

# 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

## Entwicklung von PaMesAn

2023-01-09

- Guten Morgen
- Johannes Leyrer
- Implementierung eines neuen Systems zur Erfassung von  
Versandverpackungen mit Hilfe von Bild- und Sensordaten zur Erfüllung  
der Novelle des Verpackungsgesetzes
- Entwicklung von PaMesAn - Paket-Messungs-Anlage

### 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

2023-01-09

### └ Gliederung

- Projektumfeld
- Planungsphase
- Analysephase
- Entwurf
- Implementierung
- Fazit und Ausblick

# Projektumfeld

---

Entwicklung von PaMesAn  
└ Projektumfeld

2023-01-09

Projektumfeld

---

## FLYERALARM GmbH

2002 > 2000 > 3 Mio  
gegründet Mitarbeiter Produkte

### Projektumfeld

#### FLYERALARM GmbH

1. FLYERALARM GmbH, 2002 gegründet, > 2000 Mitarbeiter und > 3 Mio Produkte

## FLYERALARM GmbH

2002 > 2000 > 3 Mio  
gegründet Mitarbeiter Produkte

- Eines der größten E-Commerce Unternehmen Deutschlands

## Entwicklung von PaMesAn

### Projektumfeld

#### FLYERALARM GmbH

2023-01-09

## FLYERALARM GmbH

2002 > 2000 > 3 Mio  
gegründet Mitarbeiter Produkte

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

## Entwicklung von PaMesAn

### Projektumfeld

#### FLYERALARM GmbH

2023-01-09

## 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

## Entwicklung von PaMesAn

### Projektumfeld

#### FLYERALARM GmbH

2023-01-09



1. FLYERALARM GmbH, 2002 gegründet, > 2000 Mitarbeiter und > 3 Mio Produkte
2. Eines der größten E-Commerce Unternehmen Deutschlands
3. Führende Online-Druckerei Europas im B2B-Bereich
4. Eigener 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

