

Projekt Online-Shop

Luca Malisan, JEE 2025

Inhaltsverzeichnis

Anforderungen und Komponenten

Verwendete Technologien

Datenbank

Deployment-Konzept

Fokusthema

Demonstration der Lösung

Anforderungen und Komponenten

Artikelkatalog

- | | |
|----|---------------------------------------------------------------|
| K1 | Filter Volltextsuche |
| K2 | Filter Haupt-/Subkategorie |
| K3 | Anzeigen von Artikelbild, Titel, Preis, Rabatt, Verfügbarkeit |
| K4 | Pagination |
| K5 | Öffnen der Detailseite durch Klick auf Artikel |

Detailansicht

- | | |
|----|---------------------------------------------------------------------|
| D1 | Zusätzlich Anzeige aller Bilder, Beschreibung und verfügbare Anzahl |
| D2 | Warenkorb-Button und Auswahl der Anzahl |

Warenkorb

- | | |
|----|--------------------------------------------|
| W1 | Anzeige aller Artikel |
| W2 | Änderung der Anzahl / Löschen |
| W3 | Anzeige Preis mit/ohne Mwst. sowie Rabatte |

Login-Page

- | | |
|----|---------------------------------------------------------------------|
| L1 | Login-Button für Redirect auf Auth0 Page |
| L2 | Artikel sind ohne Login sichtbar, für Warenkorb wird Login verlangt |
| L3 | Login-Fenster |
| L4 | Bestätigungsmail für Login-Daten |

REST-API

- | | |
|----|--------------------------------------------------------------------------|
| R1 | REST-API für CRUD-Operationen auf Artikeln, Kategorien und Subkategorien |
| R2 | Absicherung der REST-API durch Bearer Token |

