Introduction to Cloud-Native Web Applications with Python

Xiaozhou Li, Software Engineer, Ultimaker

About me

Me & 3D printing: https://ultimaker.com/careers/xiaozhou-li

Email: x.li@ultimaker.com / mail@xzli.me
GitHub: https://github.com/xli12
LinkedIn: https://www.linkedin.com/in/xiaozhouli/

- Xiaozhou Li
  - Software Engineer @ Ultimaker
  - Full-stack developer (80% backend/20% frontend)
  - Django Girls volunteer since 2019
  - Live in 🇳🇱 since 2017

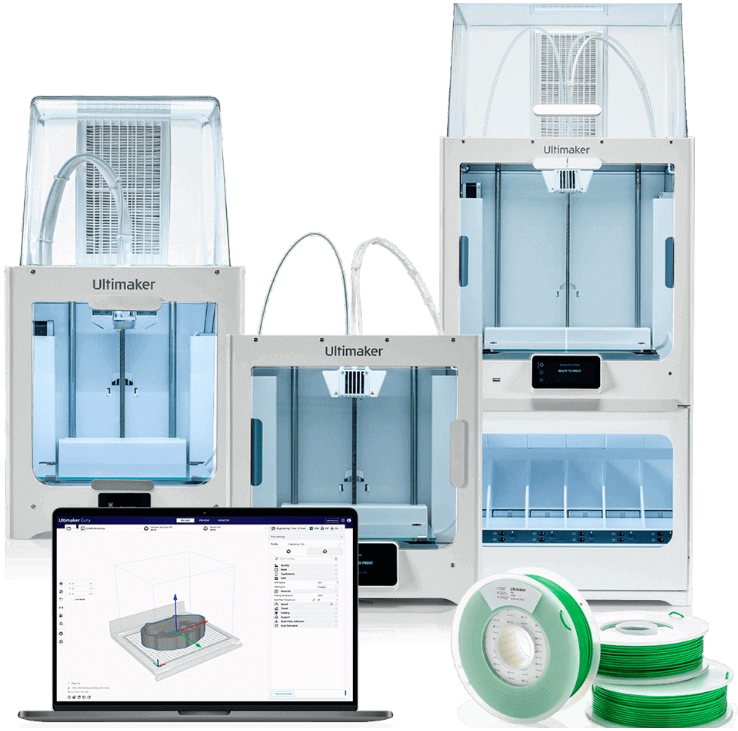
  - 🇩🇰 PhD (2017), pharmaceutical sciences (computational pharmaceutical materials science)
  - 🇬🇧 MPhil (2013), computational chemistry
  - 🇨🇳 BEng (2011), applied chemistry

Cloud-Native Web Applications with Python

Outline

Introduction	Team Stardust (the “cloud” team ☁) & Ultimaker
Cloud	What is “cloud” & a brief history about cloud platforms
Web application	The components of a simple web application
Cloud platform services	Products that are available through cloud platforms
Summary	Web application + cloud platform

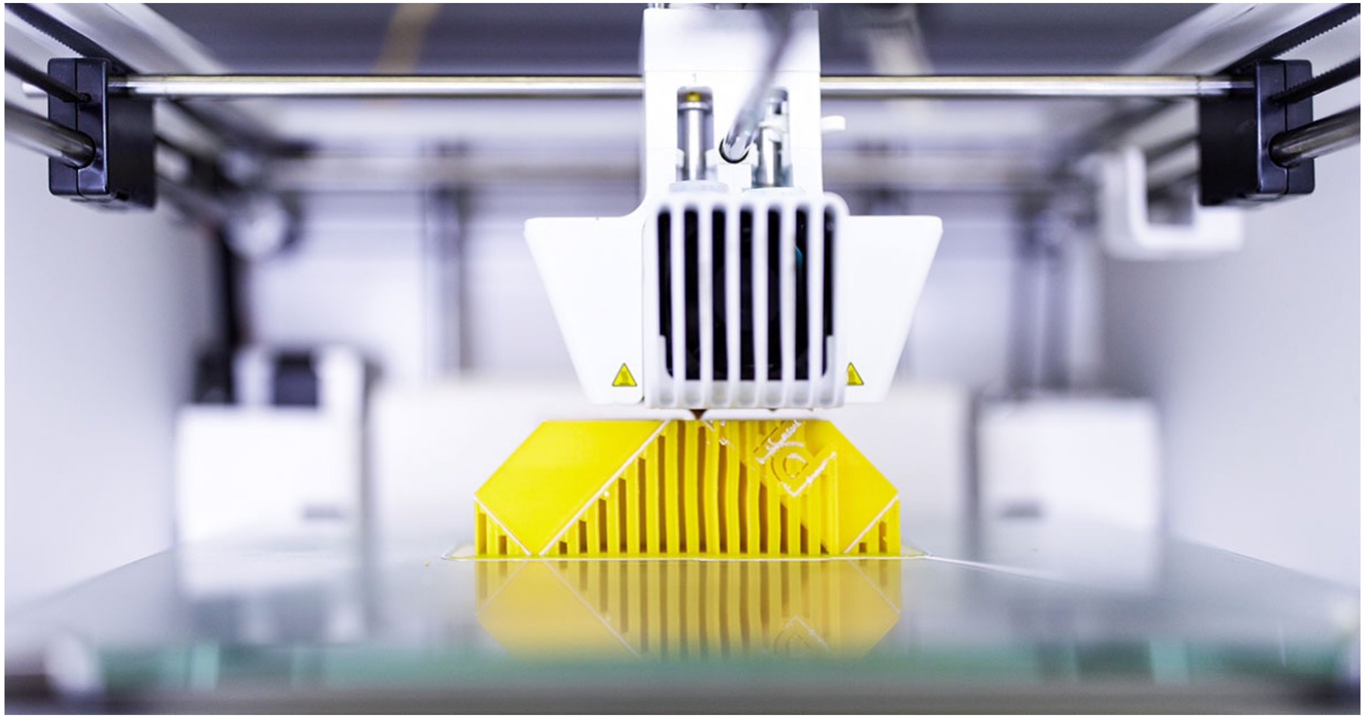
Introduction



- Ultimaker
  - Deliver a 3D printing ecosystem
  - 3D printers (FDM) Software (Cura, Digital Factory etc.)
  - Materials
  - Applications (product development, manufacturing, education)

