REST API, Express, Publishing

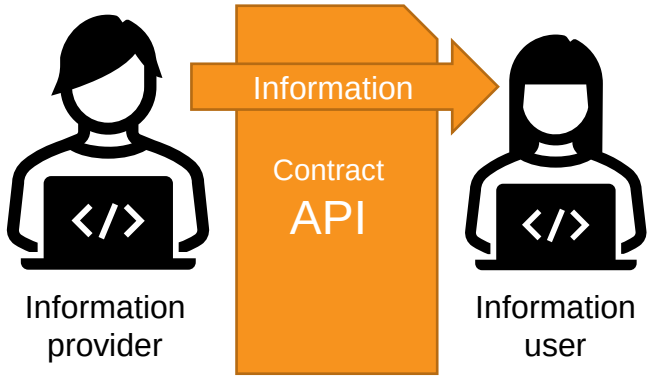
Web Programming Project / Meija Lohiniva, OAMK 2022

Current Matters

- Design document comments
  - Sorry for the delay! Work in progress
  - Not enough info -> no comments
- Schedule
  - 11th April: Connecting to Database, Licencing and packages
  - 25th April: ?? Suggest a topic in https://moodle.oulu.fi/mod/feedback/view.php?id=681505!
- Book a weekly meeting with the teacher.
  - Anne Keskitalo: 1,3,5,8,10
  - Meija Lohiniva: 2,4,7,9,11

API

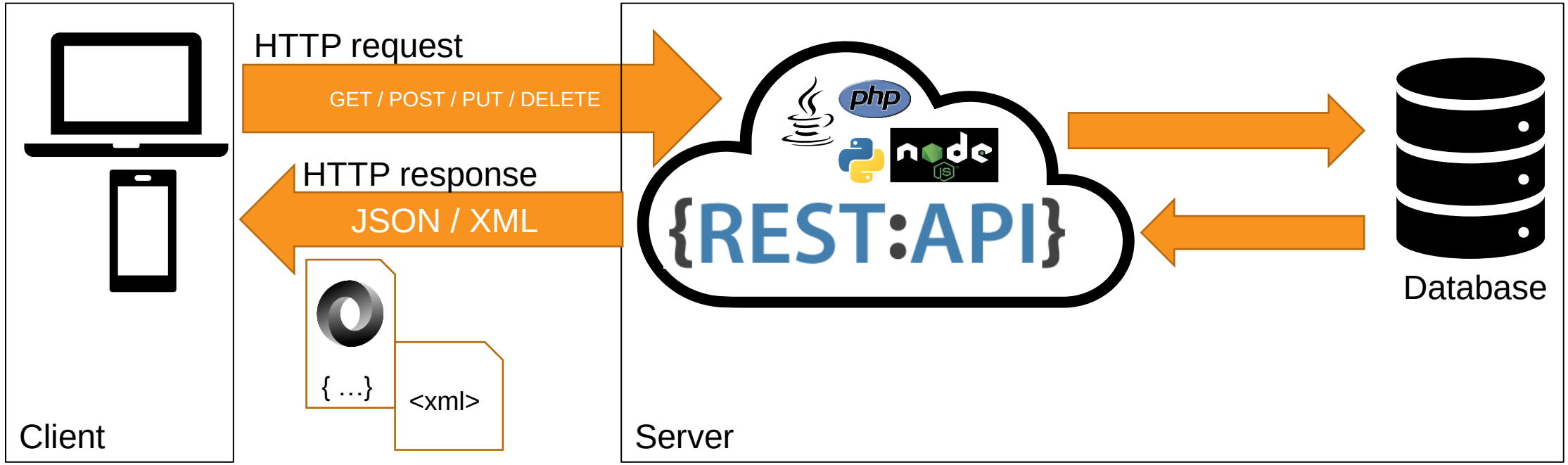
- API = Application Programming Interface
  - "A set of definitions and protocols for building and integrating application software" *(https://www.redhat.com/en/topics/api/what-is-a-rest-api#:~:text=A%20REST%20API%20(also%20known,by%20computer%20scientist%20Roy%20Fielding.)*