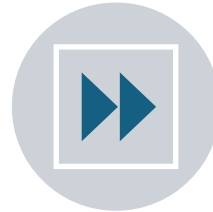
Verwendete Technologien



JAKARTA EE 11.0
BASISTECHNOLOGIE



MAVEN
BUILD-TOOL



LOMBOK
ENTWICKLUNGS-
UNTERSTÜTZUNG



AUTH0, JWT
LOGIN & API-
AUTHENTIFIZIERUNG



JUNIT, MOCKITO
UNIT-TESTING



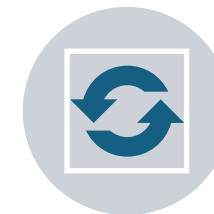
JACOCO
TESTING REPORTS



SWAGGER
API-DOKUMENTATION

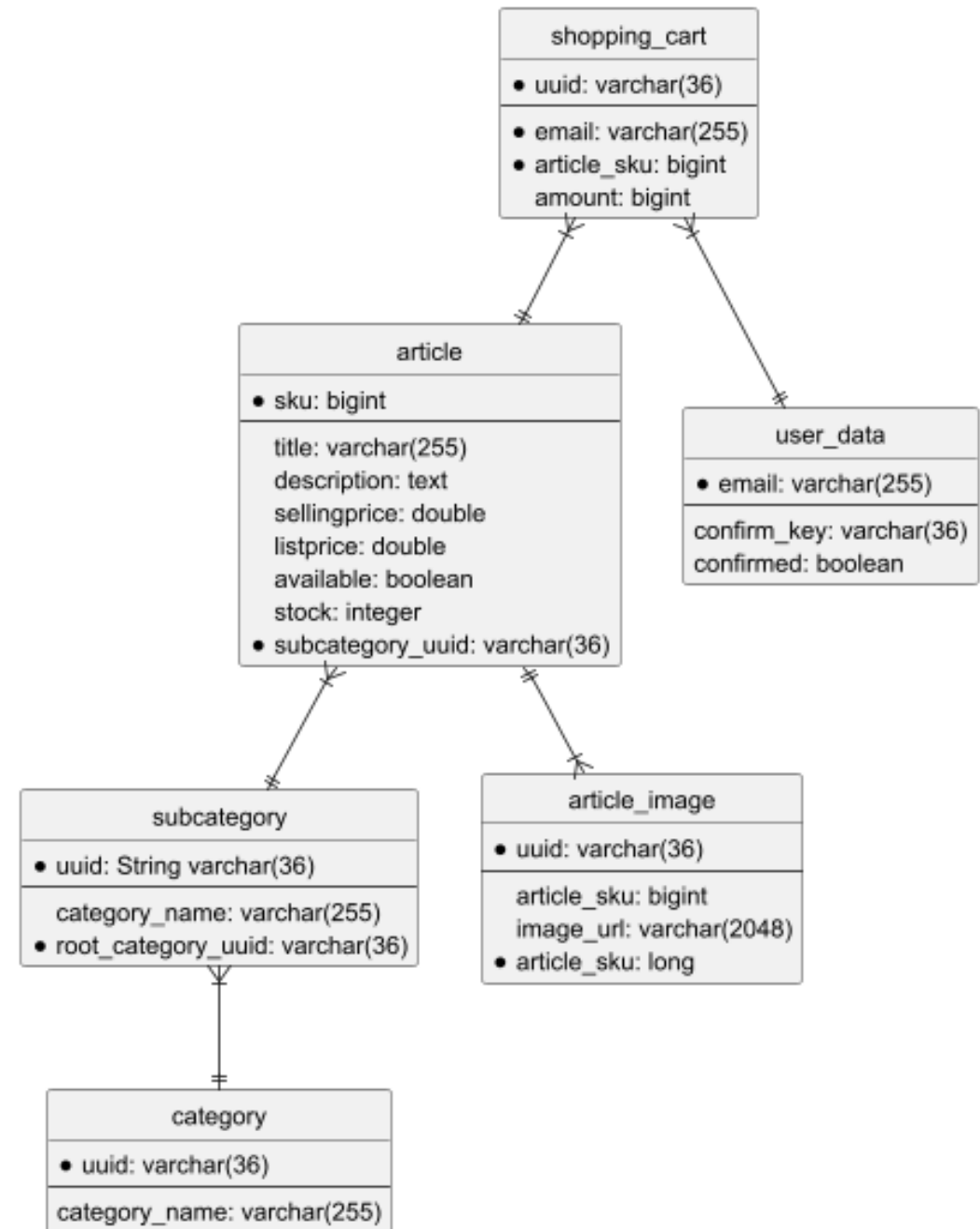


