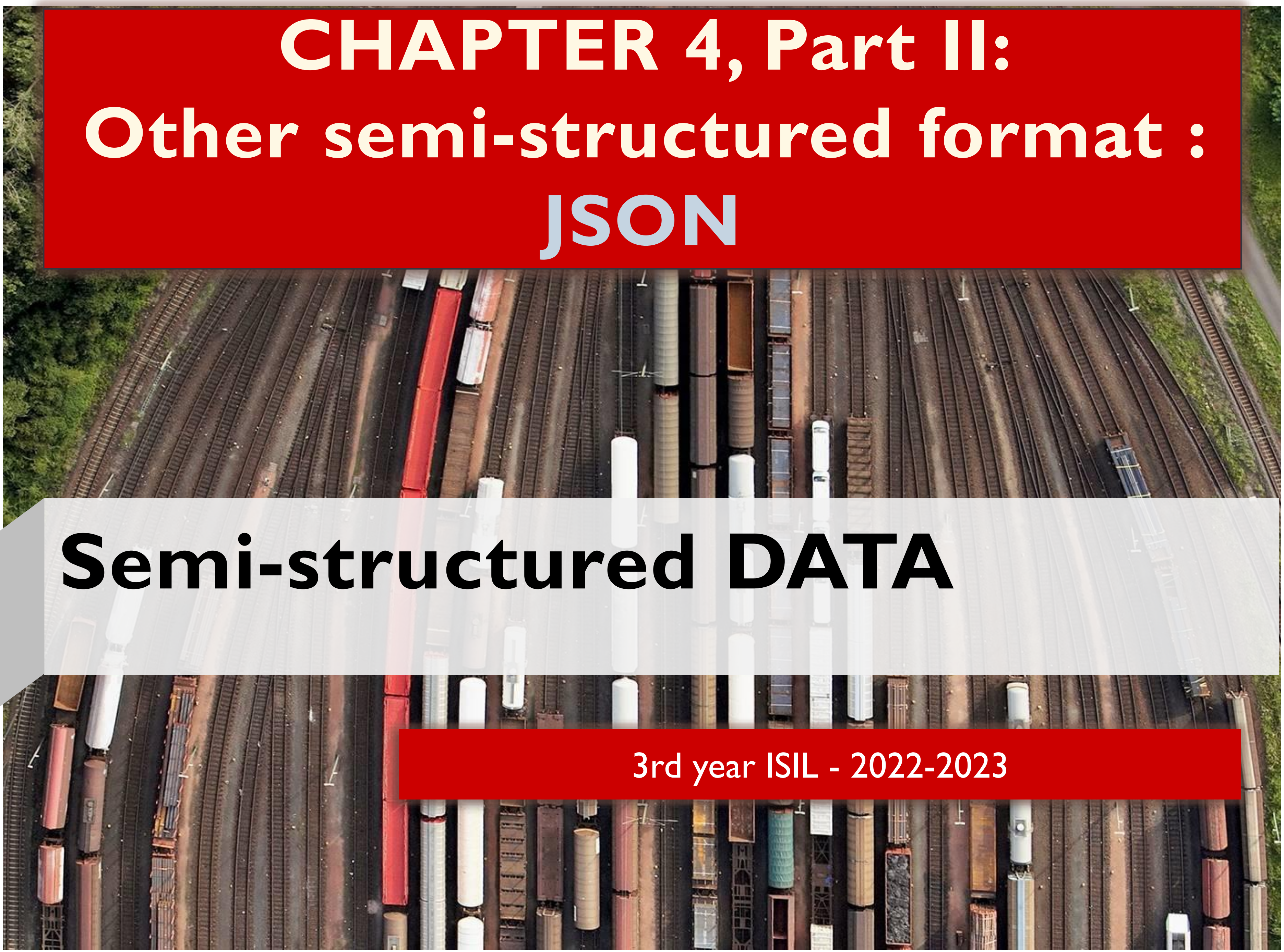
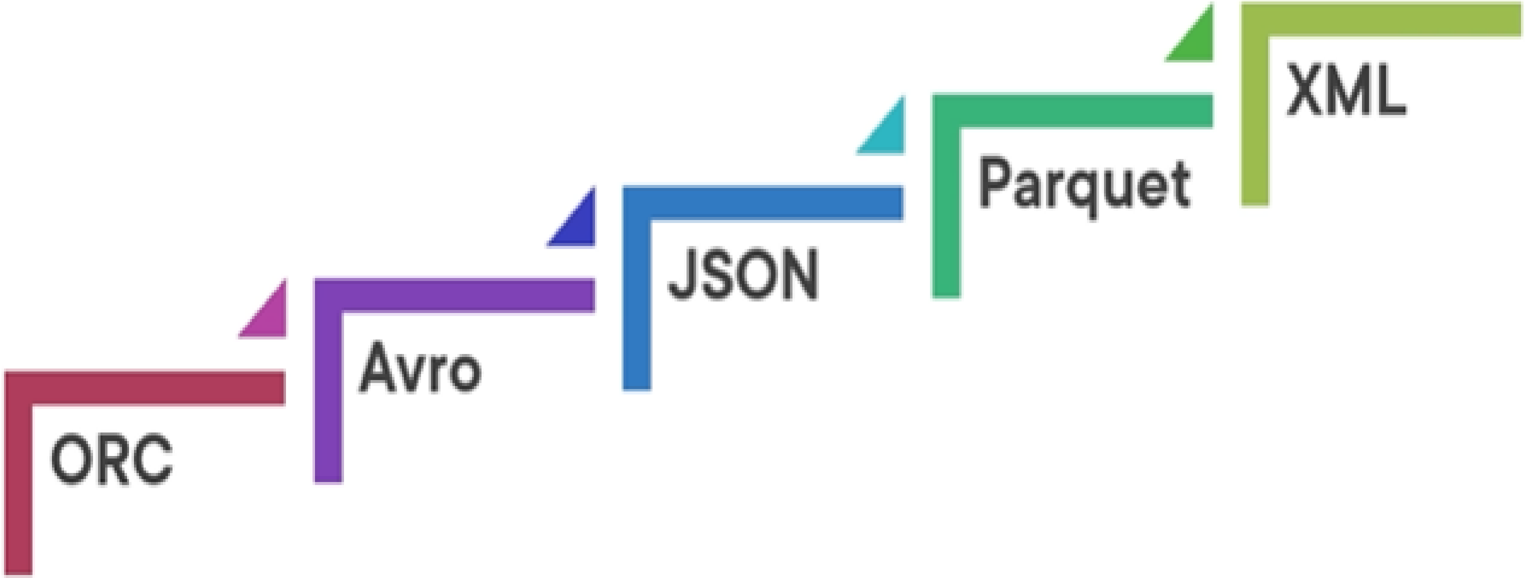
# CHAPTER 4, Part II: Other semi-structured format : JSON