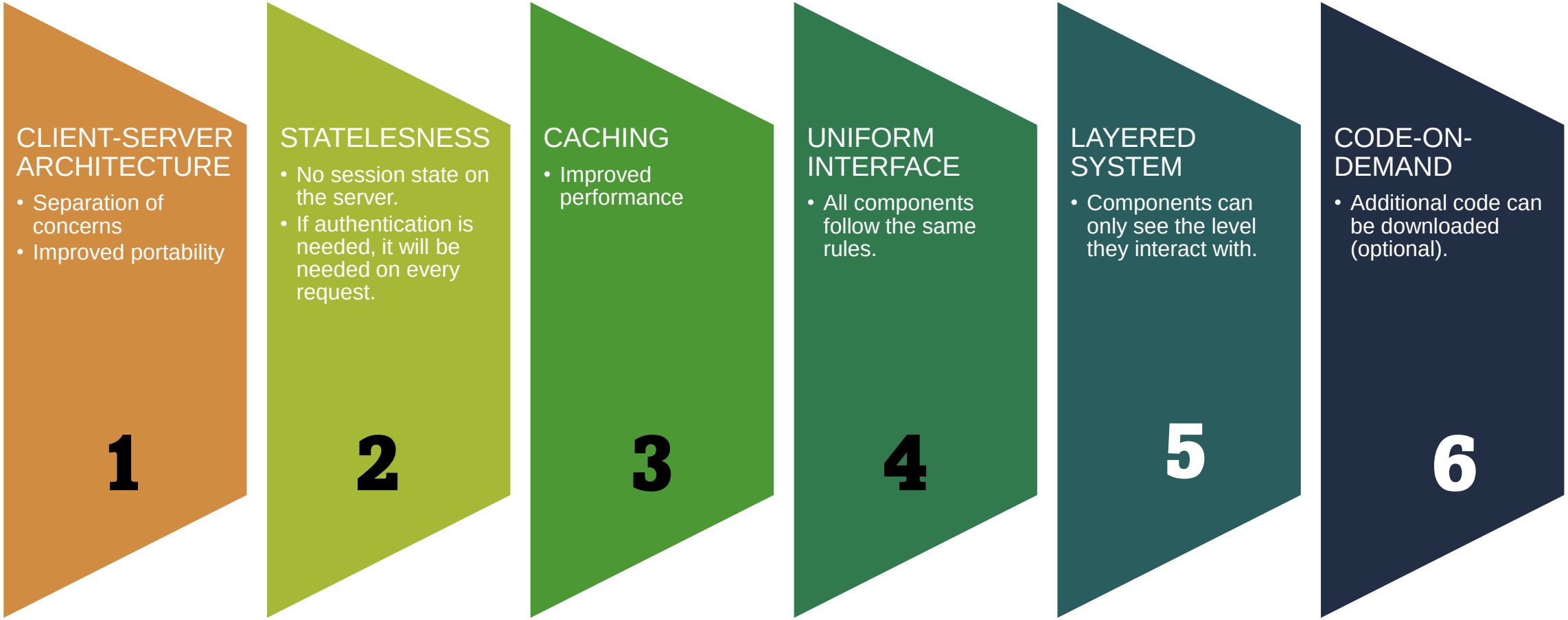
REST API

- REST API = RESTful API = Representational State Transfer.
- Software architectural style.
- Created by Roy Fielding in 2000.
- An intermediary between client and server.

REST API


HTTP request
GET / POST / PUT / DELETE
HTTP response
JSON / XML
{ …}
<xml>
Client
{REST:API}
Database
Server

REST API – The Six Principles



REST API – Benefits & Disadvantages

- Benefits
  - Separation of UI from data.
    - Portability
    - Scalability
    - Independent development of components
    - Flexibility
    - Language independence
- Disadvantages
  - Lack of standardisation.

HTTP

POST	GET	PUT	DELETE
• Add data • Data in body	• Show data • Data in header	• Update data • Data in body • Id in URL	• Delete data • Id in URL

Express.js

- Backend web application framework.
  - Web applications
  - API's
- Fast & easy

Task 1: Simple Express Server

1. Init a new project (`npm init`).
2. Install Express (`npm install express -save`).
3. Create `app.js`.
4. Test your server with node `app.js`.

App.js

```
let express = require("express");
let app = express();
app.listen(3000, () => {
 console.log("Server running on port 3000");
});
```

Task 2: Simple REST API

Continue with the app from previous slide.