GLASSFISH
DEPLOYMENT / WEB-SERVER

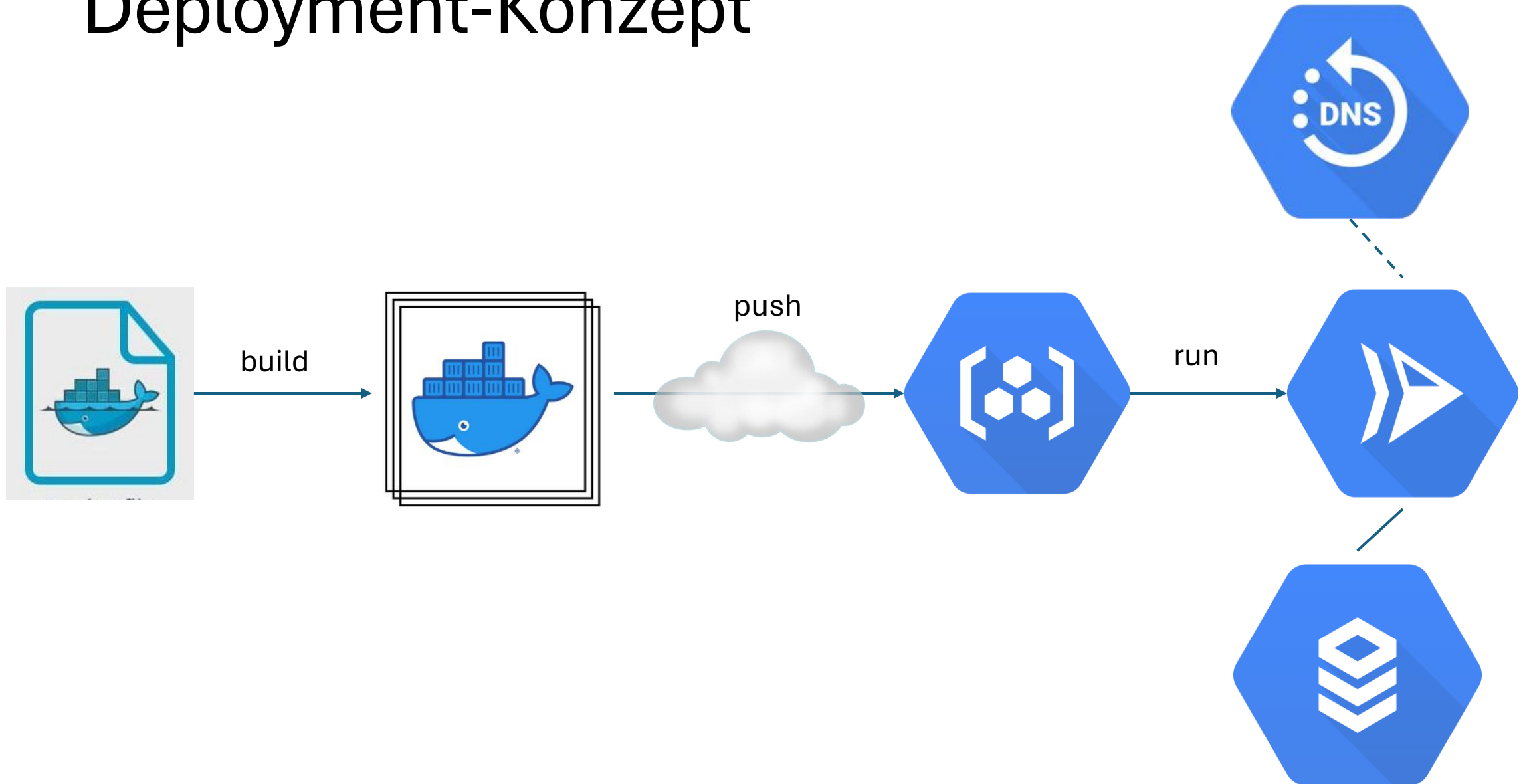


GITHUB ACTIONS
LATEX -> PDF
SWAGGER -> HTML

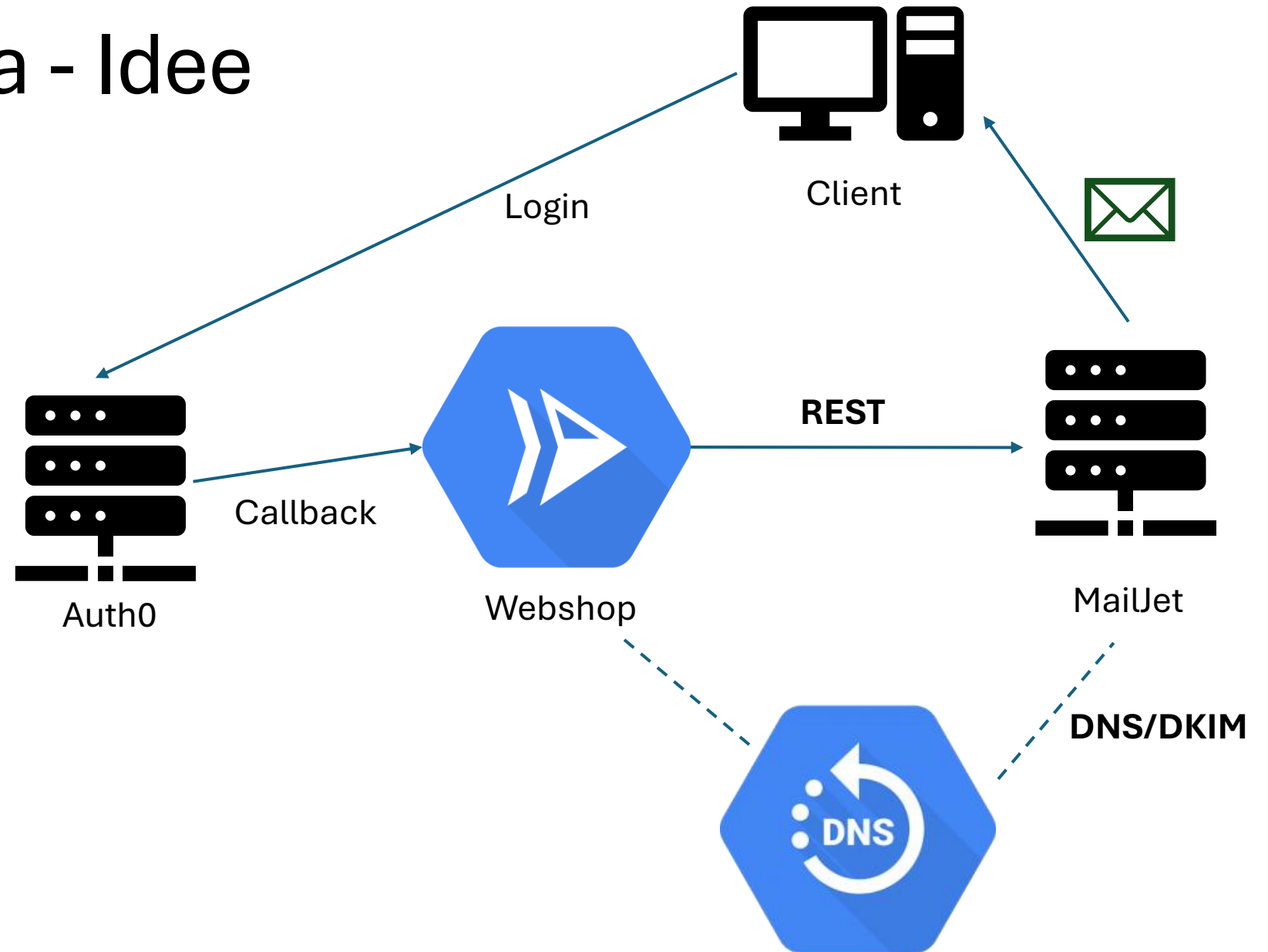
Datenbank (PostgreSQL)