## Entwicklung von PaMesAn

### Projektumfeld

#### FLYERALARM GmbH

2023-01-09

- 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

1. FLYERALARM Industrial Print GmbH (FAIP) an 8 Standorten mit ca.  
1200 MA

## FLYERALARM Industrial Print GmbH

8               ca. 1200  
Standorte   Mitarbeiter

- Tochtergesellschaft

Entwicklung von PaMesAn

└ Projektumfeld

└ FLYERALARM Industrial Print GmbH

2023-01-09

1. FLYERALARM Industrial Print GmbH (FAIP) an 8 Standorten mit ca. 1200 MA
2. Tochtergesellschaft der FLYERALARM GmbH

## FLYERALARM Industrial Print GmbH

8               ca. 1200  
Standorte   Mitarbeiter

- Tochtergesellschaft
- Eigene IT-Abteilung

Entwicklung von PaMesAn

└ Projektumfeld

└ FLYERALARM Industrial Print GmbH

2023-01-09

## FLYERALARM Industrial Print GmbH

8               ca. 1200  
Standorte   Mitarbeiter

- Tochtergesellschaft
- Eigene IT-Abteilung
- Produktion und Versand

## Entwicklung von PaMesAn

### Projektumfeld

#### FLYERALARM Industrial Print GmbH

2023-01-09

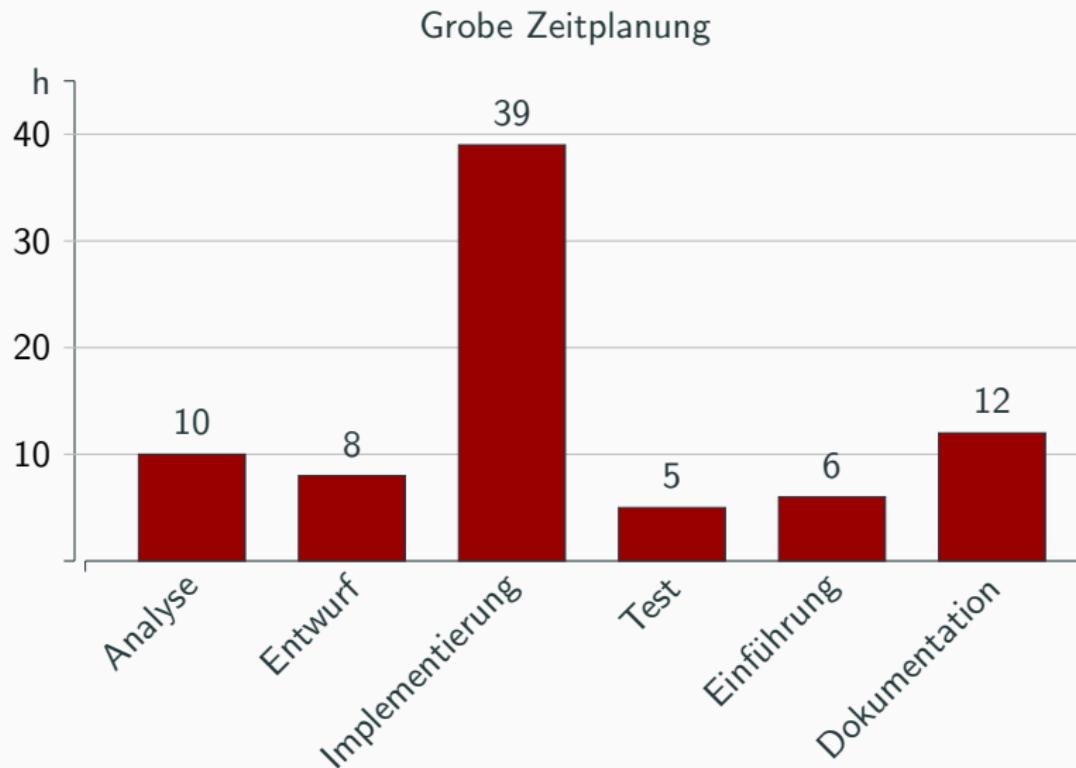
1. FLYERALARM Industrial Print GmbH (FAIP) an 8 Standorten mit ca. 1200 MA
2. Tochtergesellschaft der FLYERALARM GmbH
3. Eigene IT-Abteilung
4. Zuständig fuer Produktion und Versand

2023-01-09

# Planung

---

# Zeitschätzung Projektphasen



## Entwicklung von PaMesAn

### Planung

#### Zeitschätzung Projektphasen

- 10h Analyse
- 8h Entwurf
- 39h Implementierung
- 5h Test
- 6h Einführung
- 12h Dokumentation



