

# **Entwicklung von PaMesAn**

Implementierung eines neuen Systems zur Erfassung von  
Versandverpackungen mit Hilfe von Bild- und Sensordaten zur  
Erfüllung der Novelle des Verpackungsgesetzes

---

Johannes Leyrer

11.01.2023

FLYERALARM - Azubi-Nr.: 468322

Projektumfeld

Planung

Analyse

Entwurf

Implementierung

Fazit

## Projektumfeld

---

## FLYERALARM GmbH

2002 > 2000 > 3 Mio  
gegründet Mitarbeiter Produkte

## FLYERALARM GmbH

2002	> 2000	> 3 Mio
gegründet	Mitarbeiter	Produkte

- Eines der größten E-Commerce Unternehmen Deutschlands

## FLYERALARM GmbH

2002	> 2000	> 3 Mio
gegründet	Mitarbeiter	Produkte

- Eines der größten E-Commerce Unternehmen Deutschlands
- Führende Online-Druckerei Europas im B2B-Bereich

## FLYERALARM GmbH

2002	> 2000	> 3 Mio
gegründet	Mitarbeiter	Produkte

- Eines der größten E-Commerce Unternehmen Deutschlands
- Führende Online-Druckerei Europas im B2B-Bereich
- Betreibt eigenen Onlineshop

## FLYERALARM GmbH

2002	> 2000	> 3 Mio
gegründet	Mitarbeiter	Produkte

- Eines der größten E-Commerce Unternehmen Deutschlands
- Führende Online-Druckerei Europas im B2B-Bereich
- Betreibt eigenen Onlineshop
- Erstellung der Druckdaten

## FLYERALARM Industrial Print GmbH

8                ca. 1200  
Standorte    Mitarbeiter

## FLYERALARM Industrial Print GmbH

8                   ca. 1200  
Standorte   Mitarbeiter

- Tochtergesellschaft

## FLYERALARM Industrial Print GmbH

8                   ca. 1200  
Standorte   Mitarbeiter

- Tochtergesellschaft
- Eigene IT-Abteilung

## FLYERALARM Industrial Print GmbH

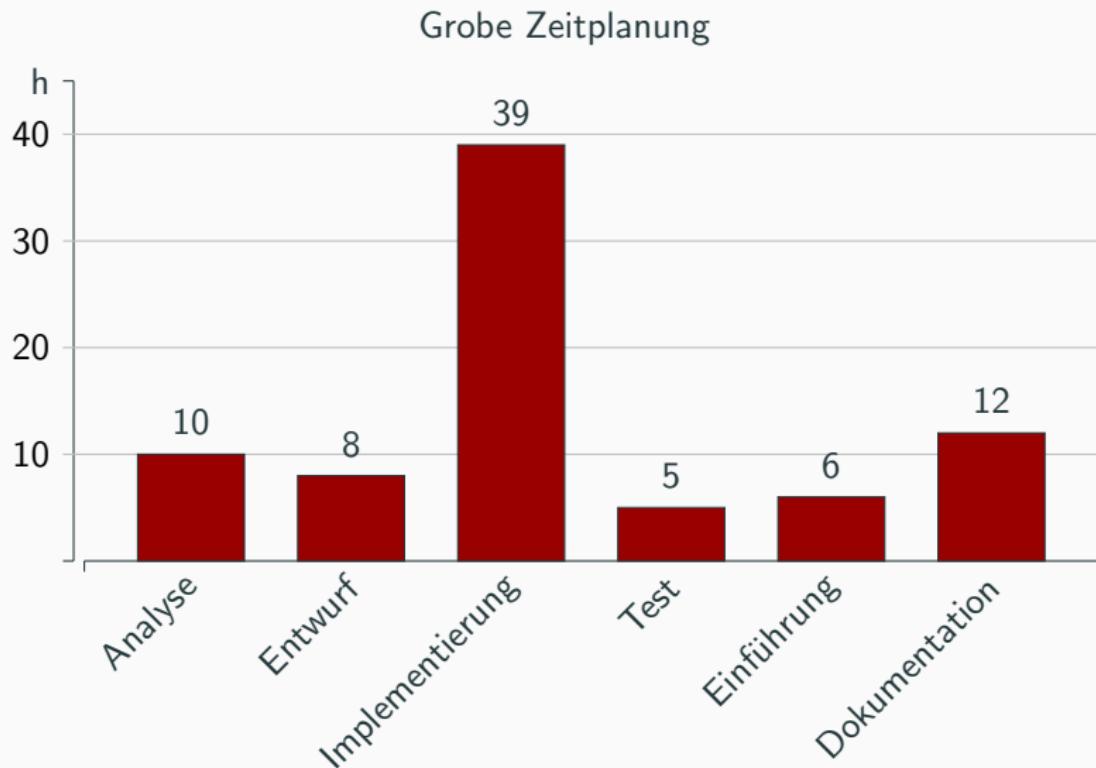
8	ca. 1200
Standorte	Mitarbeiter

- Tochtergesellschaft
- Eigene IT-Abteilung
- Produktion und Versand

# Planung

---

# Zeitschätzung Projektphasen



# Analyse

---

## Ist-Analyse

- Paketgröße und Anzahl wird manuell gepflegt

## Ist-Analyse

- Paketgröße und Anzahl wird manuell gepflegt
- Neue Anforderungen durch Verpackungsgesetz

