Correct Exam

Modern software architecture in practise

Press Space for next page →

My mojo

https://www.linkedin.com/pulse/i-have-right-do-research-software-engineering-hafedh-mili/

> Parnas noted "I would never have realized the nature of the problem, unless I had been working on that project, reviewing development documents, and sitting at that lunch table". Well, minimally, I need to be able to understand the conversation at that lunch table!

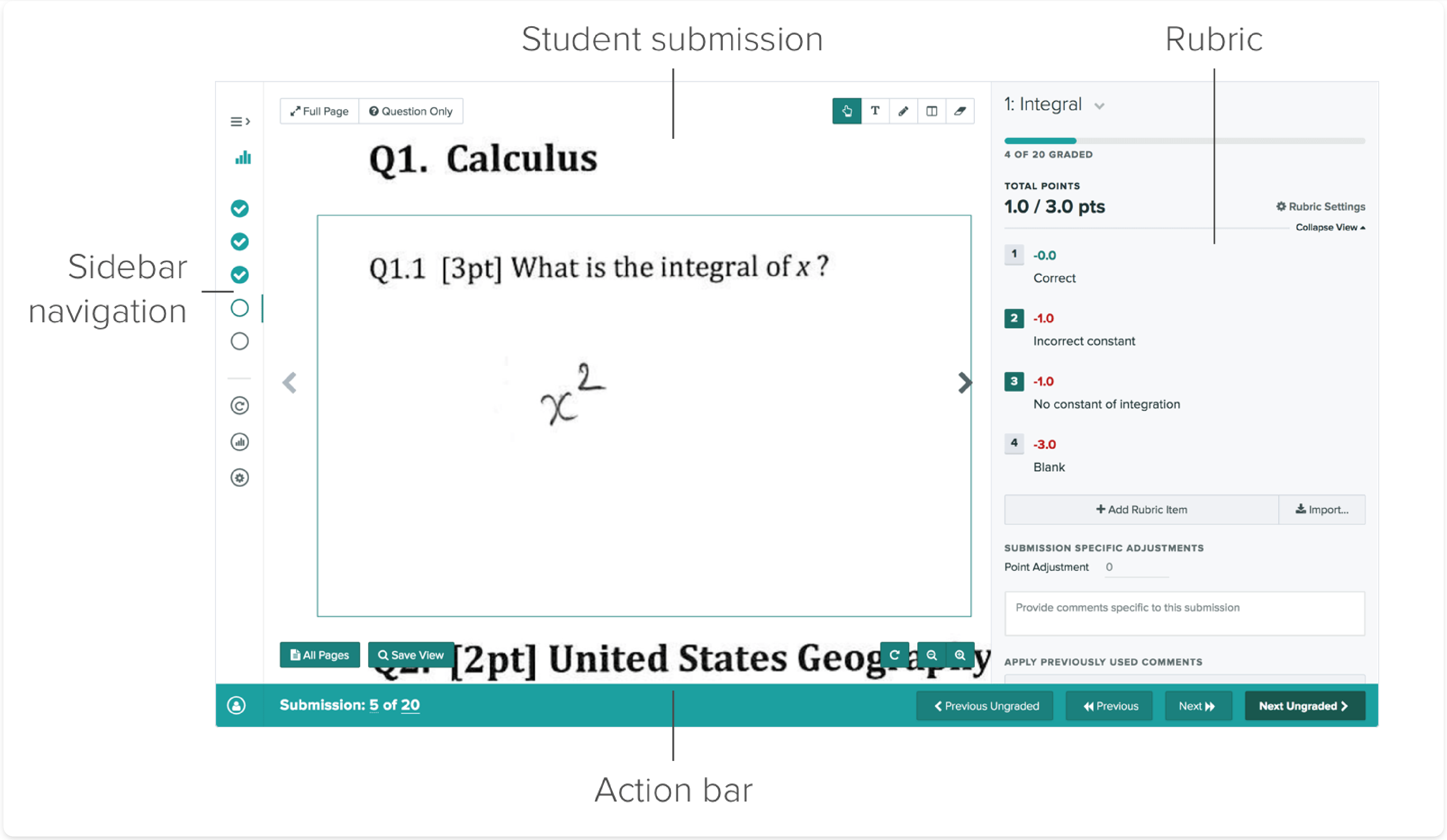
So here is a practical definition of what "understanding the conversation" means in this context: **You have no credibility to do software engineering research unless you have at least the development skills/vocabulary of your graduating bachelor students.**

That is why: I enjoy building real software, doing consultancy, working with students, ...