## Semi-structured DATA

3rd year ISIL - 2022-2023

# Other Semi-structured file formats



# JSON (JavaScript Object Notation)

- It is a **lightweight** text-based **interchange** format (file .json).
- JSON is standardized,.
- JSON is easy to read and write than XML format (alternative to XML)
- JSON format is used to **transfer structured data** over network connection from one machine to another in a truly portable way. For e.g. JSON is used to transmit data between a server and web applications.

# JSON (JavaScript Object Notation)

- JSON is used in almost all places where data exchange is needed between two modules : JavaScript based applications that includes browser extensions and websites, Rest API calls, Flask based Python applications or be in Java based micro service.

# JSON uses cases

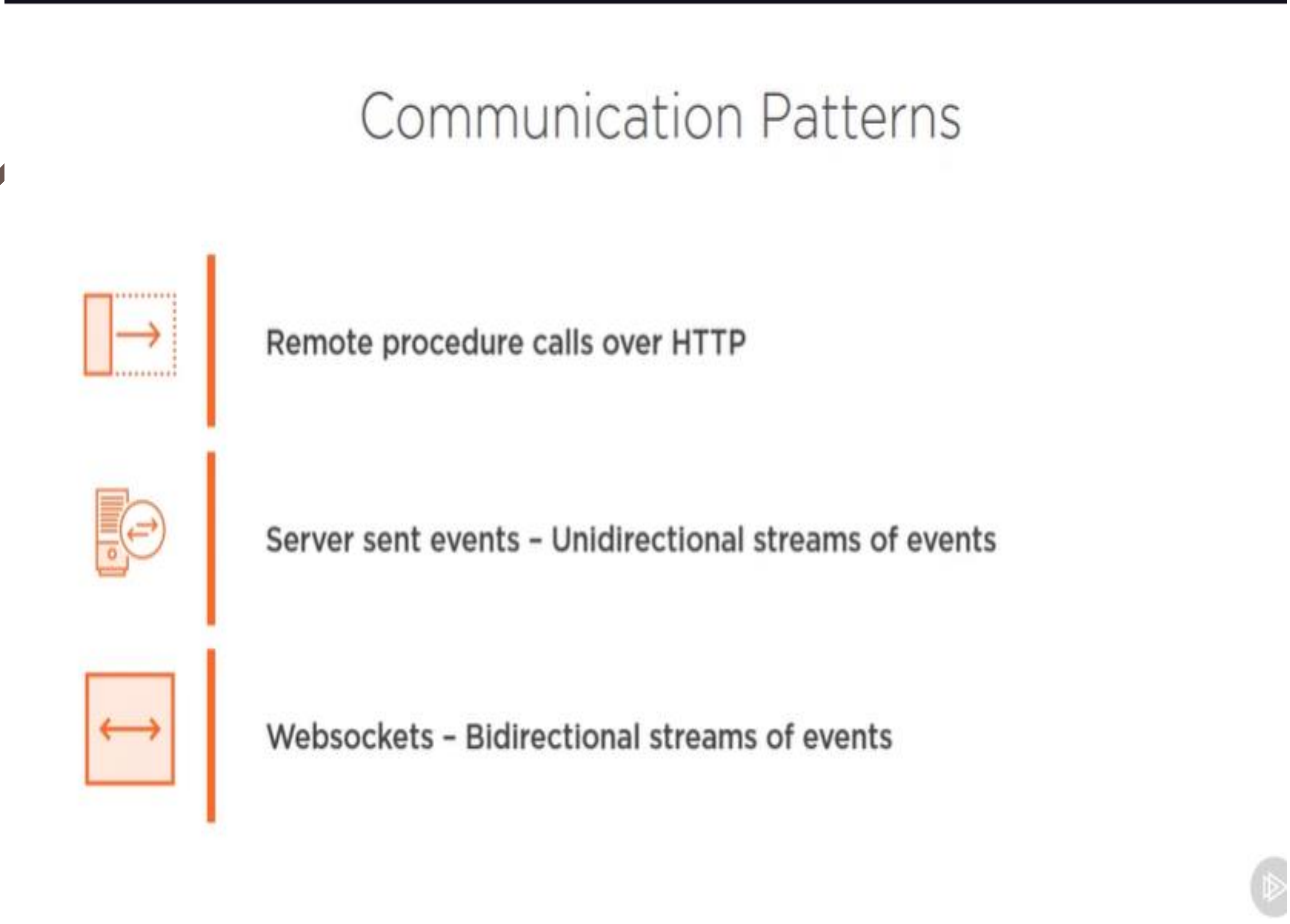
1. Communicating with Web Browsers

**Web Browser**

Sends and receives JSON

**Web Service**

Sends and receives JSON



# Communication Patterns

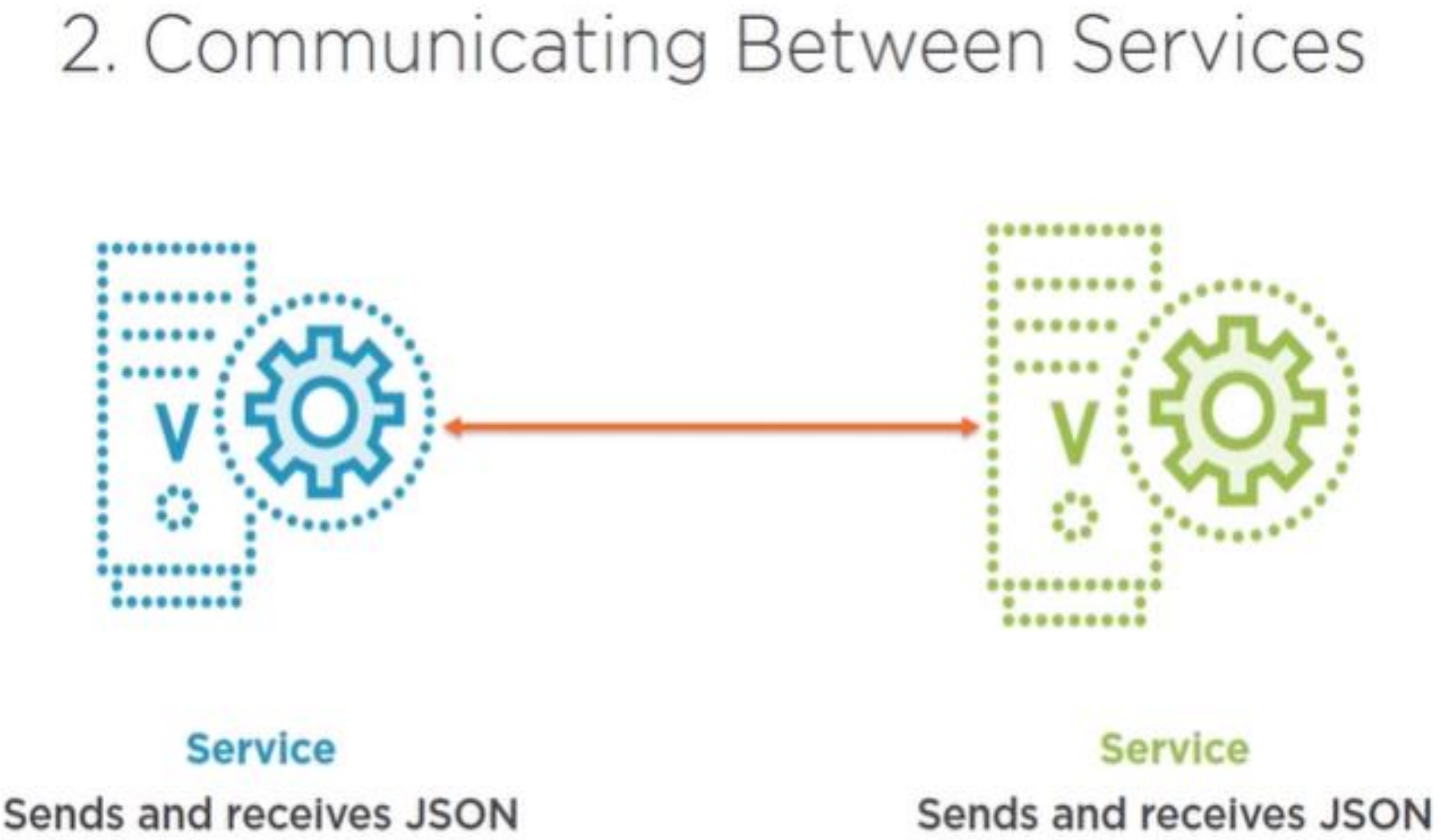