## Ist-Analyse

- Paketgröße und Anzahl wird manuell gepflegt
- Neue Anforderungen durch Verpackungsgesetz
- Hoher Wartungsaufwand für Entwickler

## Ist-Analyse

- Paketgröße und Anzahl wird manuell gepflegt
- Neue Anforderungen durch Verpackungsgesetz
- Hoher Wartungsaufwand für Entwickler
- Paketgröße ungenau

## Soll-Analyse

- Paketgröße und Anzahl automatisch erfassen

## Soll-Analyse

- Paketgröße und Anzahl automatisch erfassen
- Erfüllung der Anforderungen durch Verpackungsgesetz

## Soll-Analyse

- Paketgröße und Anzahl automatisch erfassen
- Erfüllung der Anforderungen durch Verpackungsgesetz
- Wartungsaufwand minimieren durch automatische Erfassung

# Angebot der Elektro Löther GmbH

## Angebotsvergleich Fa. Löther - Kartonagenerkennung

Standorte	Dillberg	Heuchelhof	Klipphausen	Kesselsdorf	Summe Standorte
<b>Kostenaufteilung</b>					
Kamerahardware	13.338,79 €	13.338,79 €	13.338,79 €	19.081,04 €	59.097,41 €
Eletrik / Mechanik	2.095,00 €	2.095,00 €	2.675,00 €	3.845,00 €	10.710,00 €
Software & IBN	5.225,00 €	2.900,00 €	5.150,00 €	6.900,00 €	20.175,00 €
Summe	20.658,79 €	18.333,79 €	21.163,79 €	29.826,04 €	89.982,41 €
19% MwSt.	3.925,17 €	3.483,42 €	4.021,12 €	5.666,95 €	17.096,66 €
Summe Gesamt	<b>24.583,96 €</b>	<b>21.817,21 €</b>	<b>25.184,91 €</b>	<b>35.492,99 €</b>	<b>107.079,07 €</b>

# Kostenverteilung Hardware

<b>Hardware</b>	<b>Gesamt</b>
ARCELI Shield Board Kit	17,99 €
AZDelivery 5 x Mega 2560 R3	14,99 €
Benewake TF MINI PLUS	182,40 €
Microsoft Lifecam Studio	42,99 €
item-Systemprofile	29,52 €
item-Verbindungsstücke	192,00 €
item-Füße	48,00 €
Dell Wyse 5070 Thin Client	450,00 €
<b>Gesamtkosten</b>	<b>977,89 €</b>

# Kostenverteilung Personal und Hardware

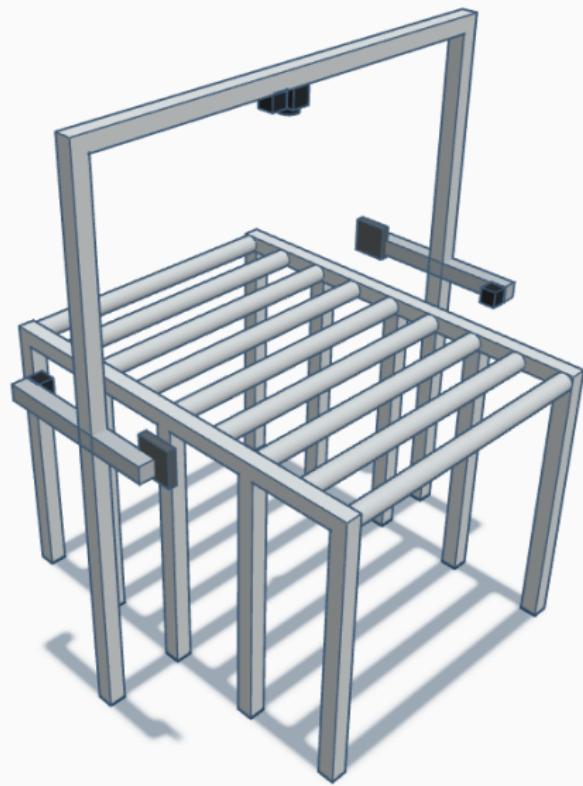
Personal	Zeit in Stunden	Kosten pro Stunde	Gesamt
Auszubildender	80	6,00 € + 15,00 €	1680,00 €
Teamleitung	2	31,50 € + 15,00 €	93,00 €
Teammitglied	2	21,50 € + 15,00 €	73,00 €
Haustechnik	8	19,00 € + 15,00 €	272,00 €
<b>Gesamtkosten</b>			<b>2118,00 €</b>