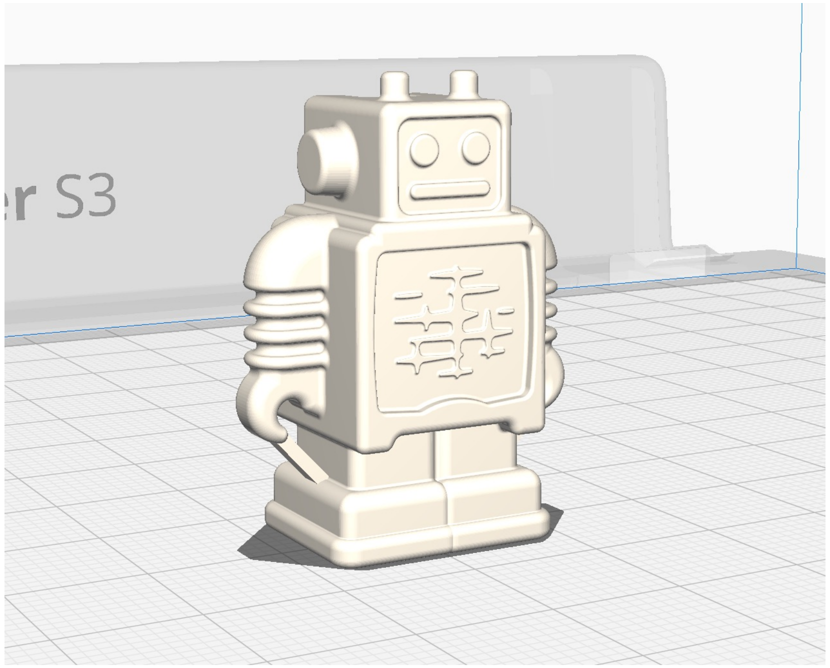
Introduction

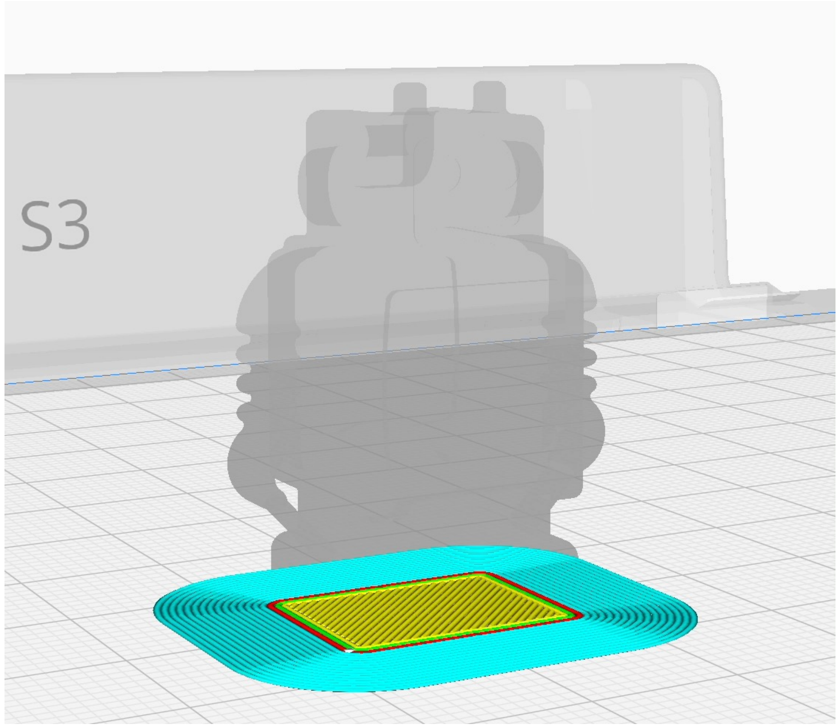
3D printing



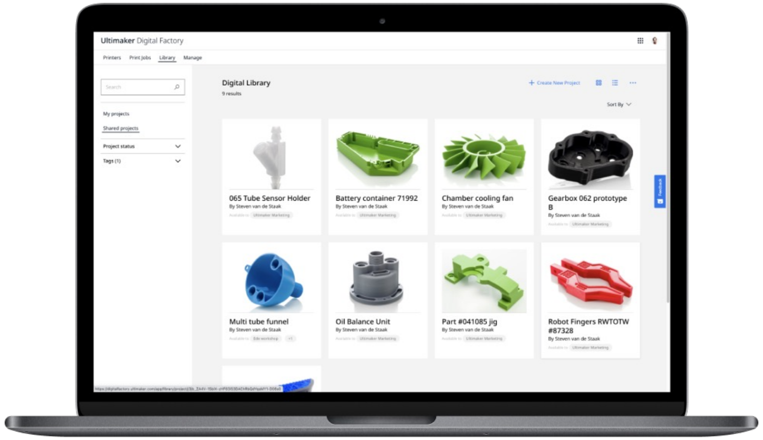
Introduction

Slice





Introduction



Digital Library

9 results

065 Tube Sensor Holder
By Steven van de Staak
Available to Ultimaker Marketing

Battery container 71992
By Steven van de Staak
Available to Ultimaker Marketing

Chamber cooling fan
By Steven van de Staak
Available to Ultimaker Marketing

Gearbox 062 prototype B
By Steven van de Staak
Available to Ultimaker Marketing

Multi tube funnel
By Steven van de Staak
Available to Ede workshop +1

Oil Balance Unit
By Steven van de Staak
Available to Ultimaker Marketing

Part #041085 jig
By Steven van de Staak
Available to Ultimaker Marketing

Robot Fingers RWTOTW #87328
By Steven van de Staak
Available to Ultimaker Marketing

- Team Stardust (☁️)
  - Cloud-Native Web Applications
  - Account
  - Digital Factory
    - Printer & Print File Management
  - Marketplace
    - Plugins & Material Profiles

What is Cloud Computing?

Simply put, cloud computing is the delivery of computing services—including servers, storage, databases, networking, software, analytics, and intelligence—over the Internet ("the cloud") to offer faster innovation, flexible resources, and economies of scale. You typically pay only for cloud services you use, helping you lower your operating costs, run your infrastructure more efficiently, and scale as your business needs change.

--Microsoft Azure "What is Cloud Computing?"

Cloud, still sounds vague? 🤔

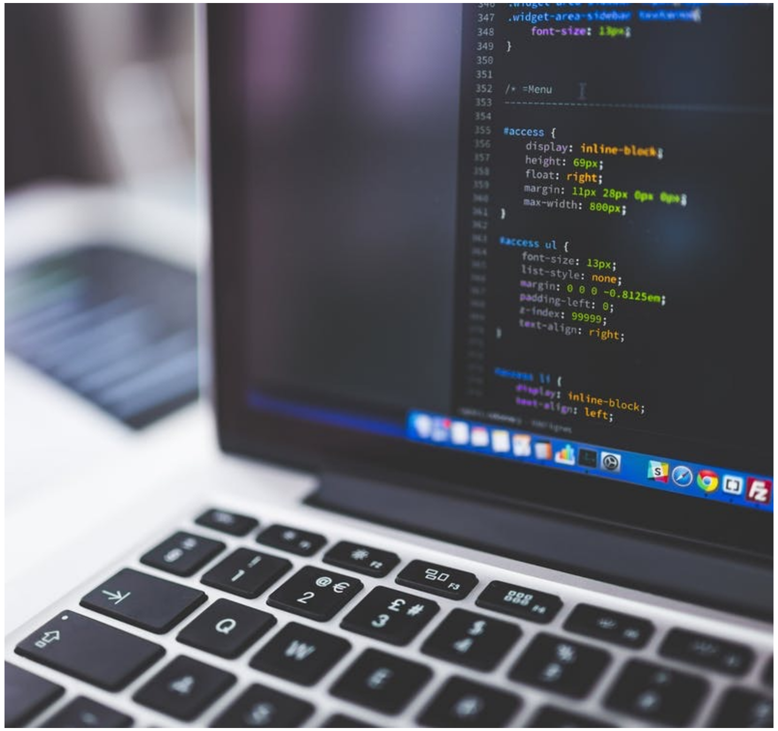
Build a website in 2000

1. Plan
   - What do we want to build?
   - Milestones for each phase?
   - Who are the targeted users?
   - How many users do we expect?

Build a website in 2000

1. Plan
2. Submit a budget plan
   - Servers
   - Operation staff
   - Maintenance cost
   - On-premises data centre
   - Peak hour/season solutions

Build a website in 2000



1. Plan
2. Submit a budget plan
3. Implementation, operation, maintenance & “what if”s
   - Number of customers much lower/higher than expected? (capacity planning) – super annoying!
   - Are the features in the app still something that the customers *exactly* want?

Built a website in 2000

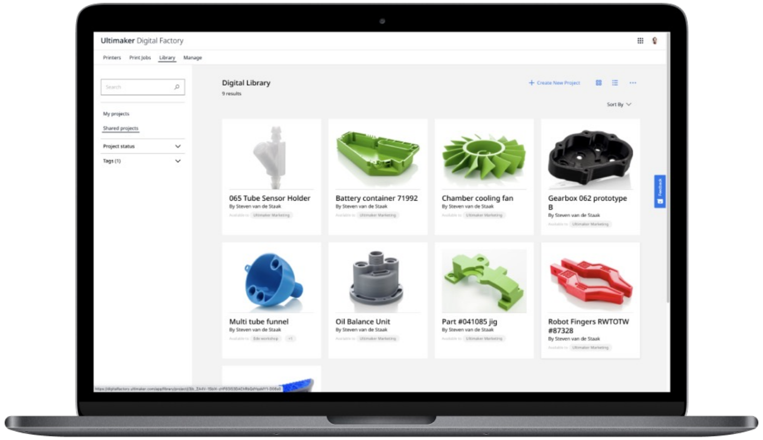
1. Plan
2. Submit a budget plan
3. Implementation & Operation

In general: Overheads!

- Nothing to do with customers or deliver the actual products

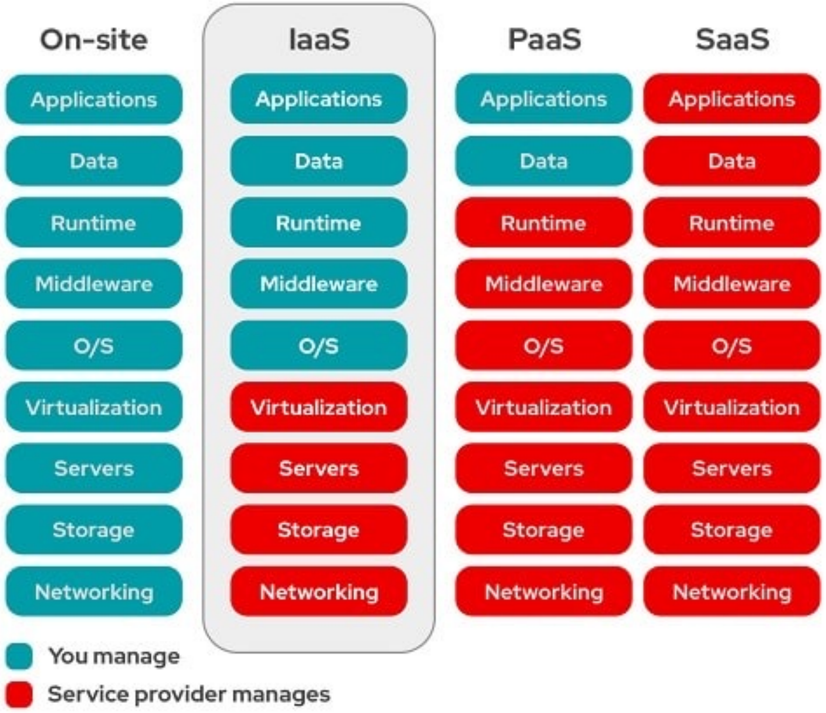
Low-budget, easy-to-start option?