The project: correct exam

Highly inspired by GradeScope Solution

> Gradescope grading software allows students to receive faster and more detailed feedback on their work, and allows instructors to see detailed assignment and question analytics. It is an easy way to take submissions digitally in order to preserve the original work and allow for quick and easy viewing from anywhere.



Q1. Calculus

Q1.1 [3pt] What is the integral of x?

x^2

Q2. [2pt] United States Geography

1: Integral

4 OF 20 GRADED

TOTAL POINTS
1.0 / 3.0 pts

1	-0.0	Correct
2	-1.0	Incorrect constant
3	-1.0	No constant of integration
4	-3.0	Blank

SUBMISSION SPECIFIC ADJUSTMENTS
Point Adjustment 0
Provide comments specific to this submission
APPLY PREVIOUSLY USED COMMENTS

Submission: 5 of 20

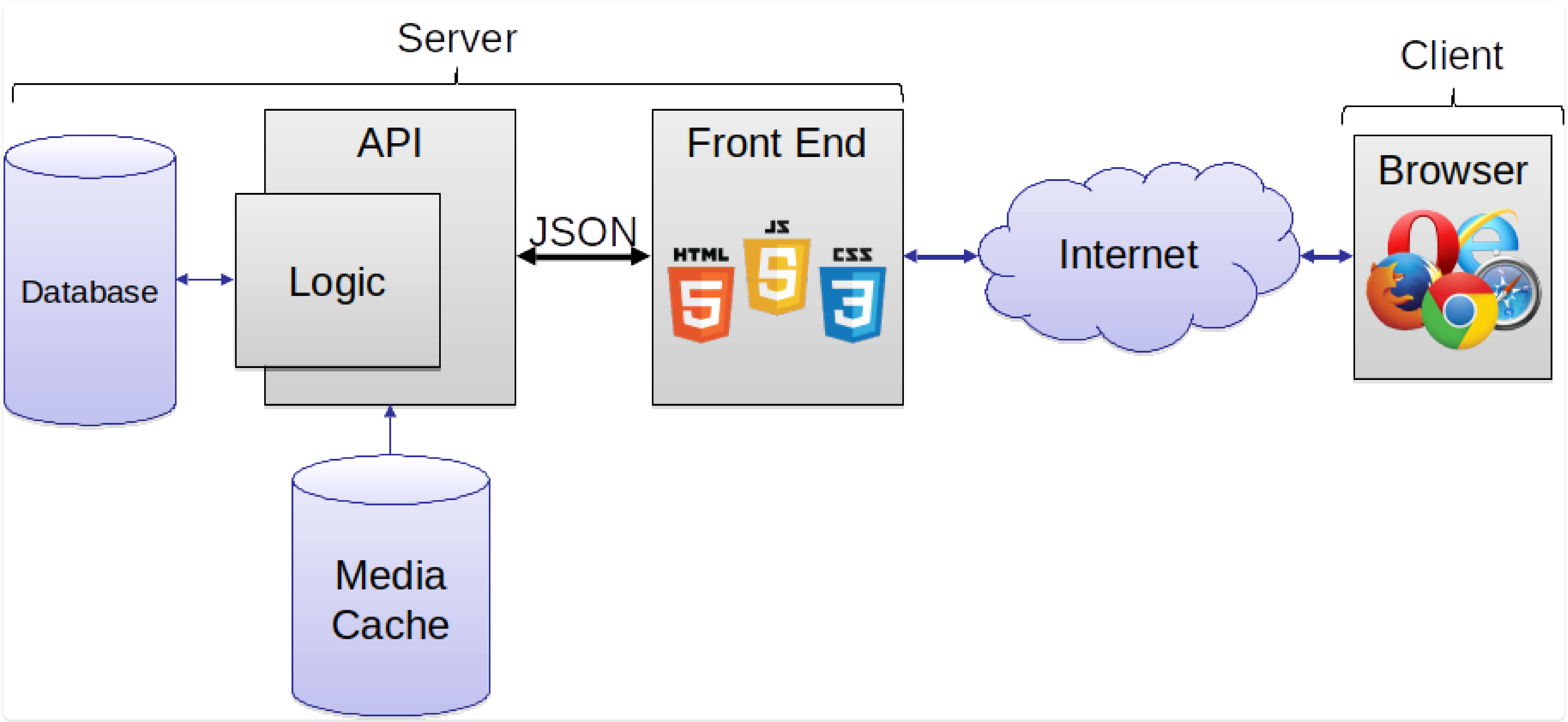
Why building that piece of software ?