**Gesamtkosten Personal und Hardware: 3095,89 €**

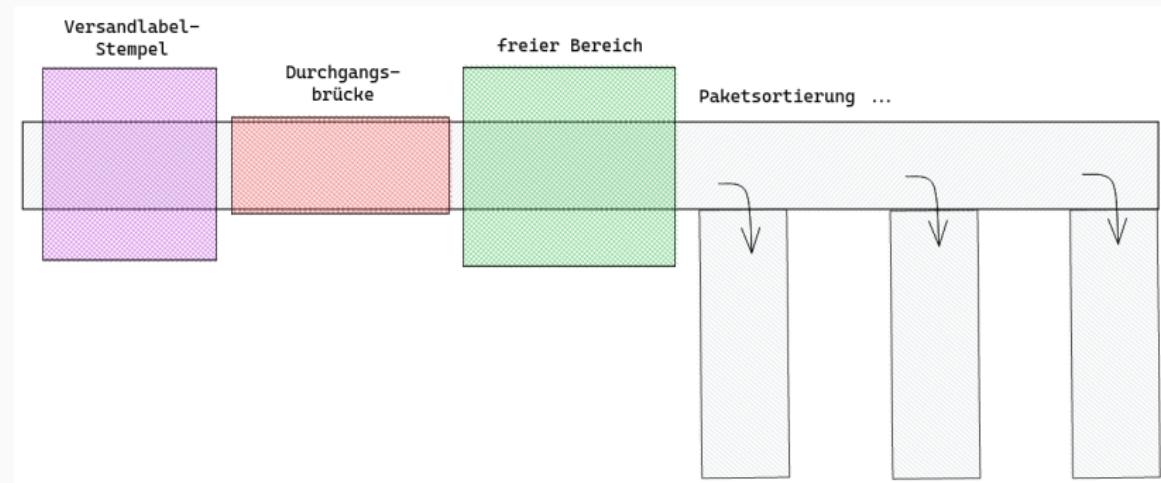
# Entwurf

---

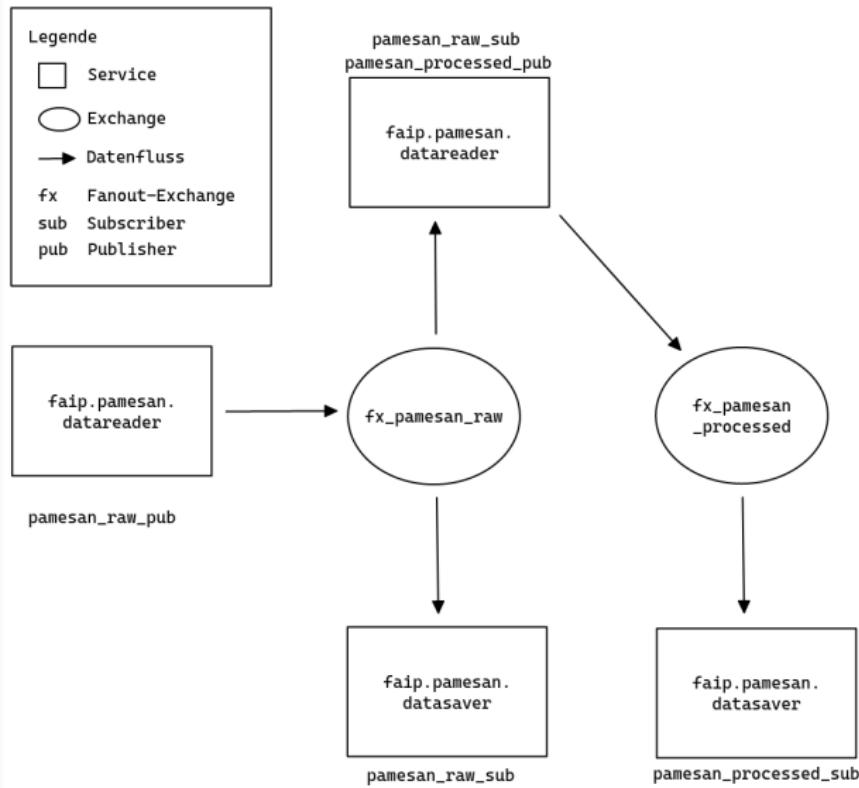