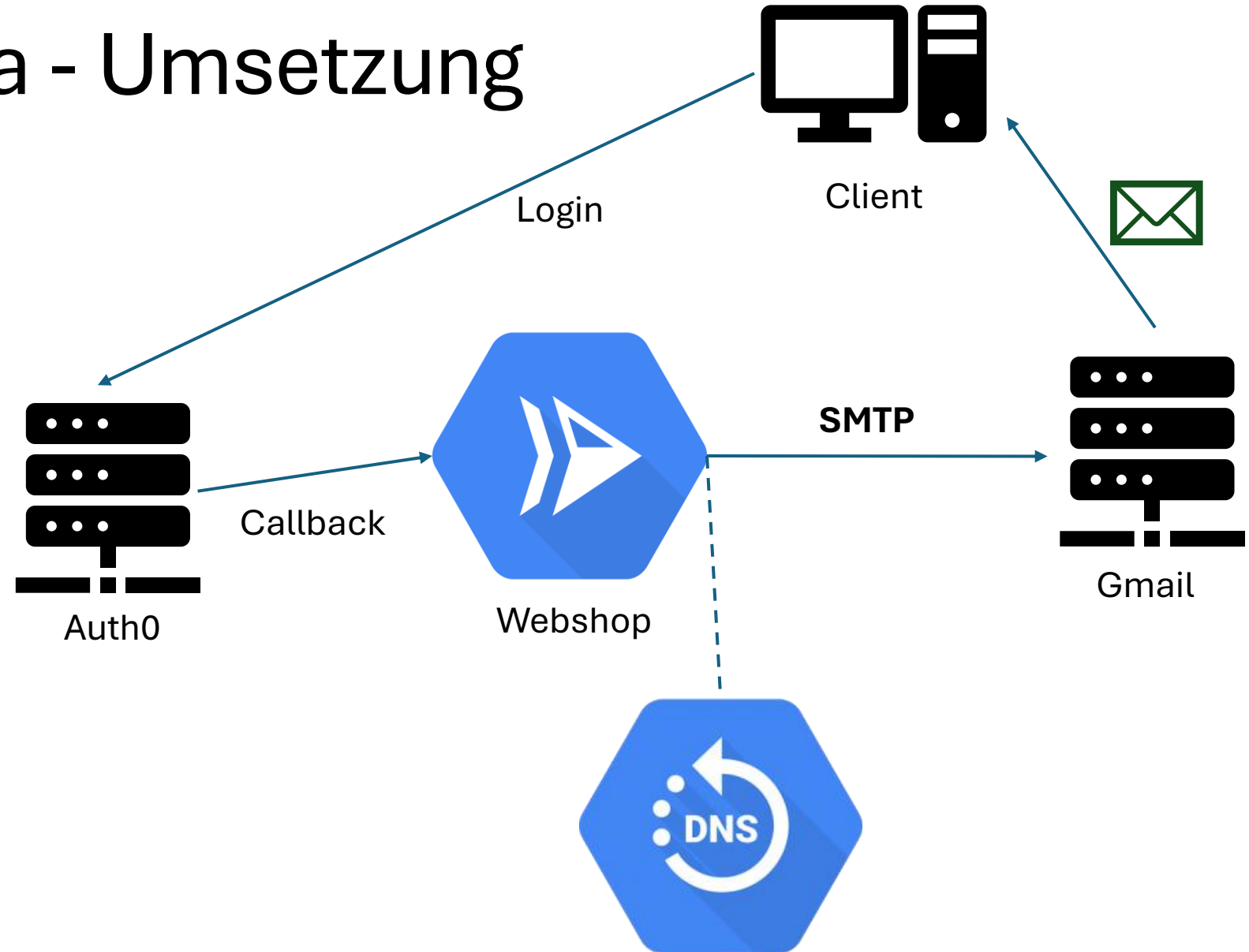
Deployment-Konzept



Fokusthema - Idee



Fokusthema - Umsetzung



Fokusthema - Mailjet vs SMTP

Mailjet ESP (Free Plan)

- Geeignet für Marketing [3]
- Statistiken [3]
- SMTP-Relay [3]
 - Weniger Spam-Risiko durch vertrauenswürdigen Absender [4]
 - Ausgerichtet auf Application Mail = mehr Skalierbarkeit [4]
 - Support / Ansprechperson bei Problemen [4]

Gmail SMTP (Personal Account)

- Kleines Volumen (max. 500 Mails / 24h) [1]
- Geeignet für Testing [2]
- Keine eigene Domain sondern nur @gmail = hohes Risiko für Spam-Klassifizierung [1]
- Keinerlei Marketing-Funktionen [1]
 - Keine Templates
 - Keine Kundensegmentierung
 - Kein automatisierter Mailversand
 - Potentiell viele kostenpflichtige Extensions nötig (z.B. für GDPR-Compliance)
 - Keine Analytics

Überblick ESP

Name	Free-Plan	Maximale Mails/Monat
Mailjet	6'000 Mails / Monat (200/d) [3]	15'000 Mails / Monat für 25€ monatlich [3]
E-Mail Octopus	10'000 E-Mails / Monat [5]	5'000'000 Mails/Monat für 1224€ monatlich [5]
Mailsender	500 Mails / Monat [6]	50'000 Mails / Monat für 75€ monatlich [6]
Mailgun	3000 Mails / Monat [7]	100'000 Mails / Monat für 82€ monatlich [7]
Mailomat	200 Mails / Monat [8]	ca. 83300 Mails / Monat für CHF 249 monatlich [8]
Amazon SES	x	\$0.0001 pro Mail, \$0.12 pro Gigabyte Anhang [9]