Digital Factory: A cloud-native web application



- Cloud-native
  - Utilise cloud computing to "build and run scalable applications in modern, dynamic environments”
- Rent services from cloud providers
  - Infrastructure as a service (IaaS)
  - Platform as a service (PaaS)
  - Software as a service (SaaS)
  - Pay-as-you-go
  - Autoscale (scale up, scale out)

Iaas, PaaS, SaaS



On-site	IaaS	PaaS	SaaS
🟦 Applications	🟦 Applications	🟦 Applications	🟥 Applications
🟦 Data	🟦 Data	🟦 Data	🟥 Data
🟦 Runtime	🟦 Runtime	🟥 Runtime	🟥 Runtime
🟦 Middleware	🟦 Middleware	🟥 Middleware	🟥 Middleware
🟦 O/S	🟦 O/S	🟥 O/S	🟥 O/S
🟦 Virtualization	🟥 Virtualization	🟥 Virtualization	🟥 Virtualization
🟦 Servers	🟥 Servers	🟥 Servers	🟥 Servers
🟦 Storage	🟥 Storage	🟥 Storage	🟥 Storage
🟦 Networking	🟥 Networking	🟥 Networking	🟥 Networking

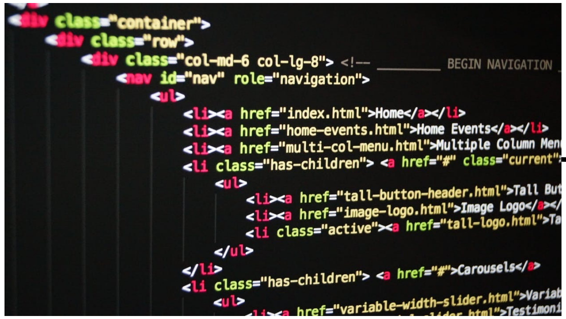
🟦 You manage

🟥 Service provider manages

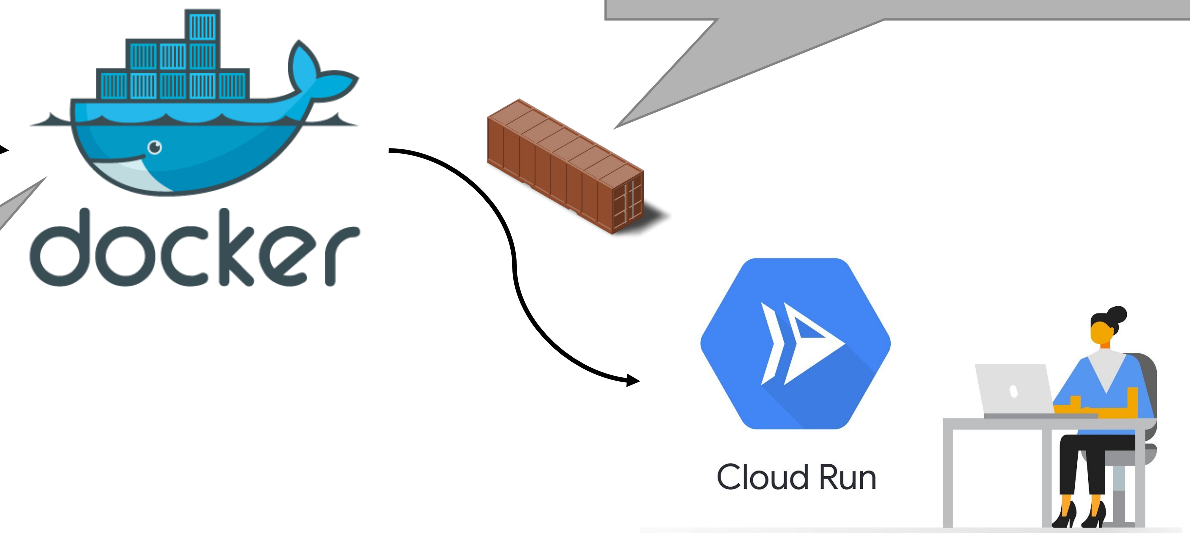
Source: https://www.redhat.com/en/topics/cloud-computing/what-is-iaas

Shipping the code

We’ll do the small exercise later!



```
<div class="container">
    <div class="row">
        <div class="col-md-6 col-lg-8"> <!-- _________ BEGIN NAVIGATION
            <nav id="nav" role="navigation">
                <ul>
                    <li><a href="index.html">Home</a></li>
                    <li><a href="home-events.html">Home Events</a></li>
                    <li><a href="multi-col-menu.html">Multiple Column Men
                    <li class="has-children"> <a href="#" class="current"
                        <ul>
                            <li><a href="tall-button-header.html">Tall But
                            <li><a href="image-logo.html">Image Logo</a></
                            <li class="active"><a href="tall-logo.html">Ta
                        </ul>
                    </li>
                    <li class="has-children"> <a href="#">Carousels</a>
                        <ul>
                            <li><a href="variable-width-slider.html">Variab
                                              slider.html">Testimoni
```

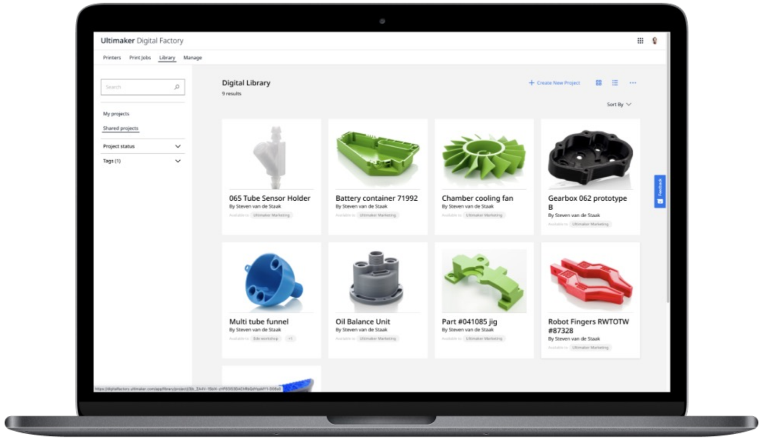


Containers are lightweight packages of your application code together with dependencies such as specific versions of programming language runtimes and libraries required to run your software services.

Docker: a set of platform as a service products that use OS-level virtualization to deliver software in packages called containers.

More about Docker and containerisation: https://www.docker.com/resources/what-container/

Digital Factory: A cloud-native web application



- Digital Factory up and running!
- Digital Library: managing print jobs and print files.
- https://digitalfactory.ultimaker.com/app/library?page=1&shared=true

Summary: Cloud Computing

- On-demand availability of computer system resources without direct active management by the user
- Three major Cloud providers:
  - Amazon Web Services (AWS)
  - Microsoft Azure
  - Google Cloud Platform (GCP)

Still have questions? Of course!

- All we have done was delivering a Docker image to the Cloud platform… and that was it?
  - No, that’s not the case of course!
- Okay, what else?
  - Databases
  - Message queues
  - Logging
  - Alerting
  - Load balancer
  - Code repositories
  - Continuous integration/continuous delivery (CI/CD)
  - And more… 🤯

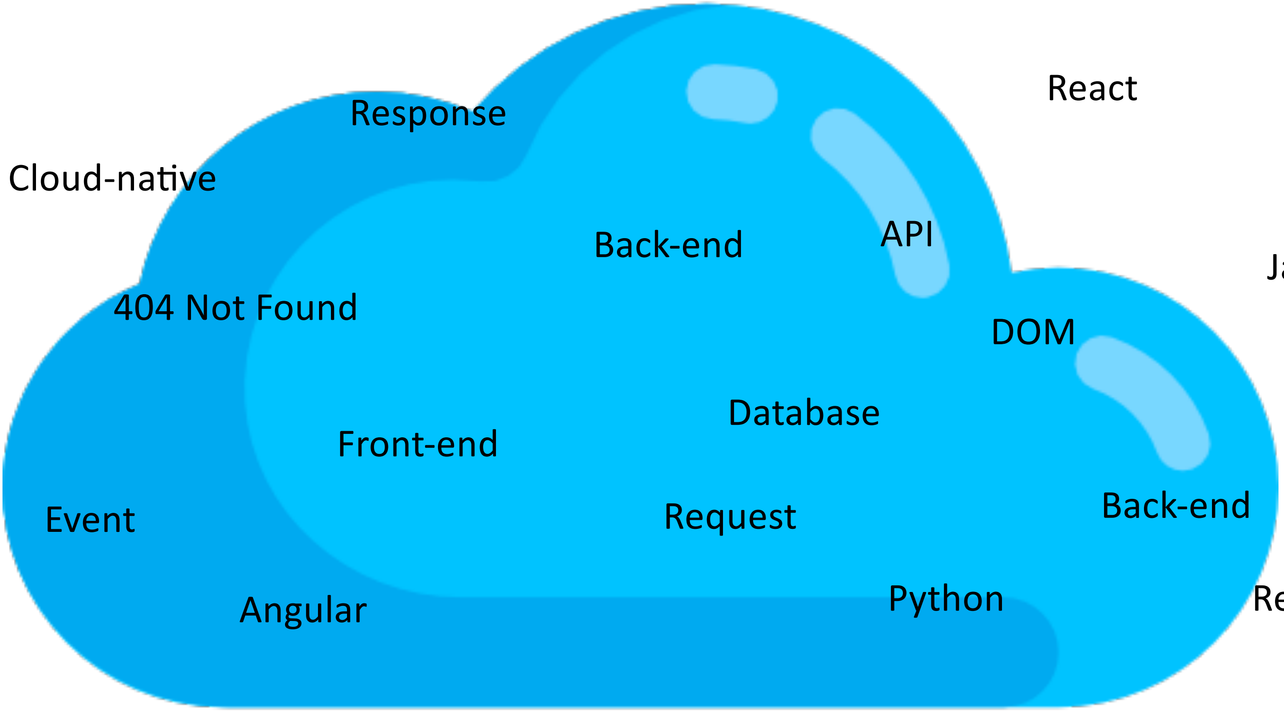
Direction?