1. Add a handler for HTTP get request to return a string array with any data.
2. Test with url http://localhost:3000/fruits

App.js

```
let express = require("express");
let app = express();
app.get("/fruits", (req, res, next) => {
    res.json(["Banana","Apple","Kiwi"]);
   });
app.listen(3000, () => {
 console.log("Server running on port 3000");
});
```

REST API Best Practices

- Use JSON for sending and receiving data.
  - Set the Content-Type type in the response header to application/json while making the request.
  - Express has the express.json() middleware for this purpose.
- Use Nouns Instead of Verbs in Endpoints
  - Instead of getPosts use posts
  - HTTP methods handle the verbs.
- Name Collections with Plural Nouns.
- Use Status Codes in Error Handling.
- Use Nesting on Endpoints to Show Relationships.
  - Don't make nesting too deep (max three levels).

REST API Best Practices

- Use Filtering, Sorting, and Pagination to Retrieve the Data Requested.
- Use SSL (Secure Socket Layer) for Security.
- Be Clear with Versioning.
- Provide Accurate API Documentation.

Deployment in OAMK's Server

- You can publish web sites on OAMK's web server.
  - Support for PHP etc
  - You cannot publish Node.js servers.
- You can publish databases on OAMK's database server.
  - To connect, you need to have VPN. (This is not a free feature in Heroku.)

Deployment in Heroku

- Instructions: https://devcenter.heroku.com/articles/deploying-nodejs
- Register to Heroku.
- Install Heroku CLI.
- In your app make sure…
  - …you have package.json file.
  - …you use environment variable to define the port to listen to
  - …your script is named server.js or you have a procfile or start script defined.
  - …you're not relying on any system level dependencies (`rm -rf node_modules; npm install –production; heroku local web`)
  - …you have node version set in package.json

```
const PORT = process.env.PORT || 8080;
app.listen(PORT, () => {
```

```
"engines": { "node": "14.x" },
```

Deployment in Heroku

If you get this error

```
PS C:\Opintojaksot\Web Programming Project\sampleCode\express> heroku local web
heroku : File C:\Users\meijaloh\AppData\Roaming\npm\heroku.ps1 cannot be loaded because running scripts is disabled on this system. For mor
e information, see about_Execution_Policies at https:/go.microsoft.com/fwlink/?LinkID=135170.
At line:1 char:1
+ heroku local web
+ ~~~~~~
    + CategoryInfo          : SecurityError: (:) [], PSSecurityException
    + FullyQualifiedErrorId : UnauthorizedAccess
```

add the following to your settings.json file

```
"terminal.integrated.profiles.windows": {
      "PowerShell": {
        "source": "PowerShell",
        "icon": "terminal-powershell",
        "args": ["-ExecutionPolicy", "Bypass"]
      }
    },
    "terminal.integrated.defaultProfile.windows":
"PowerShell",
```

Deployment in Heroku

.gitignore

```
/node_modules
npm-debug.log
.DS_Store
/*.env
```

- Create .gitignore to ignore node_modules.
- Deploy your app to heroku:
  - Initialize git repo.
  - Commit changes
  - Login to heroku (`heroku login`)
  - Create app in heroku (`heroku create`)
  - Push your code to heroku (`git push heroku master`)
  - Open your app in heroku (`heroku open`)
    - To run your deployed app locally use `heroku local`.

Deployment in Heroku

If you get the error

```
PS C:\Opintojaksot\Web Programming Project\sampleCode\express> heroku login
 »   Warning: heroku update available from 7.60.0 to 7.60.1.
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/d02ff51a-1866-4052-ad1e-fc5821b235c2?requestor=SFMyNTY.g2gDb
QAAAAw4NS4yOS42NS4xMzNuBgAwpYvwfwFiAAFRgA.ZqXxNx_sqde-XTOpxzBcTr6Sv5fXGC0rSiB0mN1r4Ns
Logging in... done
Error: ENOENT: no such file or directory, open 'Z:/_netrc'
```

try running heroku commands in Git Bash or Command prompt.

Deployment in Heroku

Database

- Install Heroku Postgres (`heroku addons:create heroku-postgresql:hobby-dev`)
- Install Postgres on your local machine.
- Install Postgres on your Node.js app (`npm install pg`).

Deployment in Heroku

Best Practices

- Always start projects with `npm init`.
- Define engine in `package.json` to match your node version.
- File names in lower-kebab-case.
- Use environment variables to avoid environment-specific config.
- Gitignore generated files like `node_modules`.

Group Work

1. Go to your breakout room.
2. Continue working on the project.

Teacher will visit groups.

OULUN AMMATTIKORKEAKOULU