# Sensorträger



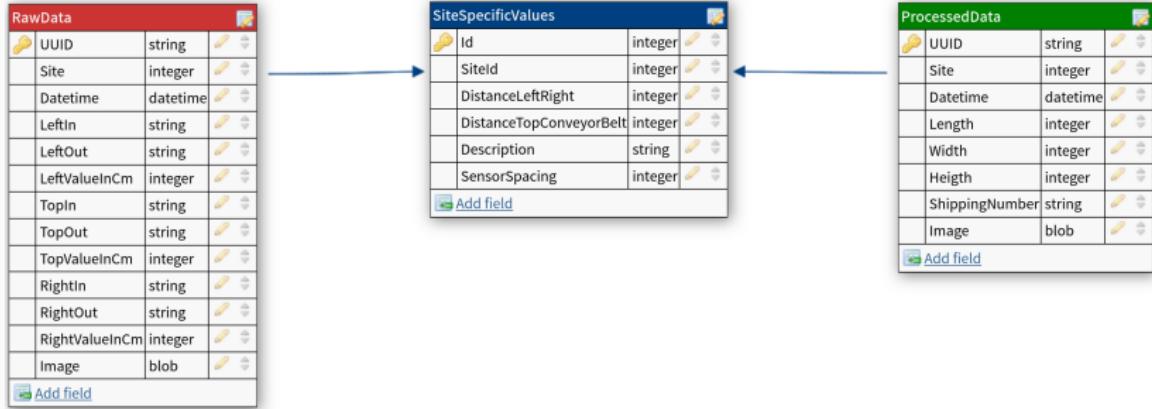
# Standort



# Programmübersicht



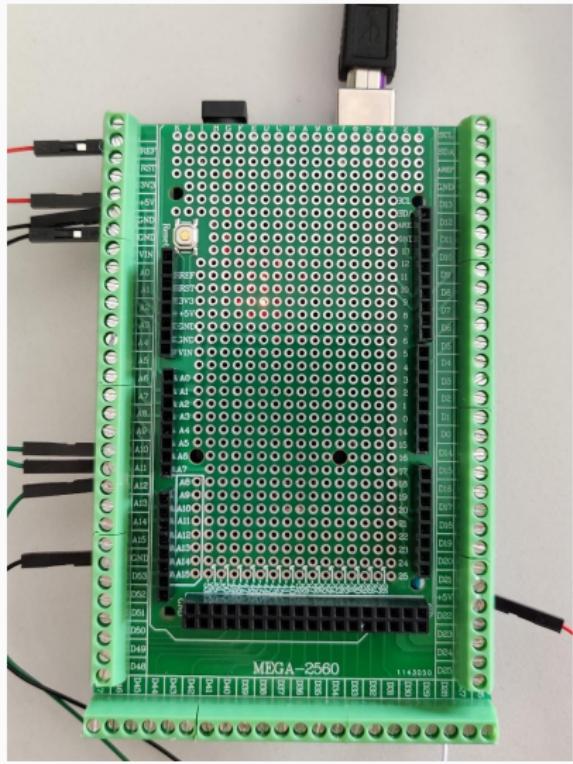
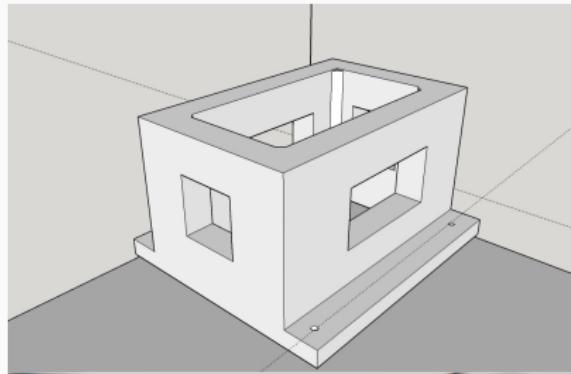
# Datenmodell