1. What is cloud computing? ✅
2. What is a web application? 🐍
3. Combine cloud computing with a web application ☁️

The boring definition of a web application

A web application (or web app) is application software that is stored on a remote server and delivered over the internet through a web interface, usually a web browser.

Welcome to the jargon cloud of web applications



Bootstrap

React

Response

Cloud-native

Back-end

API

Java

404 Not Found

DOM

JavaScript

Database

Front-end

Back-end

Event

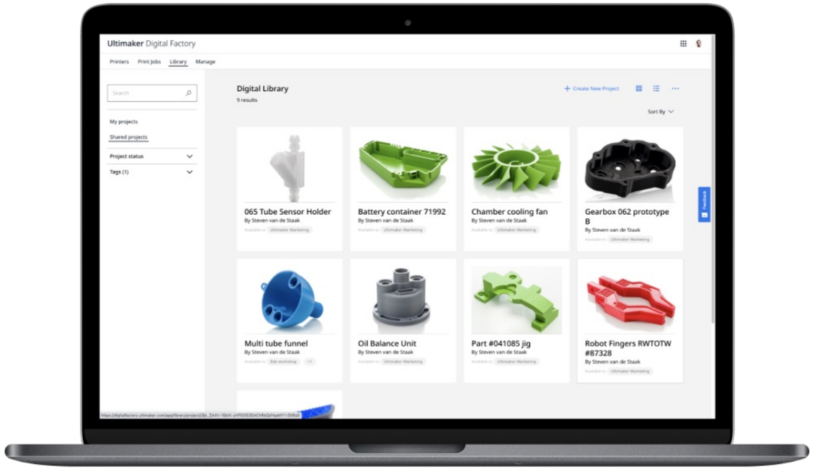
Request

Python

Responsive design

Angular

Use an example to understand a web application



Digital Library

9 results

065 Tube Sensor Holder
By Steven van de Staak
Available to Ultimaker Marketing

Battery container 71992
By Steven van de Staak
Available to Ultimaker Marketing

Chamber cooling fan
By Steven van de Staak
Available to Ultimaker Marketing

Gearbox 062 prototype B
By Steven van de Staak
Available to Ultimaker Marketing

Multi tube funnel
By Steven van de Staak
Available to Ede workshop +1

Oil Balance Unit
By Steven van de Staak
Available to Ultimaker Marketing

Part #041085 jig
By Steven van de Staak
Available to Ultimaker Marketing

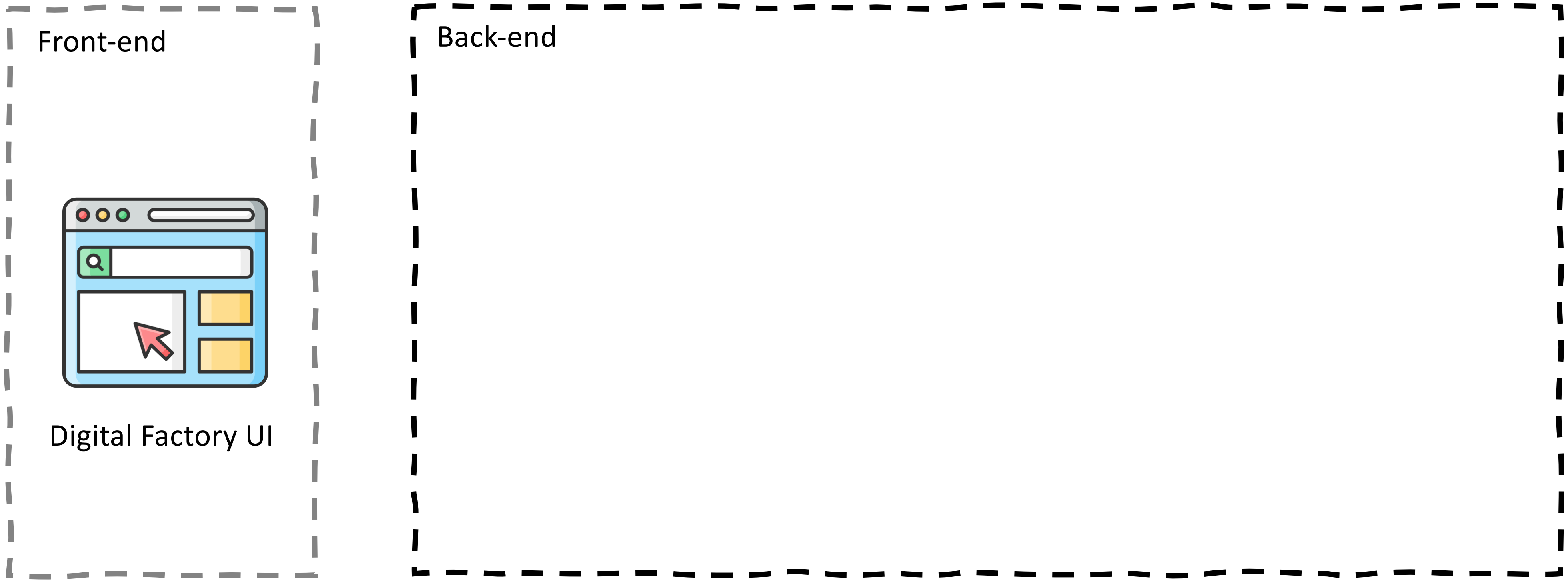
Robot Fingers RWTOTW #87328
By Steven van de Staak
Available to Ultimaker Marketing

digitalfactory.ultimaker.com

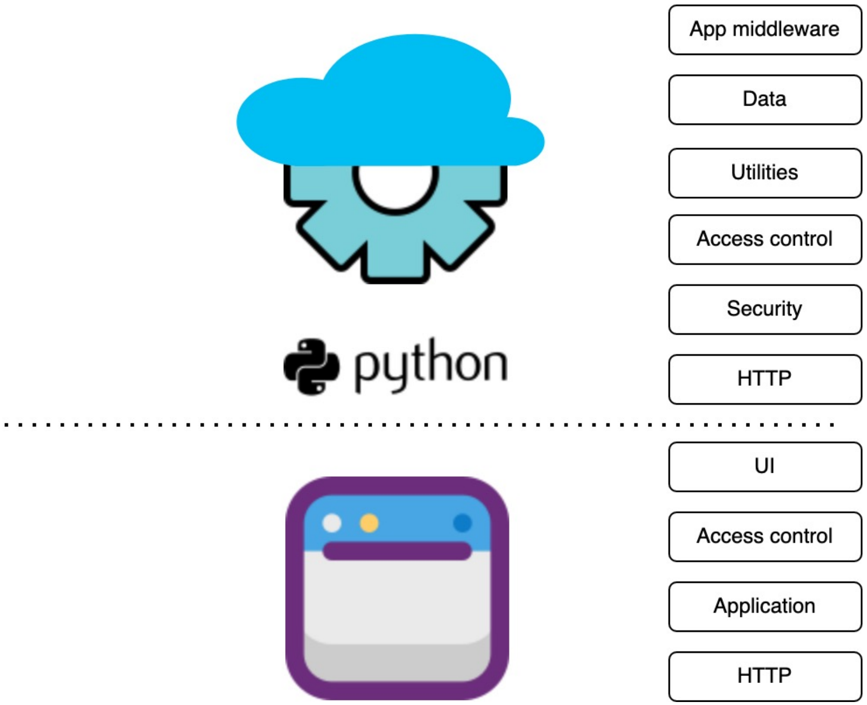
Front-end

- An interface that users directly interact with
- Web application: Browsers
  - HTML (contents)
  - CSS (styling)
  - JavaScript (Effects)

Front-end & back-end: Request



Re-inventing the wheels?



App middleware

Data

Utilities

Access control

Security

HTTP

UI

Access control

Application

HTTP

User registration

Routes:
/projects
/printers

Password validation and encryption

The DRY principle

Don't repeat yourself

- Don’t repeat yourself (DRY) is a principle of software development that aims at reducing the repetition of pattern and code duplication in favour of abstraction and avoiding redundancy.