Code für Mailjet

```
public void sendMail(UserEmailConfirmed userEmailConfirmed, String baseUrl)
    throws NamingException, MailjetException {
    MailjetClient client;
    Context env = (Context) new InitialContext().lookup("java:comp/env");

    client =
        new MailjetClient(
            (String) env.lookup("mailjet.apiKey"), (String) env.lookup("mailjet.secretKey"));
    MailjetRequest request =
        new MailjetRequest(Emailv31.resource)
            .property(
                Emailv31.MESSAGES,
                new JSONArray()
                    .put(
                        new JSONObject()
                            .put(
                                Emailv31.Message.FROM,
                                new JSONObject().put("Email", "luca.malisan@students.ffhs.ch"))
                            .put(
                                Emailv31.Message.TO,
                                new JSONArray()
                                    .put(
                                        new JSONObject()
                                            .put("Email", userEmailConfirmed.getEmail()))
                            .put(Emailv31.Message.SUBJECT, "Please confirm your email")
                            .put(
                                Emailv31.Message.TEXTPART,
                                "Thanks for your registration at JEE webshop")
                            .put(
                                Emailv31.Message.HTMLPART,
                                String.format(
                                    "Thanks for your registration at JEE webshop. Please confirm your email <a
                                    baseUrl, userEmailConfirmed.getConfirmKey()))));
```

```

public void sendMail(User user, String baseUrl)
    throws NamingException, MessagingException {
    Context env = (Context) new InitialContext().lookup( name: "java:comp/env");

    String to = user.getEmail();
    String from = (String) env.lookup( name: "email.address");

    Properties props = new Properties();
    props.put("mail.smtp.auth", "true");
    props.put("mail.smtp.starttls.enable", "true");
    props.put("mail.smtp.host", "smtp.gmail.com");
    props.put("mail.smtp.port", "587");

    //authenticate with email account credentials (email + app-password)
    Session session =
        Session.getInstance(
            props,
            getPasswordAuthentication() -> {
                try {
                    return new PasswordAuthentication(from, (String) env.lookup( name: "email.password"));
                } catch (NamingException e) {
                    throw new RuntimeException(e);
                }
            }
        );
}

```

```

//set properties of e-mail
Message message = new MimeMessage(session);
message.setFrom(new InternetAddress(from));
message.setRecipients(Message.RecipientType.TO, InternetAddress.parse(to));
message.setSubject("Please confirm your email");

//provide link to user containing the confirmation key stored in database
message.setText(
    String.format(
        "Thanks for your registration at JEE webshop. Please confirm your email at %s/confirm-email",
        baseUrl, user.getConfirmKey());
);

new Thread(
    () -> {
        try {
            Transport.send(message);
        } catch (Exception ignored) {
        }
    }
).start();
}

```

Fokusthema - Code

Fokusthema - Code

<http://localad-9cineg7:8080/jee-webshop/application/confirm-email/07ea9ac8-5d0f-431d-9032-bb8065f248f5>

	email	confirm_key	confirmed
1	lucamali@icloud.com	247e0b3b-8056-46f0-adfb-7edba56e000b	• true
2	luca.malisan@students.ffhs.ch	07ea9ac8-5d0f-431d-9032-bb8065f248f5	false



Demonstration der
Lösung

Literaturverzeichnis

- [1] <https://emailoctopus.com/blog/email-marketing-with-gmail-pros-cons-alternatives>
- [2] <https://medium.com/rails-to-rescue/how-to-set-up-smtp-credentials-with-gmail-for-your-app-send-email-cf236d11087d>
- [3] <https://www.mailjet.com/de/preise/>
- [4] <https://postmarkapp.com/blog/smtp-relay-services>
- [5] <https://emailoctopus.com/pricing>
- [6] <https://www.mailersend.com/pricing?currency=eur&billing=yearly>
- [7] <https://www.mailgun.com/pricing/?currency=eur>
- [8] <https://mailomat.swiss/preise/preismodell>
- [9] <https://aws.amazon.com/ses/pricing/>