- Enable to correct exams during meetings 😁
- Save $5 per student copy
- Create an open source implementation of real software with complex architecture to have a case study for
  - experiments in software engineering research
  - explaining modern software architecture to students
- Try to keep the credibility (in my vision) to do research in software engineering

The technical architecture

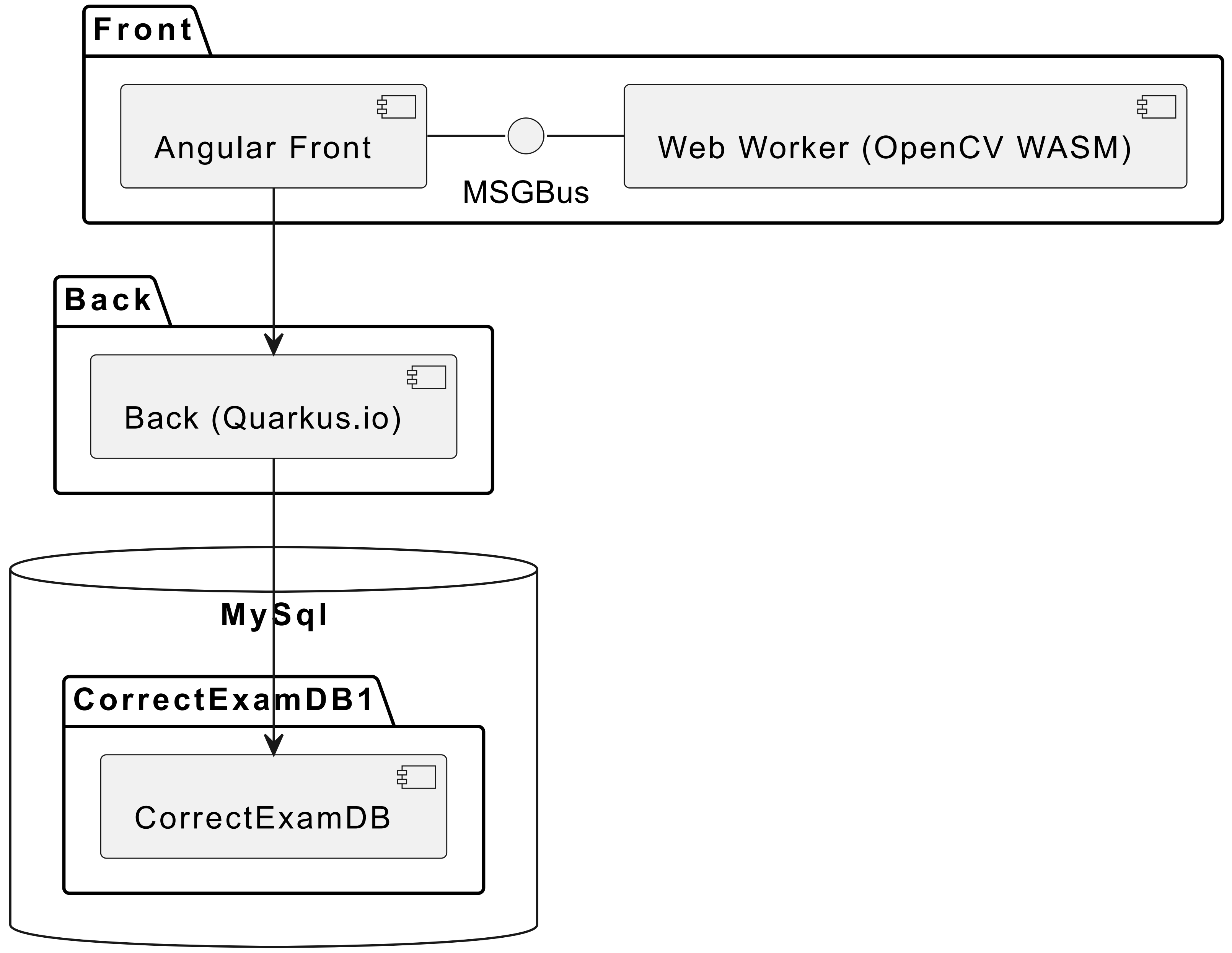
- **Quarkus** for the back (Java + native compilation through GraalVM)
- **Angular 13.3** for the front
  - **pdf.js** to play with pdf (exam, scan exam, feedback for students)
  - **fabric.js** to draw on top of a pdf
  - **opencv** in wasm within a web worker to analyse the scan
  - **tensorflow JS** with the browser for digit and letter recognition
  - …
- **Docker** and K8S to deploy the back and the monitoring layer
- Front is hosted in a CDN to follow the JamStack architecture (currently github page)
- CI/CD using **github action**, **dockerhub webhook**, and **gowebhook**
- Pluggable architecture based on micro-services to integrate question type automatic correction (let developper use their own programming language to provides solutions that automatically correct some questions)