Web framework

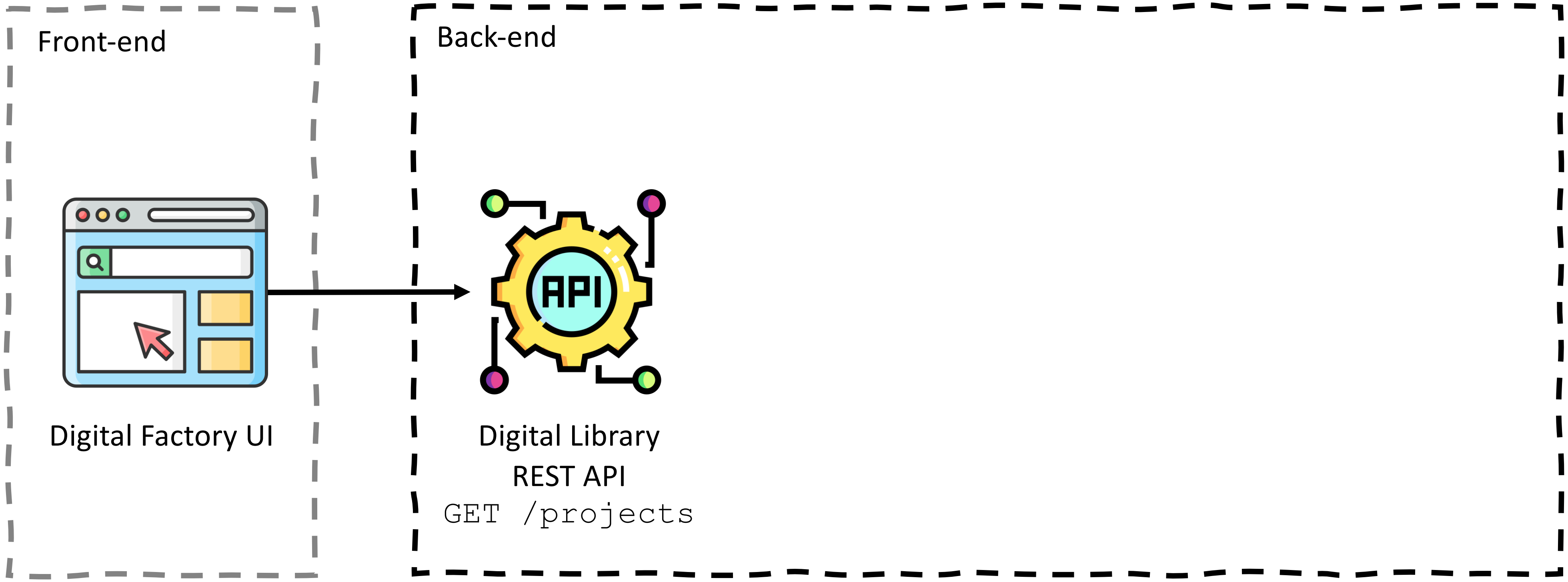
Definition

- Software framework designed for developing web applications
  - Web services
  - Web resources
  - Web APIs

Examples of web frameworks

- Server-side
  - Java: Spring, Grails etc.
  - Ruby: Ruby on Rails, Sinatra etc.
  - Python: Django, Flask, Tornado, FastAPI etc.
- Client-side (single-page application)
  - React
  - Vue.js
  - Angular

Front-end & back-end: Request



Front-end

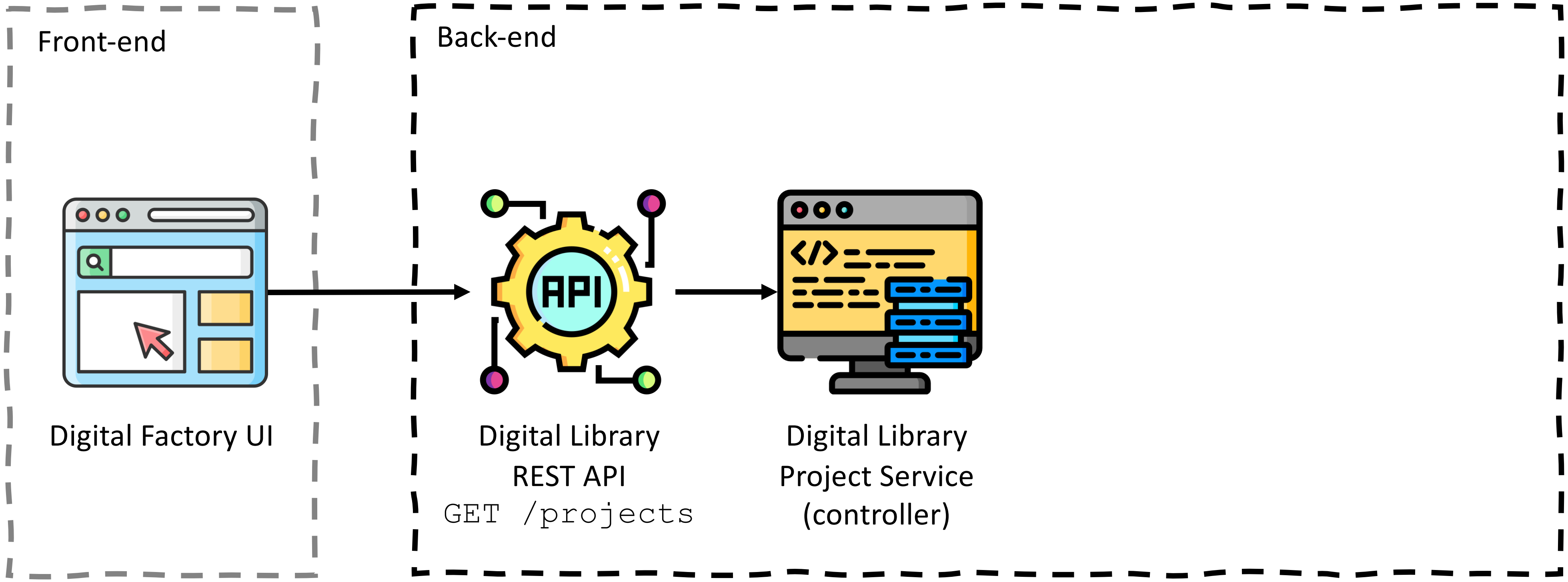
Back-end

Digital Factory UI

Digital Library
REST API

```
GET /projects
```

Front-end & back-end: Request



Front-end

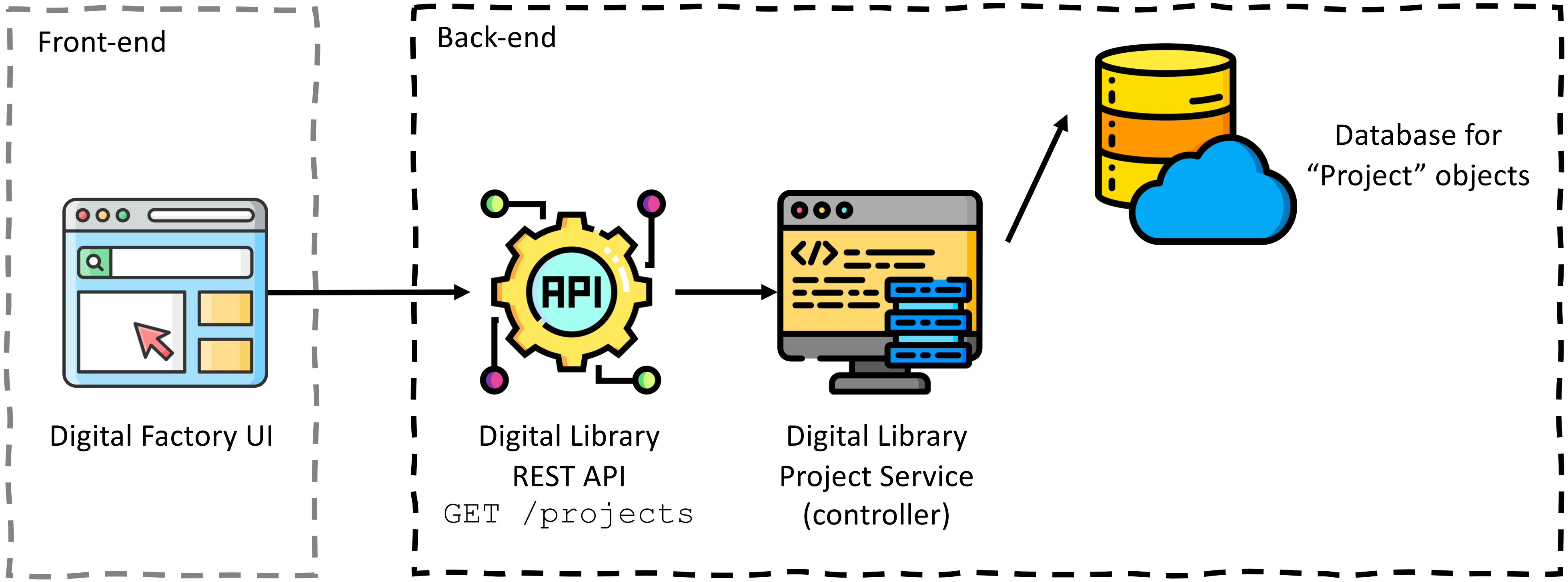
Back-end

Digital Factory UI

Digital Library
REST API
`GET /projects`

Digital Library
Project Service
(controller)