2023-01-09

# Analyse

---

- Paketgröße und Anzahl wird manuell gepflegt

2023-01-09

## 1. Manuelle Pflege

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

2023-01-09

1. Manuelle Pflege
2. Anforderungen ans Verpackungsgesetz nicht erfüllt

## Ist-Analyse

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

## Entwicklung von PaMesAn

### Analyse

#### Ist-Analyse

2023-01-09

1. Manuelle Pflege
2. Anforderungen ans Verpackungsgesetz nicht erfüllt
3. Hohe Wartung

- 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

## Entwicklung von PaMesAn

### Analyse

#### Ist-Analyse

2023-01-09

1. Manuelle Pflege
2. Anforderungen ans Verpackungsgesetz nicht erfüllt
3. Hohe Wartung
4. Paketgröße ungenau

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

- Paketgröße und Anzahl automatisch erfassen

2023-01-09

## 1. Automatische Erfassung

## Soll-Analyse

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

## Entwicklung von PaMesAn

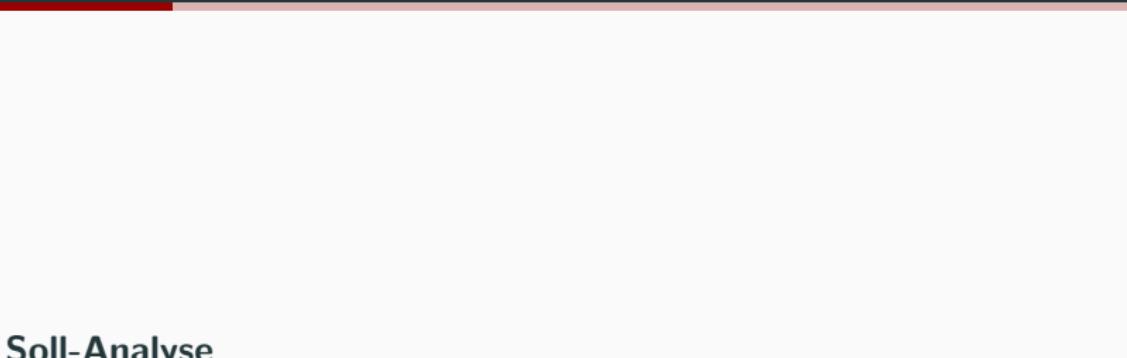
### Analyse

#### Soll-Analyse

2023-01-09

1. Automatische Erfassung
2. Helfen bei Anforderungen durch Verpackungsgesetz

- 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

2023-01-09

1. Automatische Erfassung
2. Helfen bei Anforderungen durch Verpackungsgesetz
3. Wartungsaufwand minimieren

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

## 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 & IBM	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 €
<b>Summe Gesamt</b>	<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>

## Entwicklung von PaMesAn

- Analyse

### Angebot der Elektro Löther GmbH

2023-01-09

Angebotsvergleich Fa. Löther - Kartonagenerkennung					
Standorte	Dillberg	Heuchelhof	Klipphausen	Kesselsdorf	Summe Standorte
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 & IBM	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 €
<b>Summe Gesamt</b>	<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>

- Angebote eingeholt
- Unternehmen Elektro Loether GmbH
- Kostenvoranschlauf fuer 4 Standorte
- Kesselsdorf 35k Euro

Hardware	Gesamt
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>

### Analyse

#### Kostenverteilung Hardware

2023-01-09

- Eigenentwicklung
- K977,89 Euro Hardware
- 3 verschiedene Sensor-Typen getestet

Hardware	Gesamt
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 h	Kosten pro h	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 €**

## Entwicklung von PaMesAn

### Analyse

#### Kostenverteilung Personal und Hardware

2023-01-09

Kostenverteilung Personal und Hardware

Personal	Zeit in h	Kosten pro h	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 €

- 2118 Euro Pesonalkosten
- Gesamtkosten 3095,89 Euro
- Etwa 30k Euro guenstiger in Eigenentwicklung
- Entscheidung fuer Eigenentwicklung