## Implementierung

---

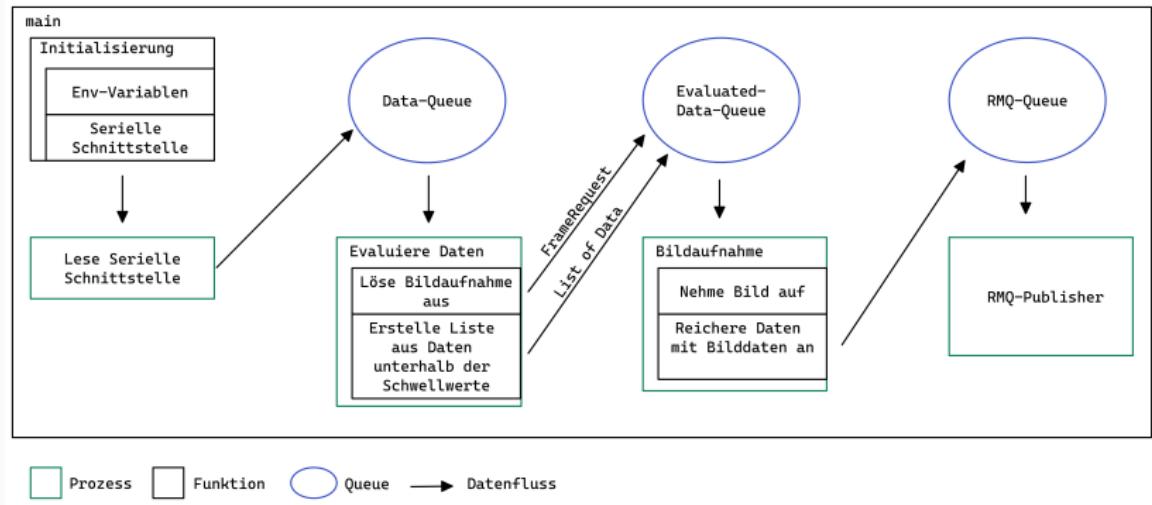
# Lasersensor und Arduino



# Umsetzung des Sensorträgers

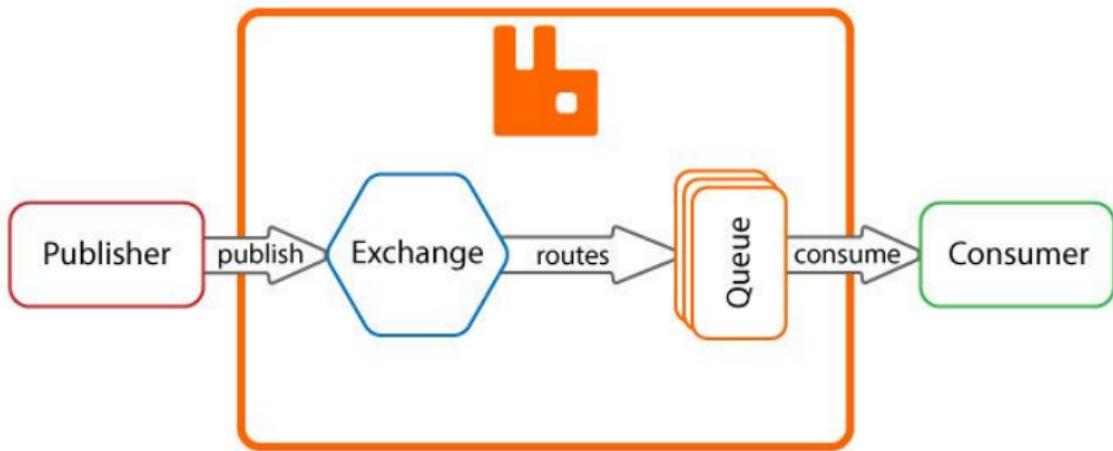


# Implementierung des Ausleseservice

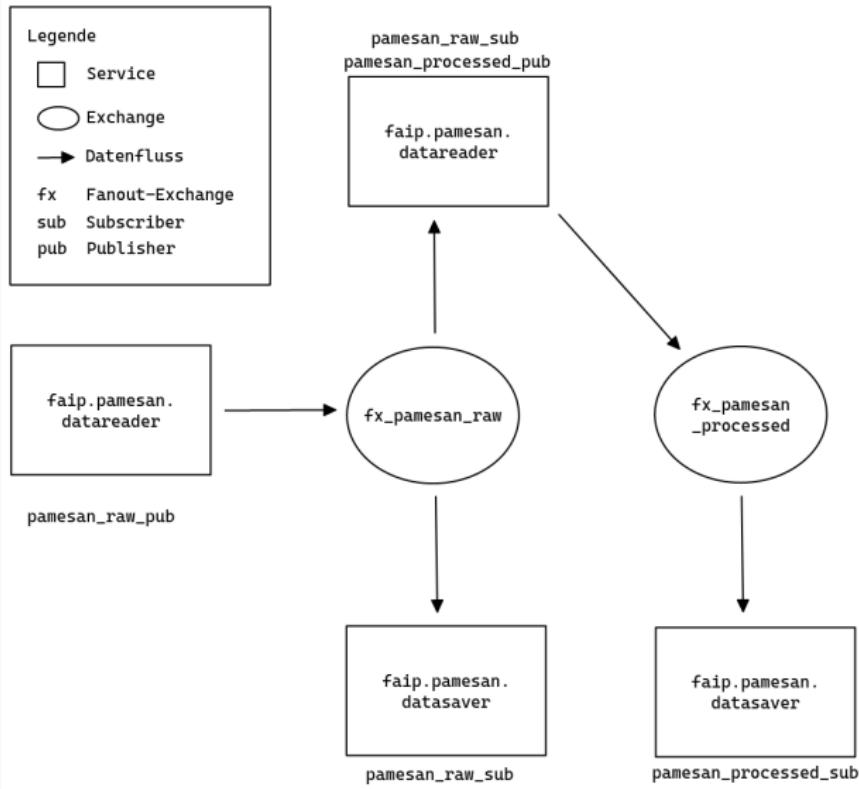


# Implementierung des Ausleseservice – Pika

```
75 def rmq_sender(queue: Queue):
76     config_reader = EnvConfig()
77     config_reader.initialize_env()
78     pub = Publisher(config_reader)
79     pub.connect()
80     site_id = config_reader.get_site_id()
81     while True:
82         try:
83             if not queue.empty():
84                 data = queue.get()
85                 hyd_data = data_hydration(data, site_id)