Front-end & back-end: Request



Front-end

Digital Factory UI

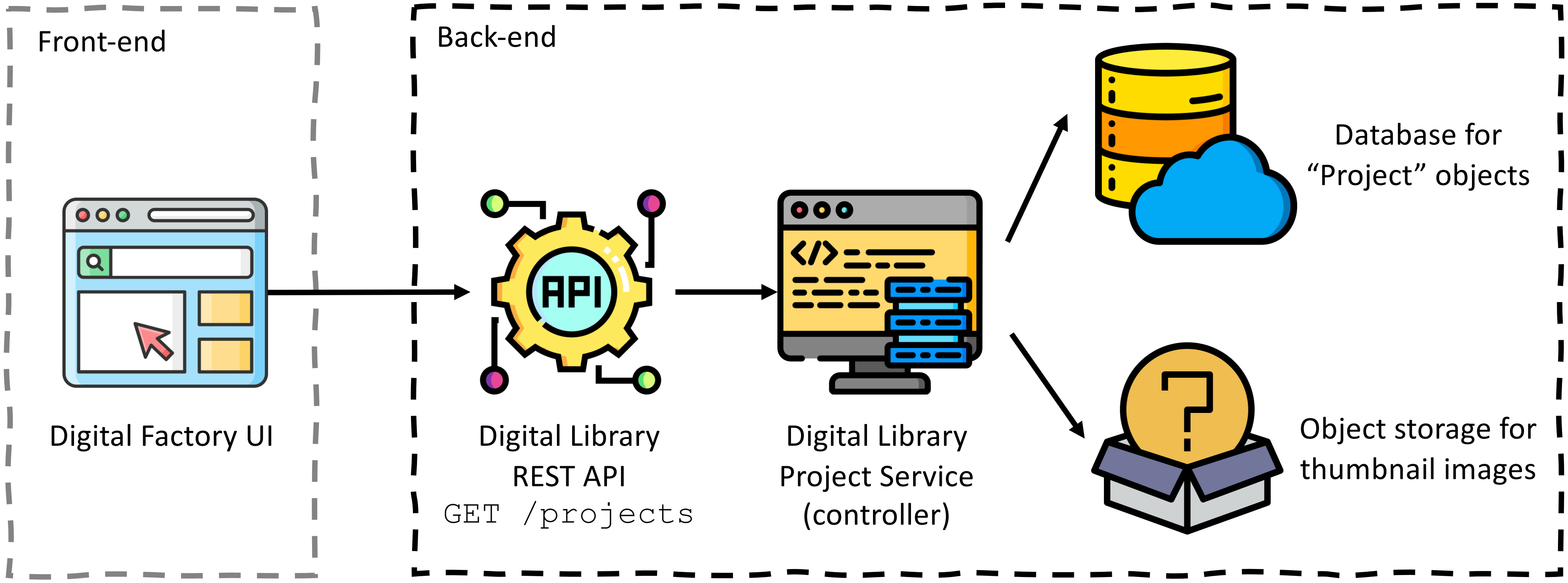
Back-end

Digital Library
REST API
`GET /projects`

Digital Library
Project Service
(controller)

Database for
“Project” objects

Front-end & back-end: Request



Front-end

Digital Factory UI

Back-end

Digital Library REST API

`GET /projects`

Digital Library Project Service (controller)

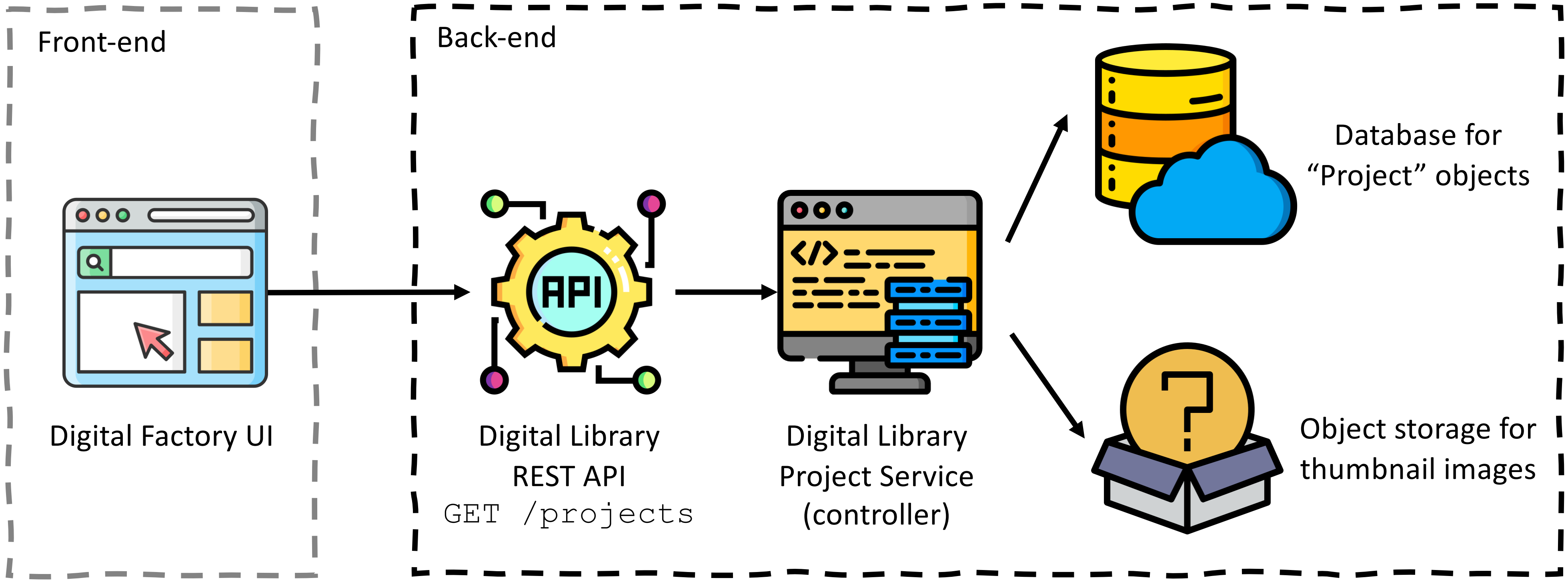
Database for “Project” objects

Object storage for thumbnail images

Cloud object storage

- Each object has a unique name
- The object name is stored in the database
- Based on the unique object name, we find it in the Cloud Storage

A simple cloud-native web application



Front-end

Digital Factory UI

Back-end

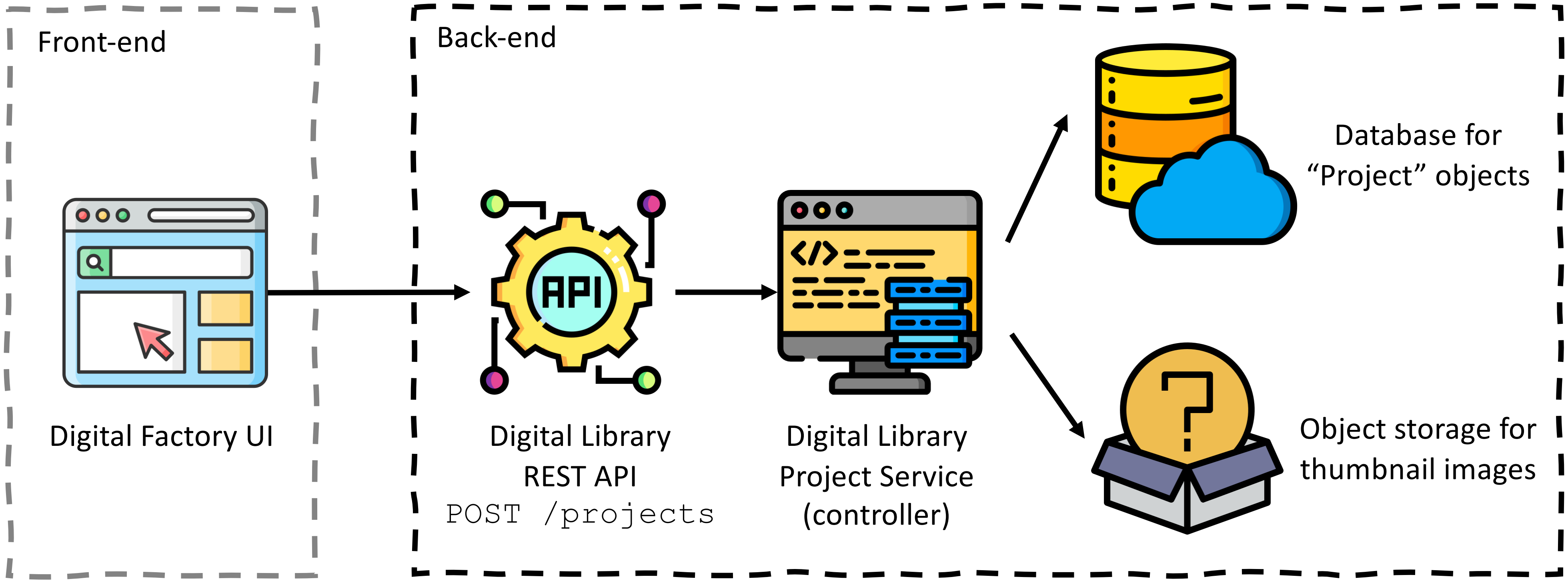
Digital Library
REST API
`GET /projects`