Remote procedure calls over HTTP

Server sent events - Unidirectional streams of events

Websockets - Bidirectional streams of events

JSON represents data. It is not tied to a transport technology

# JSON uses cases

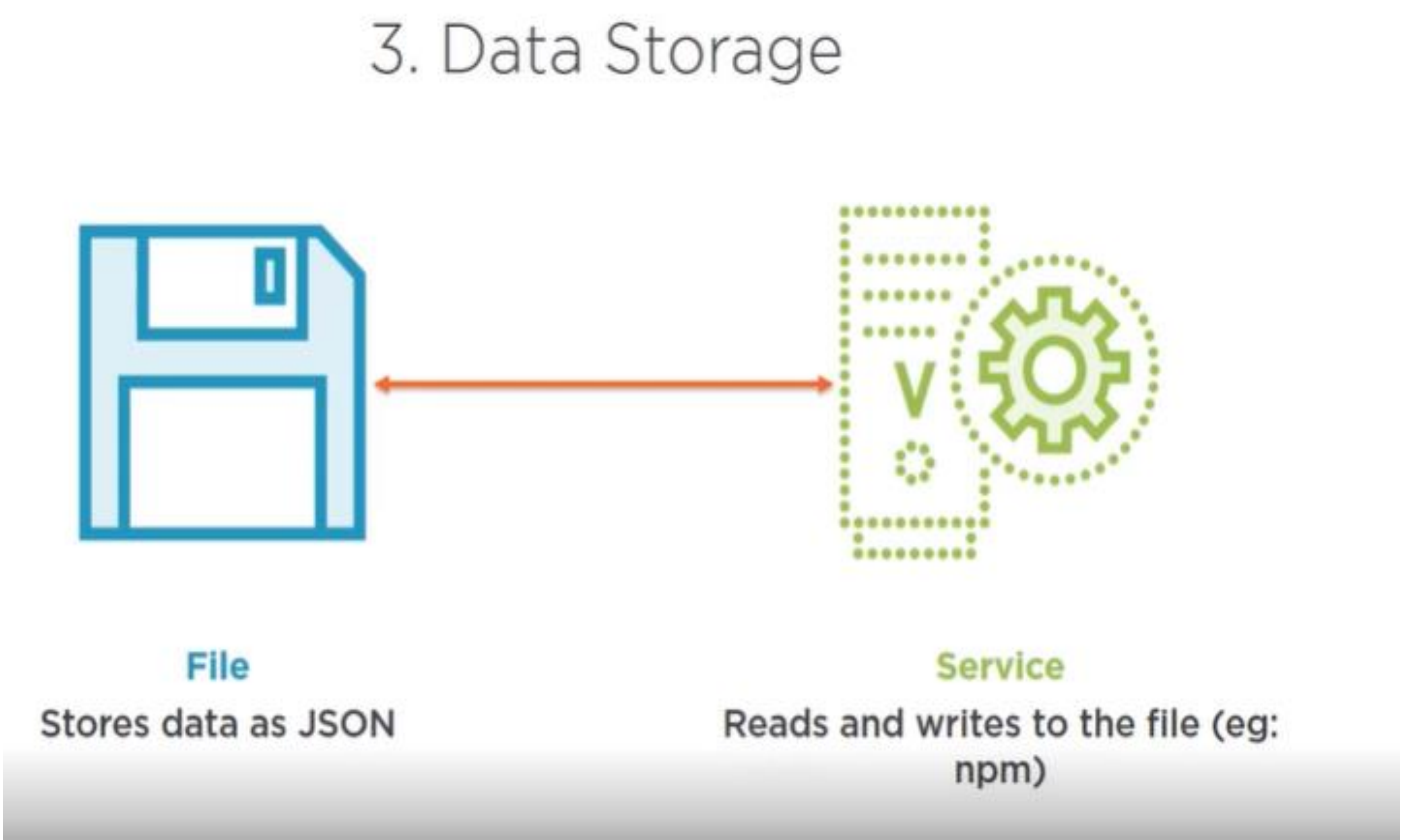


2. Communicating Between Services

Service
Sends and receives JSON

Service
Sends and receives JSON

# JSON uses cases



## 3. Data Storage

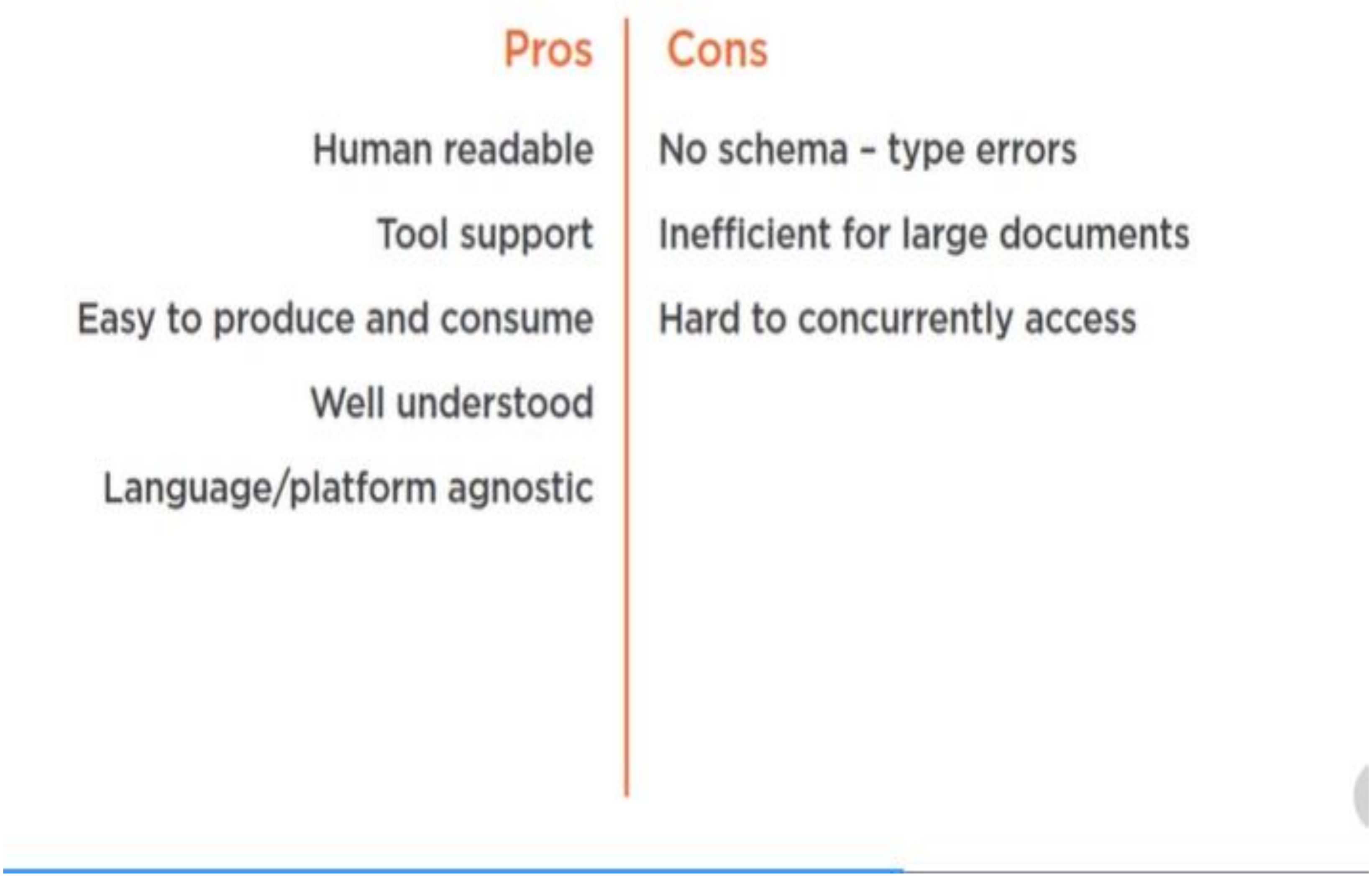
**File**

Stores data as JSON

**Service**

Reads and writes to the file (eg: npm)

# JSON as a File Format



| Pros | Cons |
|---|---|
| Human readable | No schema - type errors |
| Tool support | Inefficient for large documents |
| Easy to produce and consume | Hard to concurrently access |
| Well understood | |
| Language/platform agnostic | |

# JSON DATA type

- Typed objects, while XML is not

  - JSON types include : **object, array, string, numbers, boolean value, null.**

  - XML data is always string (a text file).