86                 pub.publish(hyd_data)
87
88                 time.sleep(0.5)
89         except Exception as ex:
```



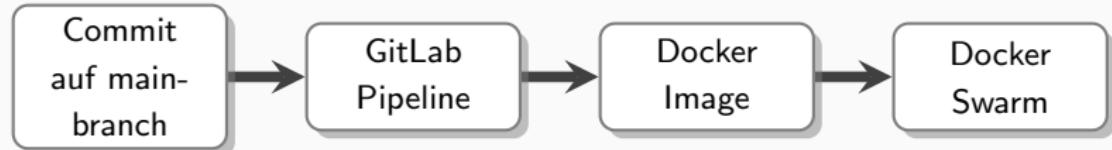
# RabbitMQ



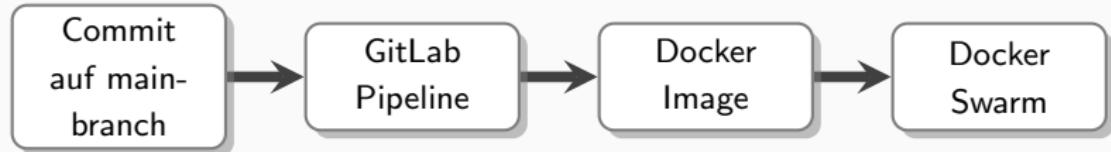
# Implementierung der Datenbanktabellen

```
1  CREATE TABLE PAMESAN.dbo.RawData (
2      UUID uniqueidentifier NOT NULL,
3      Site int NOT NULL,
4      [Datetime] datetime NULL,
5      LeftIn nvarchar(50) COLLATE Latin1_General_CI_AS
6      NULL,
7
8      :
9
10     [Image] varbinary(MAX) NULL,
11     CONSTRAINT PK_RawData PRIMARY KEY (UUID),
12     CONSTRAINT RawData_FK FOREIGN KEY (Site)
13     REFERENCES PAMESAN.dbo.SiteSpecificValues(Id)
14 );
15
16
17 
```