Digital Library
Project Service
(controller)

Database for
“Project” objects

Object storage for
thumbnail images

A simple cloud-native web application



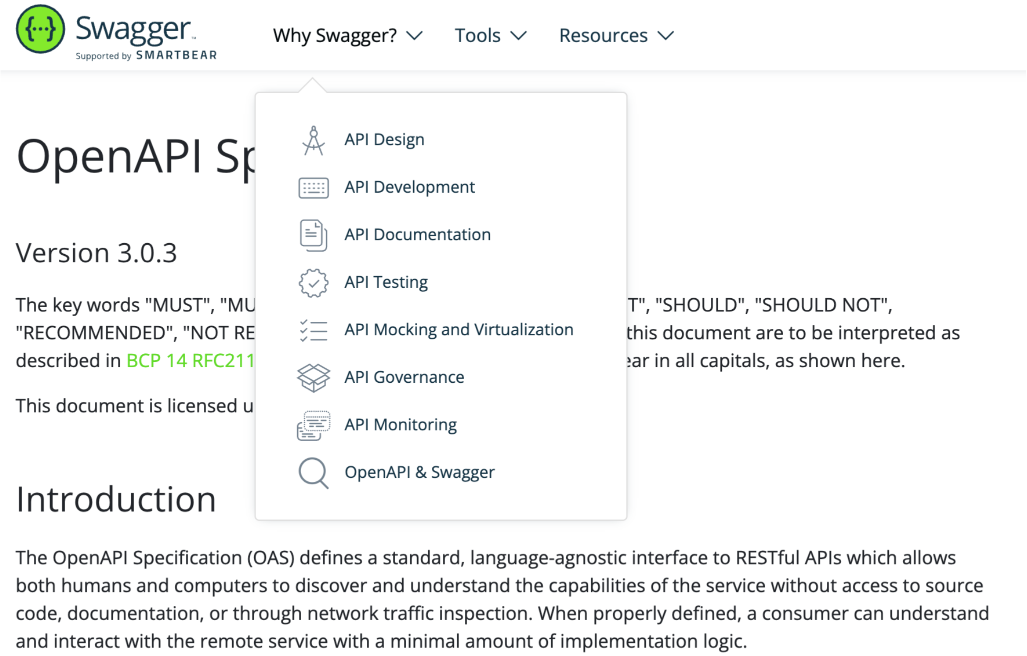
Front-end
Back-end
Digital Factory UI
Digital Library REST API
POST /projects
Digital Library Project Service (controller)
Database for “Project” objects
Object storage for thumbnail images
API

Two questions

1. Why the same path, but different methods?
2. What are the possible methods?

Same path, different methods

The OpenAPI Specification (OAS) defines a standard, language-agnostic interface to RESTful APIs which allows both humans and computers to discover and understand the capabilities of the service without access to source code, documentation, or through network traffic inspection.



OpenAPI Sp

Version 3.0.3

The key words "MUST", "MU T", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RE this document are to be interpreted as described in BCP 14 RFC211 ear in all capitals, as shown here.

This document is licensed u

Introduction

The OpenAPI Specification (OAS) defines a standard, language-agnostic interface to RESTful APIs which allows both humans and computers to discover and understand the capabilities of the service without access to source code, documentation, or through network traffic inspection. When properly defined, a consumer can understand and interact with the remote service with a minimal amount of implementation logic.

1. OpenAPI specification

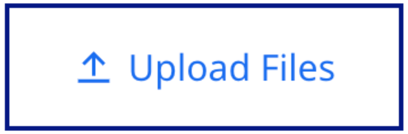
Possible methods

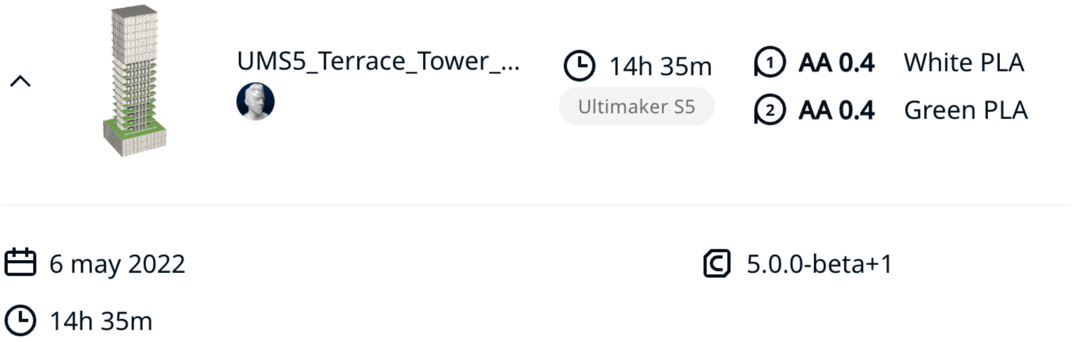
- HTTP: Hypertext Transfer protocol
  - HTTP defines a set of **request methods** to indicate the desired action to be performed for a given resource.

- GET
- HEAD
- POST
- PUT
- DELETE
- CONNECT
- OPTIONS
- TRACE
- PATCH

Source: https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods

File upload





UMS5_Terrace_Tower_...