# JSON systax

- JSON follows JavaScript syntax (object syntax).

1. Data is stored and transmitted in the form of name/value pairs with colon ‘:’ used as separator, inside a curly braces{} :

   **{“name1 “: “value1”, “name 2“: “value2”, }.**

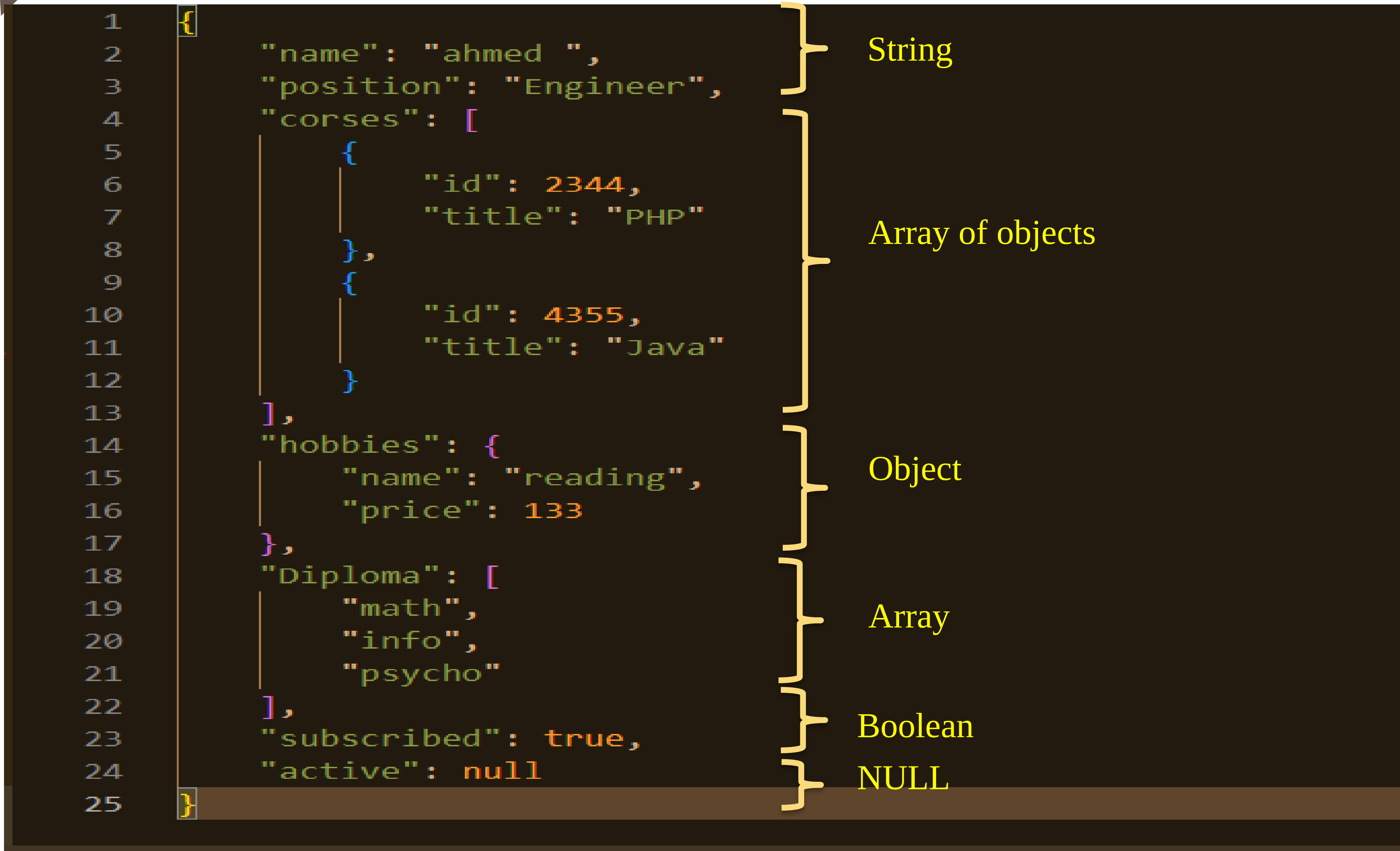
2. Square brakets for arrays, and {} for ojects.

3. Square brackets [] hold list or arrays

4. The key within the JSON should not be repeated

5. As a programmer, you must validate your json file before using it. There are many freely available JSON validation programs such as JSONLint.