Architecture overview

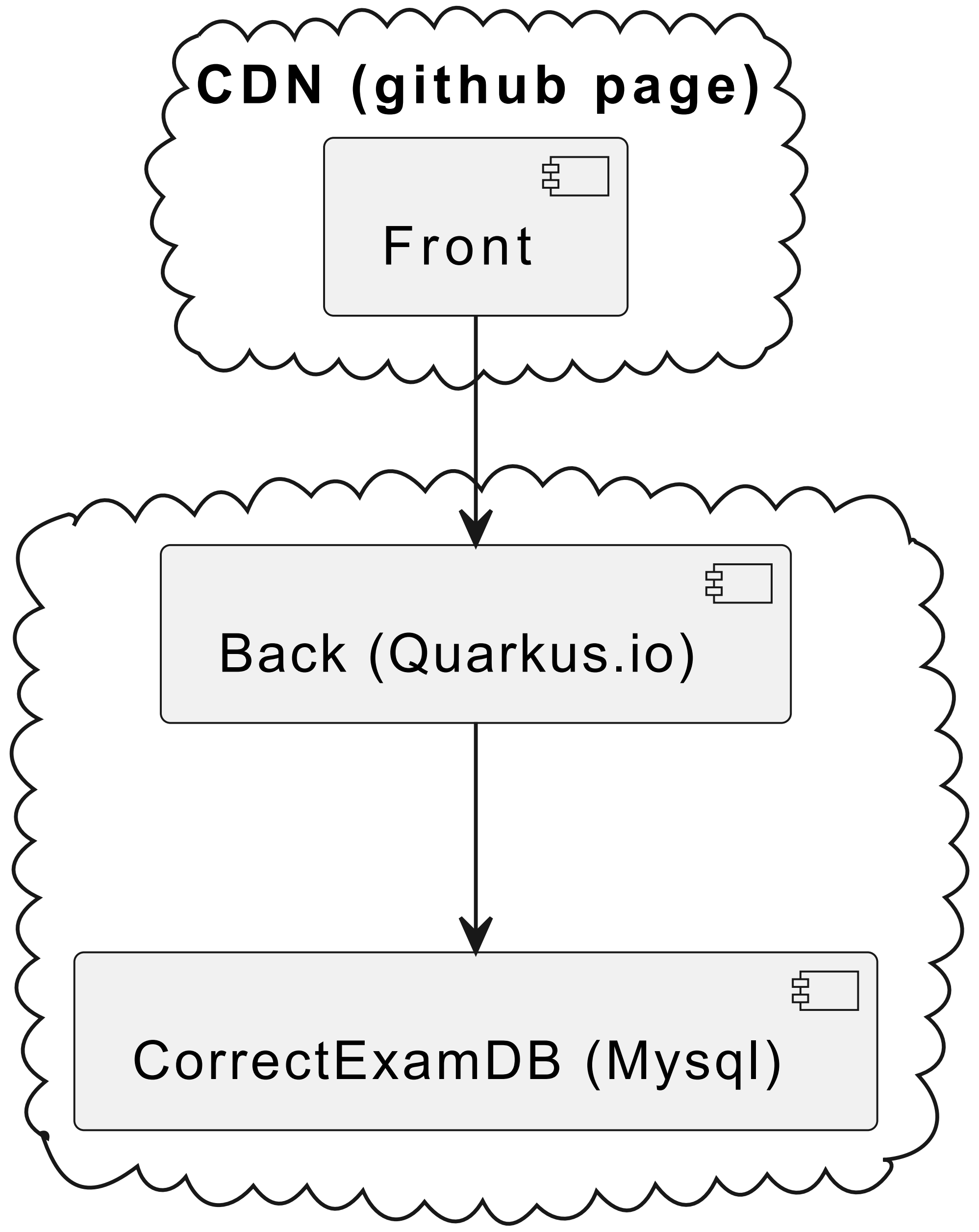


Diagrams

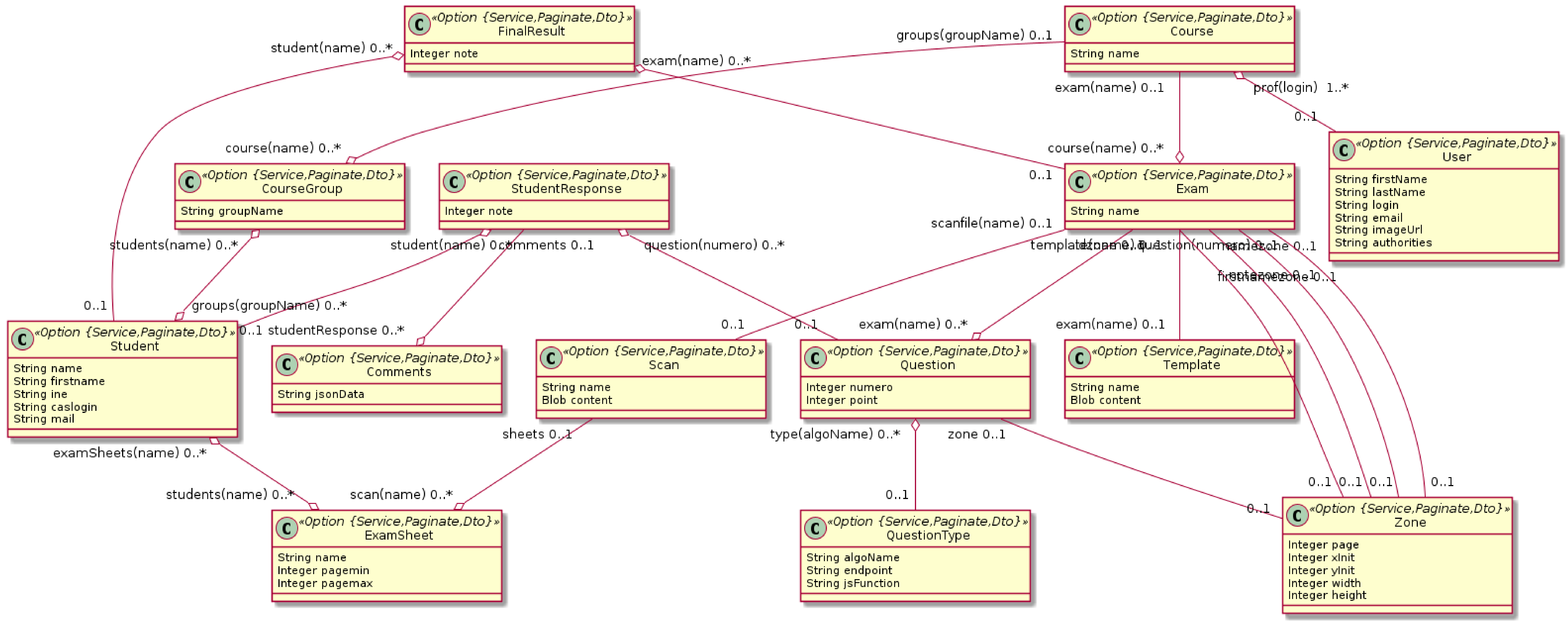
Conceptual architecture



deployed architecture



Business model



JHipster to generate the application skeleton

> JHipster is a development platform to quickly generate, develop, & deploy modern web applications & microservice architectures.

- jhipster 7.7.0 for the front
- jhisper 6.10.15

> Manual alignement of the API evolution 😀 (@Djamel)

Demo and code overview

- demo
- repo back
- repo front

A simple RPC to manage the communication between openCV worker and Angular

In the worker

```
addEventListener('message', e => {
  switch (e.data.msg) {
    case 'hello': {
      const response = `worker response to ${e.data.msg}`;
      postMessage({ msg: response, uid: e.data.uid });
      break;
    }
    ...
```

In the service

```
ready!: Promise<void>;
 subjects = new Map<string, Subject<any>>();
 constructor() {
   if (typeof Worker !== 'undefined') {
     // Create a new