14h 35m

Ultimaker S5

1 **AA 0.4** White PLA

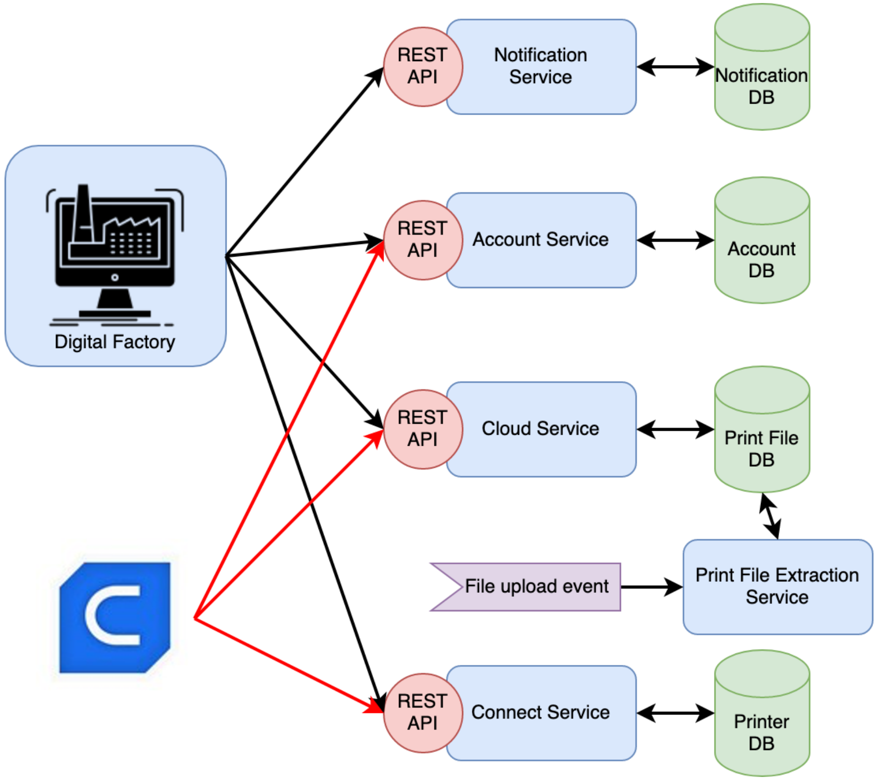
2 **AA 0.4** Green PLA

6 may 2022

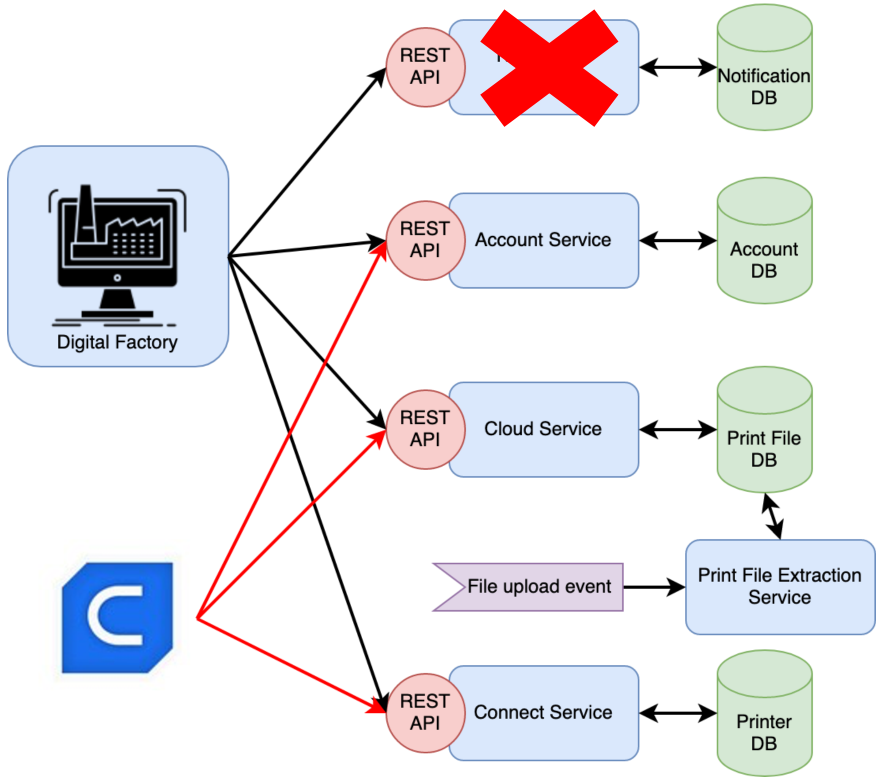
5.0.0-beta+1

14h 35m

Microservices & Event-driven architecture



Microservices & Event-driven architecture



Digital Factory

REST API

Notification DB

REST API

Account Service

Account DB

REST API

Cloud Service

Print File DB

File upload event

Print File Extraction Service

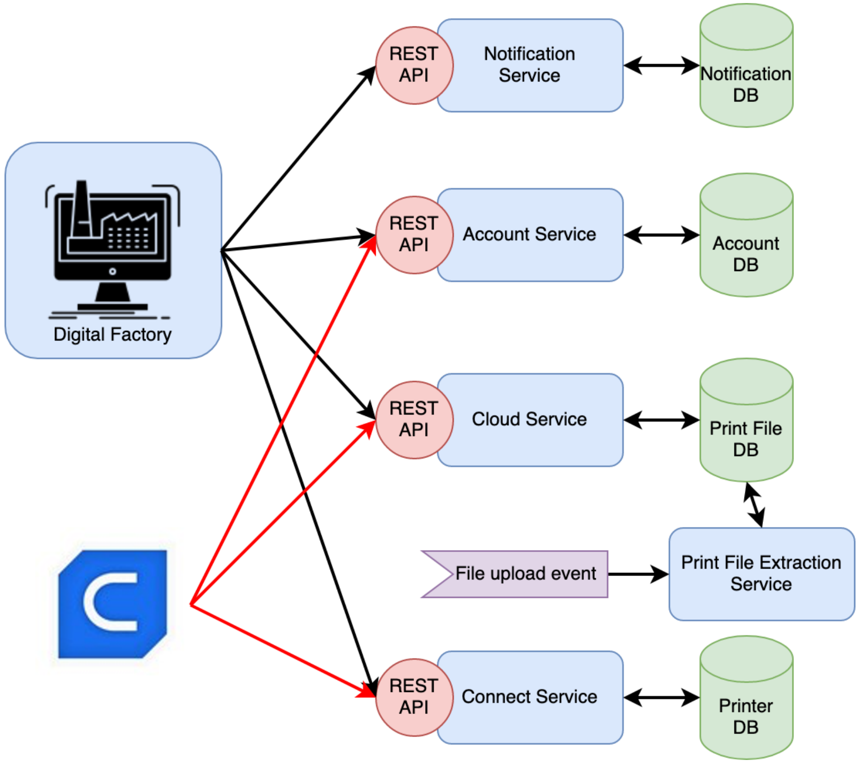
REST API

Connect Service

Printer DB

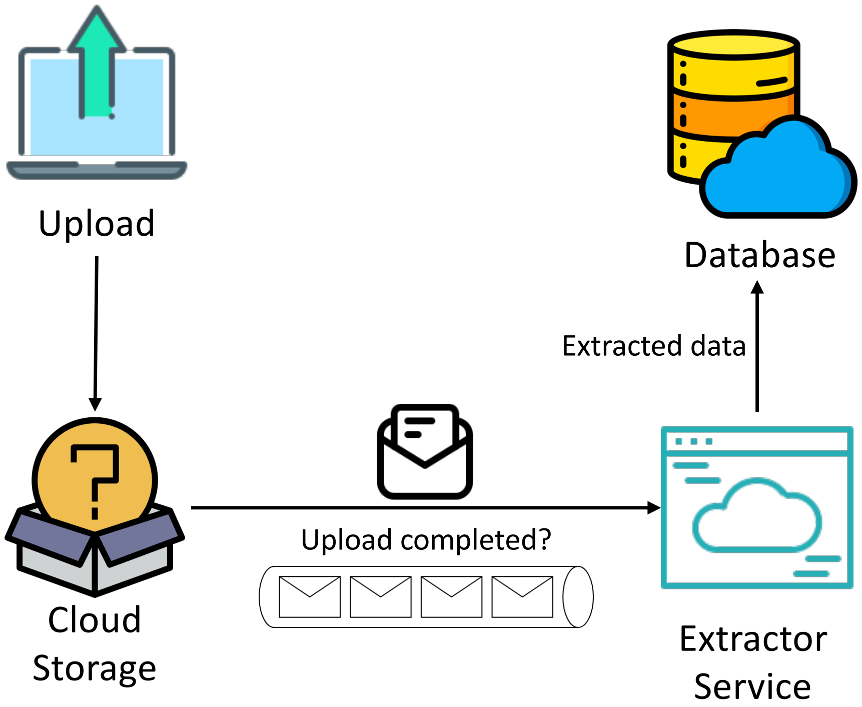
Microservices

- A microservices architecture is a type of application architecture where the application is developed as a collection of services. It provides the framework to develop, deploy, and maintain microservices architecture diagrams and services independently.



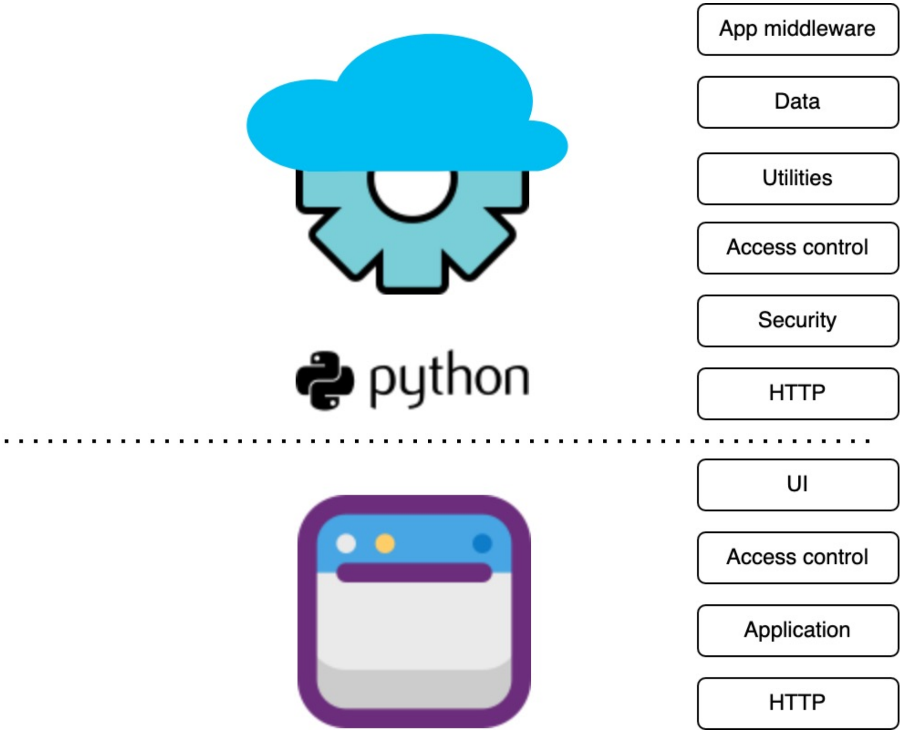
https://cloud.google.com/learn/what-is-microservices-architecture

Event-driven architecture & message queue



- Upload a file to Cloud Storage
- Once the upload is completed, Cloud Storage sends a message to the message queue
- The Extractor service picks up the message from the queue, and process the message (extract the data from the uploaded file)
- The Extrator service stores the extracted data to the database

Summary



App middleware

Data

Utilities

Access control

Security

HTTP

UI

Access control

Application

HTTP

- Web frameworks
  - Server-side (front-end)
  - Client-side (back-end)
- Database & Object storage
- OpenAPI specification
- HTTP request methods
- Microservices
- Event-driven architecture

Pros & Cons: Cloud-native web applications

Pros

- Cost efficient
- Scalability
- Automation and flexibility
- Faster release
- No overhead burdens

Cons

- Management can be complex
  - Container orchestration
  - Terraform configuration
- Cloud provider reliability

Demo

- Deploy a Django App to Google Cloud Run
  - Google’s managed compute platform which lets the users to run containers directly on Google’s scalable infrastructure

Homework

Given the list of possible cloud services, could you guess how the application works?

- 👩‍💻 User uploads a file to a project in Digital Library
- 📧 Emails sent to all users who have access to that project (A file has been uploaded to the project "Python", Please check it out.)
- The email sending functionality is also used by other microservices (e.g. send emails if a print job finishes)

- Google Cloud Run (Environment for containerised applications)
- Cloud storage (blob storage)
- SQL database
- Message queue

Homework

Hints

- Emails can be sent using external, 3rd party services. The developers provide all the information needed in the email, send it to the external service, and that service will sent emails out.

Homework

Ultimaker