# Implementierung des Datenspeicherservice



# Implementierung des Datenspeicherservice

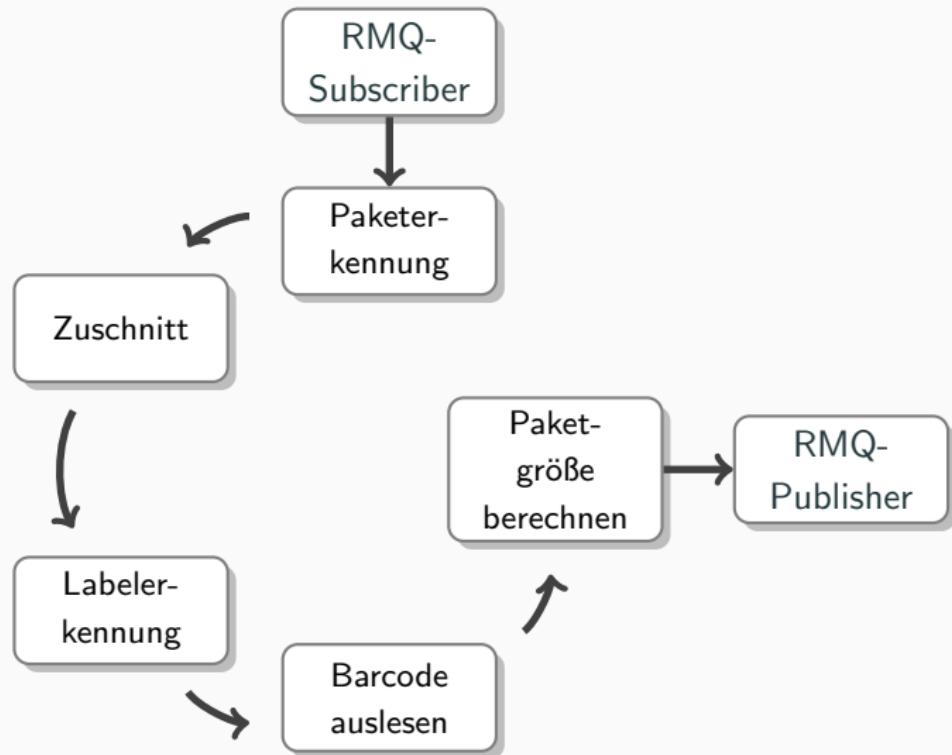


- 1 Scaffold –DbContext "Data Source=sql-mar-01.druckhaus.local; Initial Catalog=PAMESAN; persist security info=True; user id=pamesan-rw; password=\*\*\*\*\*" Microsoft.  
EntityFrameworkCore.SqlServer –OutputDir DatabaseContext –Tables RawData, ProcessedData

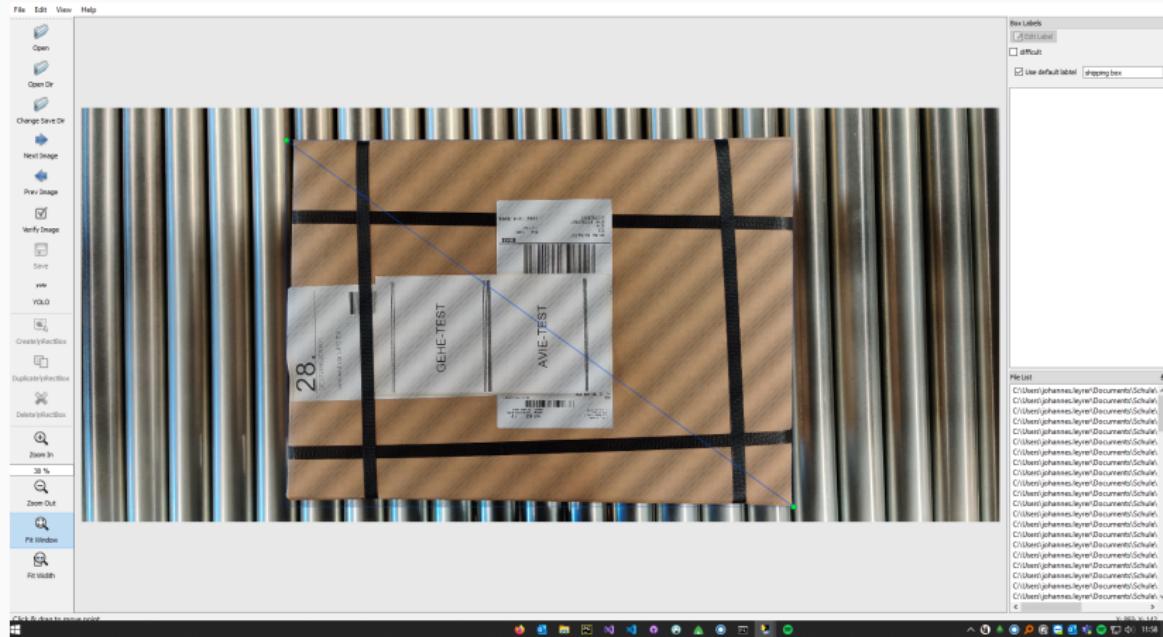
# Implementierung des Datenspeicherservice

```
38 internal static void SaveDataToRaw(string data,
39     PAMESANContext dbContext) {
40     try {
41         rawData rawData = JsonConvert.DeserializeObject<
42             rawData>(data)!;
43
44         dbContext.RawData.Add(rawData);
45         dbContext.SaveChanges();
46     }
47     catch {
48         throw;
49     }
50 }
```

# Implementierung des Datenverarbeitungsservice



# Labeln der Bilder



# Training des YOLOv7-Models

```
1  python train.py --device 0 --batch-size 16 --epochs 100  
   --img 640 640 --data data/custom_data.yaml --hyp data  
   /hyp.scratch.custom.yaml --cfg cfg/training/yolov7_custom.  
   yaml --weights yolov7.pt --name yolo7-custom  
2
```