# Exemple



```
{
    "name": "ahmed ",
    "position": "Engineer",
    "corses": [
        {
            "id": 2344,
            "title": "PHP"
        },
        {
            "id": 4355,
            "title": "Java"
        }
    ],
    "hobbies": {
        "name": "reading",
        "price": 133
    },
    "Diploma": [
        "math",
        "info",
        "psycho"
    ],
    "subscribed": true,
    "active": null
}
```

String

Array of objects

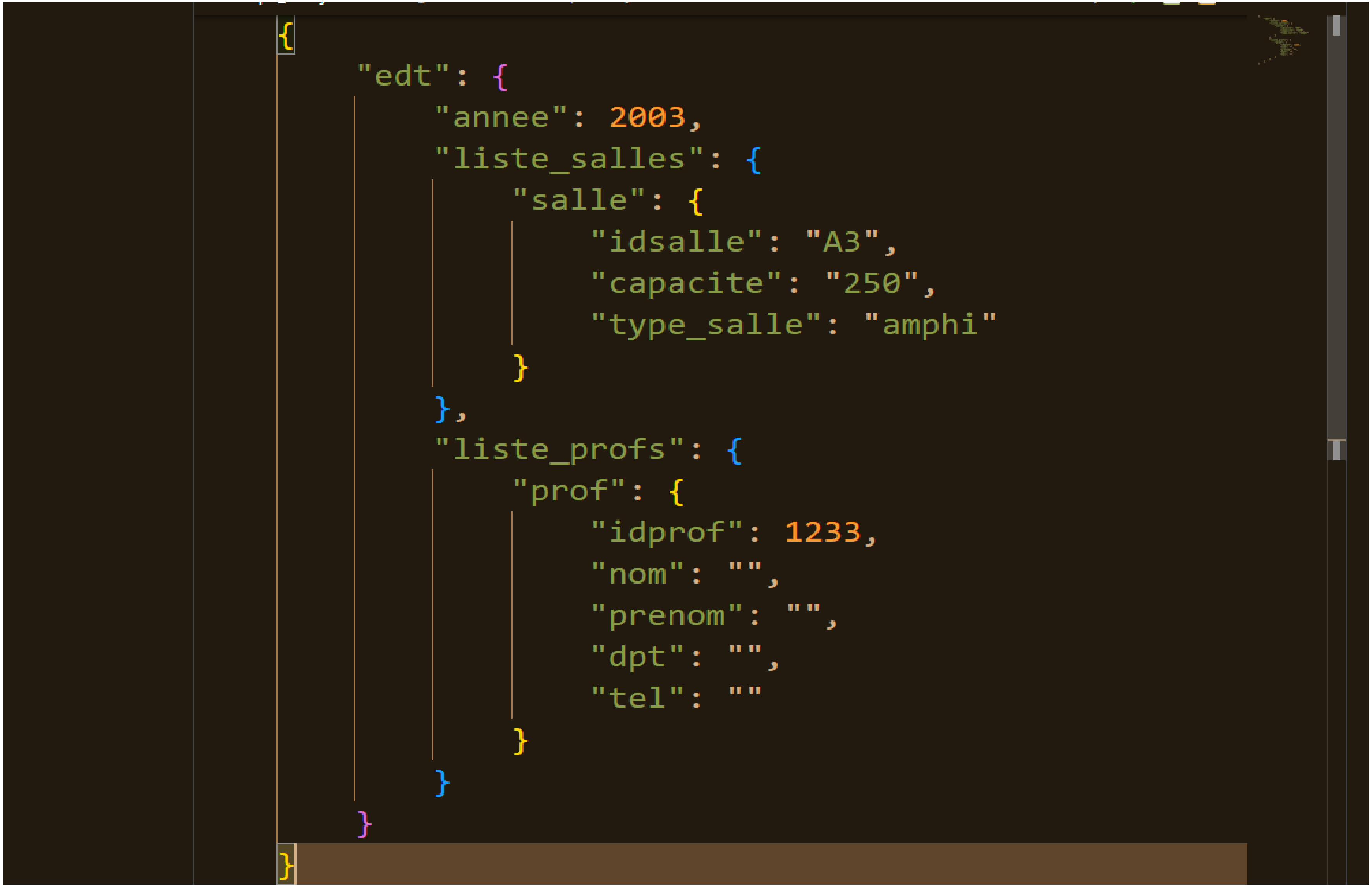
Object

Array

Boolean

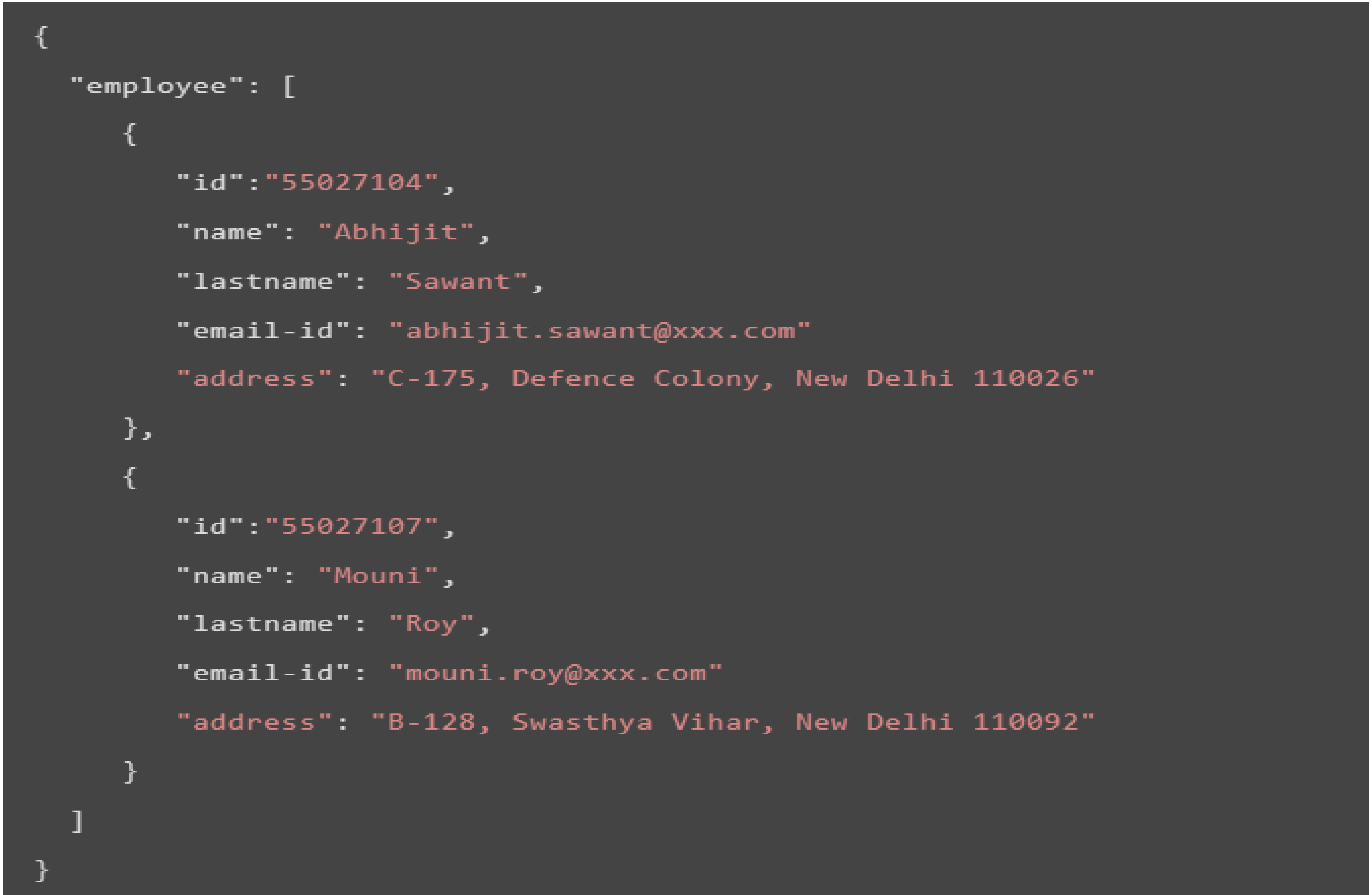
NULL

# Exercice : Convert this JSON file to XML



```json
{
    "edt": {
        "annee": 2003,
        "liste_salles": {
            "salle": {
                "idsalle": "A3",
                "capacite": "250",
                "type_salle": "amphi"
            }
        },
        "liste_profs": {
            "prof": {
                "idprof": 1233,
                "nom": "",
                "prenom": "",
                "dpt": "",
                "tel": ""
            }
        }
    }
}
```

# Exercice :
# JSON to XML : convert it to XML !



```
{
  "employee": [
     {
        "id":"55027104",
        "name": "Abhijit",
        "lastname": "Sawant",
        "email-id": "abhijit.sawant@xxx.com"
        "address": "C-175, Defence Colony, New Delhi 110026"
     },
     {
        "id":"55027107",
        "name": "Mouni",
        "lastname": "Roy",
        "email-id": "mouni.roy@xxx.com"
        "address": "B-128, Swasthya Vihar, New Delhi 110092"
     }
  ]
}
```

- What about CSV,  YAML, PARQUET ?