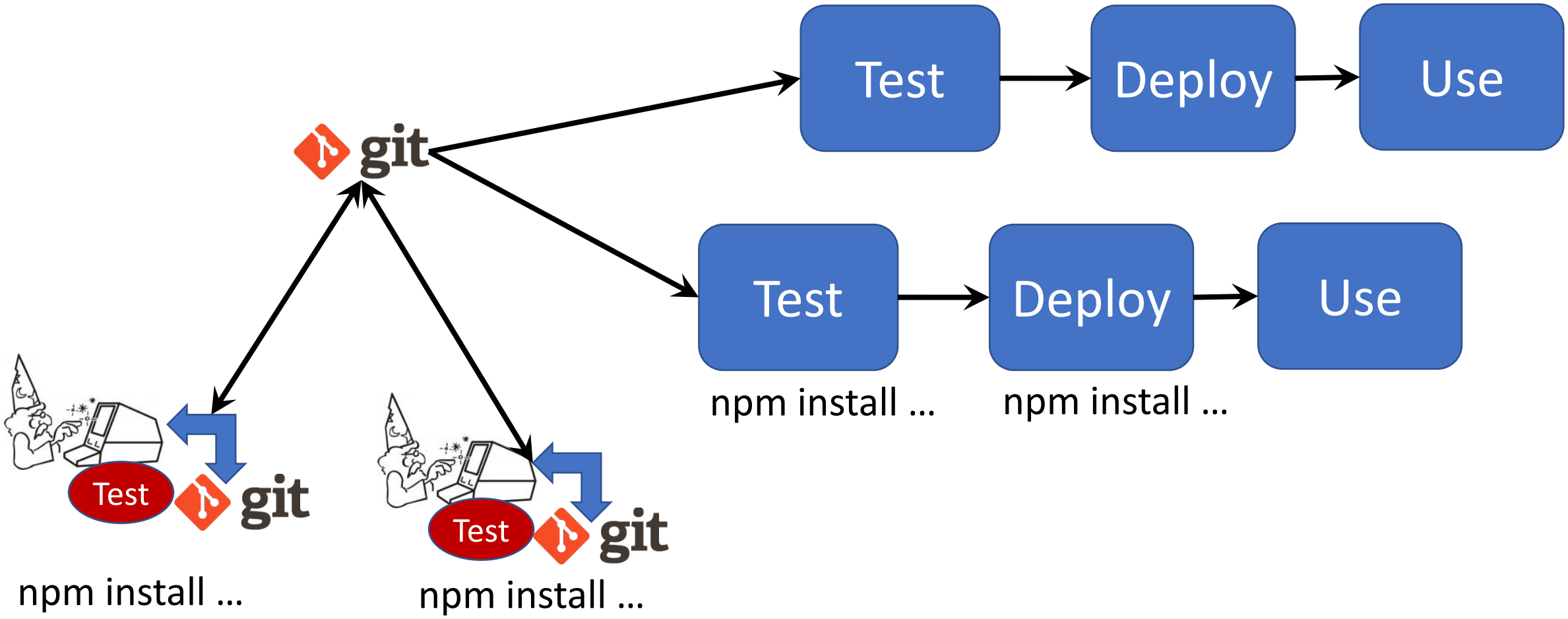
ES8 / ES2017

- Async Functions
- Object.entries
- String padding functions

ES9 / ES2018

- Object Rest/Spread const obj = { ...props };
- Asynchronous Iteration for await (...) {
- Promise finally() function
- Regular expression enhancements (lookbehind, named groups)

Development vs use



Base images used in exercise 4

- node:10 20
- node:10-alpine 3
- node:10.15.3-stretch 1
- node:10.16.3-alpine 1
- node:11 1
- node:11-alpine 1
- node:12 1
- node:12.2-alpine 1
- node:8.16.1-alpine 1
- node:8.16.1-jessie-slim 1
- node:alpine 1
- node:latest 2

- golang:alpine AS builder 1
- golang:latest 1
- python 1
- python:3 2
- python:3.6 2
- python:3.7-alpine 2
- python:latest 2
- ubuntu:latest 1

git clone ..
docker-compose up

Tampereen yliopisto
Tampere University

Towards solutions?

- Build a docker image and deploy that?

- Use package-lock.json and installation scripts?