# Ergebnis des YOLOv7-Models



# Auslesen des Versandlabels



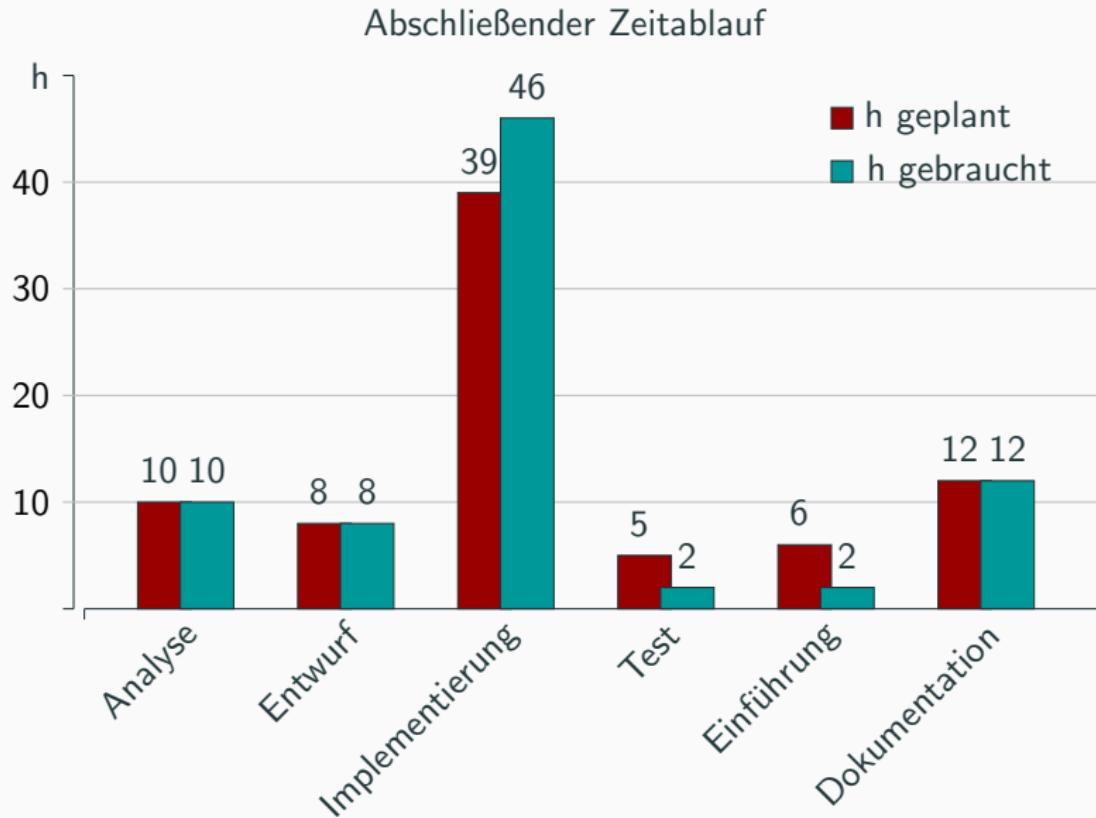
# Berechnung der Paketgröße

HIER BERECHNUNG ERSTELLEN UND EINFÜGEN

## Fazit

---

# Soll-/Ist-Vergleich



# Lessons Learned und Ausblick

## Lessons Learned

- YOLOv7 als mit geringem Vorwissen leicht einsetzbarer Objektdetektor

## Lessons Learned

- YOLOv7 als mit geringem Vorwissen leicht einsetzbarer Objektdetektor
- Pufferzeit sollte 2 % der Gesamtzeit betragen

## Lessons Learned

- YOLOv7 als mit geringem Vorwissen leicht einsetzbarer Objektdetektor
- Pufferzeit sollte 2 % der Gesamtzeit betragen
- Industriekamera > Webcam → KEYENCE-Scanner

# Lessons Learned und Ausblick

## Lessons Learned

- YOLOv7 als mit geringem Vorwissen leicht einsetzbarer Objektdetektor
- Pufferzeit sollte 2 % der Gesamtzeit betragen
- Industriekamera > Webcam → KEYENCE-Scanner

## Ausblick

- Verwendung KEYENCE-Scanner und Umschreiben auf C#

# Lessons Learned und Ausblick

## Lessons Learned

- YOLOv7 als mit geringem Vorwissen leicht einsetzbarer Objektdetektor
- Pufferzeit sollte 2 % der Gesamtzeit betragen
- Industriekamera > Webcam → KEYENCE-Scanner

## Ausblick

- Verwendung KEYENCE-Scanner und Umschreiben auf C#
- Verknüpfung erfasster Abmessungen mit bekannten Kartonagen

# Lessons Learned und Ausblick

## Lessons Learned

- YOLOv7 als mit geringem Vorwissen leicht einsetzbarer Objektdetektor
- Pufferzeit sollte 2 % der Gesamtzeit betragen
- Industriekamera > Webcam → KEYENCE-Scanner

## Ausblick

- Verwendung KEYENCE-Scanner und Umschreiben auf C#
- Verknüpfung erfasster Abmessungen mit bekannten Kartonagen
- Aufbau an anderen Standorten

# Fragen?

# Literaturverzeichnis i

**Danke für die Aufmerksamkeit!**