     this.worker = worker;
     this.worker.onmessage = ({ data }) => {
       if (this.subjects.has(data.uid)) {
         this.subjects.get(data.uid)?.next(data.payload);
         this.subjects.get(data.uid)?.complete();
       }
     };
     this.worker.onerror = e => {
       if (this.subjects.has((e as any).uid)) {
         this.subjects.get((e as any).uid)?.error(e);
       }
     };
     const p = new Subject();
     this.subjects.set('0', p); // Ask to load opencv wasm to the worker
     this.ready = new Promise((res, rej) => {
       this.subjects
         .get('0')?.asObservable().subscribe(() => {res();},() => rej());
     });

     this.worker.postMessage({ msg: 'load', uid: '0' });
```

In the service

```
 private _dispatch<T>(msg1: any, pay: any): Observable<T> {
  const uuid1 = uuid(); // ⇨ '9b1deb4d-3b7d-4bad-9bdd-2b0d7b3dcb6d'
  this.ready.then(() => {
    this.worker.postMessage({ msg: msg1, uid: uuid1, payload: pay });
  });
  const p = new Subject<T>();
  this.subjects.set(uuid1, p);
  this.ready = new Promise((res, rej) => {
    this.subjects.get(uuid1)?.asObservable().subscribe(() =>
        {res();},
        () => rej()
      );
  });
  return p.asObservable();
}
public imageProcessing(image: ImageData): Observable<ImageData> {
  return this._dispatch('imageProcessing', image);
}
```

OpenCV architecture

Alignement algo

- Marker detection
  - circle detection
  - verify if the circle is filled
  - If we find 4 markers on the orginal exam and in the scan copy

```
let h = cv.findHomography(dstTri, srcTri, cv.RANSAC);
cv.warpPerspective(srcMat2, dst, h, dsize);
```

- ELSE find markers based on AKAZE (longer process), filter found marker with custom heuristic

```
let h = cv.findHomography(dstTri, srcTri, cv.RANSAC);
cv.warpPerspective(srcMat2, dst, h, dsize);
```

Plug your own microservice to correct a custom question type

- Two kinds: *Preprocessor*, *Automatic correction*

correct(qs: Questions)

Learn More

Documentations · GitHub · Showcases