# Entwurf

---

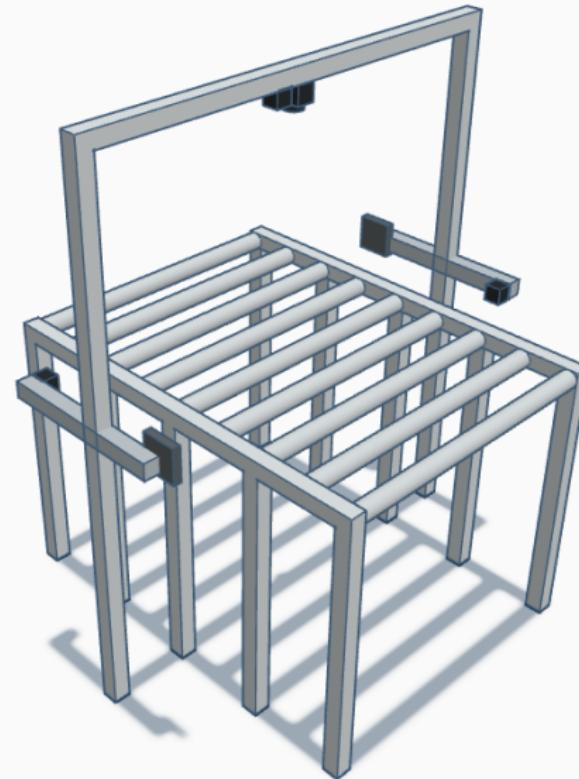
Entwicklung von PaMesAn  
└ Entwurf

2023-01-09

Entwurf

---

# Sensorträger



Entwicklung von PaMesAn

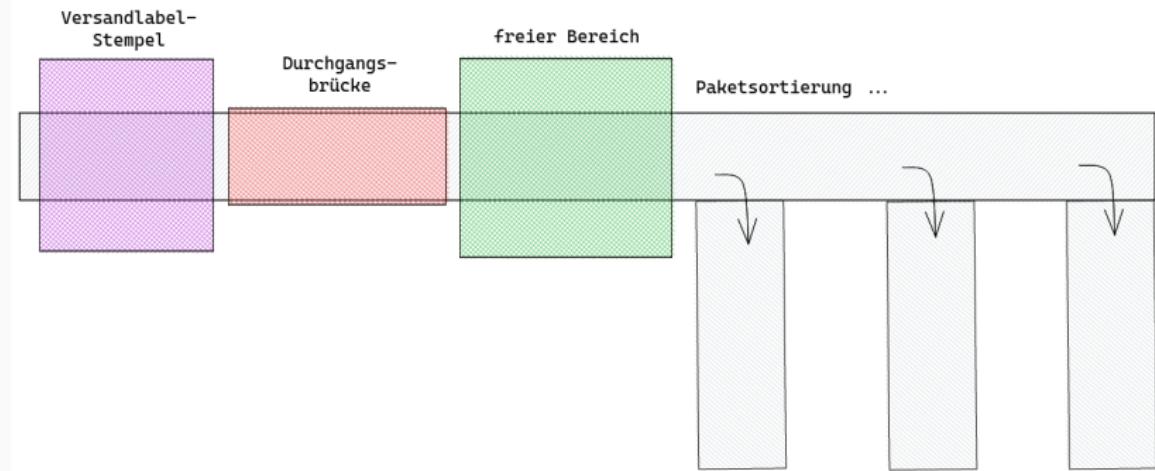
└ Entwurf

└ Sensorträger

2023-01-09

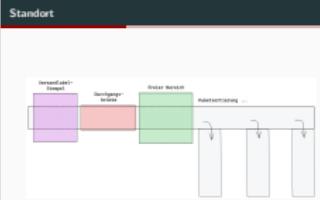
Sensorträger

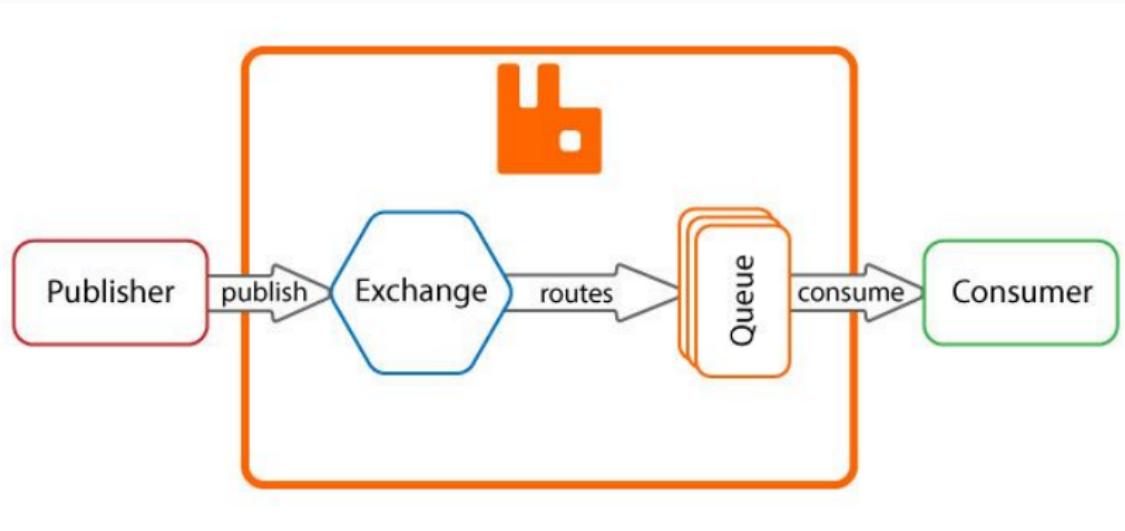




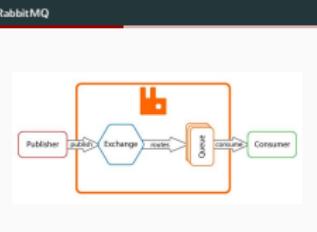
2023-01-09

- Standort
- Versandanlage



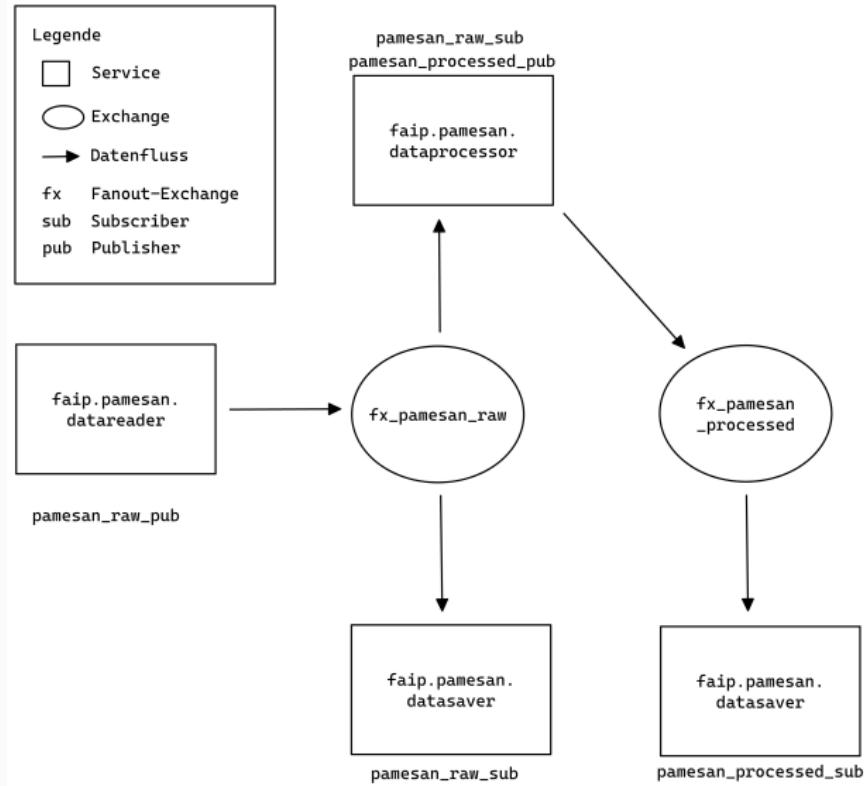


2023-01-09



- Microservices-Architektur
- Kommunikation ueber RabbitMQ
- RabbitMQ: Open Source Message Broker Software
- Postsystem

# Programmübersicht



## Entwicklung von PaMesAn

### Entwurf

#### Programmübersicht

- 4 Services, 2 Exchanges
- Datenauslese-Service
- Pamesan-Raw-Exchange
- Datenspeicherservice
- Datenverarbeitungsservice
- Datenspeicherservice



# Datenmodell

RawData		
UUID	string	
Site	integer	
Datetime	datetime	
LeftIn	string	
LeftOut	string	
LeftValueInCm	integer	
TopIn	string	
TopOut	string	
TopValueInCm	integer	
RightIn	string	
RightOut	string	
RightValueInCm	integer	
Image	blob	

SiteSpecificValues		
Id	integer	
SiteId	integer	
DistanceLeftRight	integer	
DistanceTopConveyorBelt	integer	
Description	string	
SensorSpacing	integer	

ProcessedData		
UUID	string	
Site	integer	
Datetime	datetime	
Length	integer	
Width	integer	
Height	integer	
ShippingNumber	string	
Image	blob	

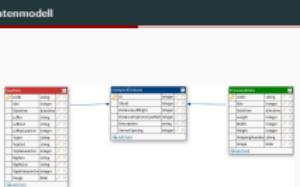
## Entwicklung von PaMesAn

### Entwurf

#### Datenmodell

2023-01-09

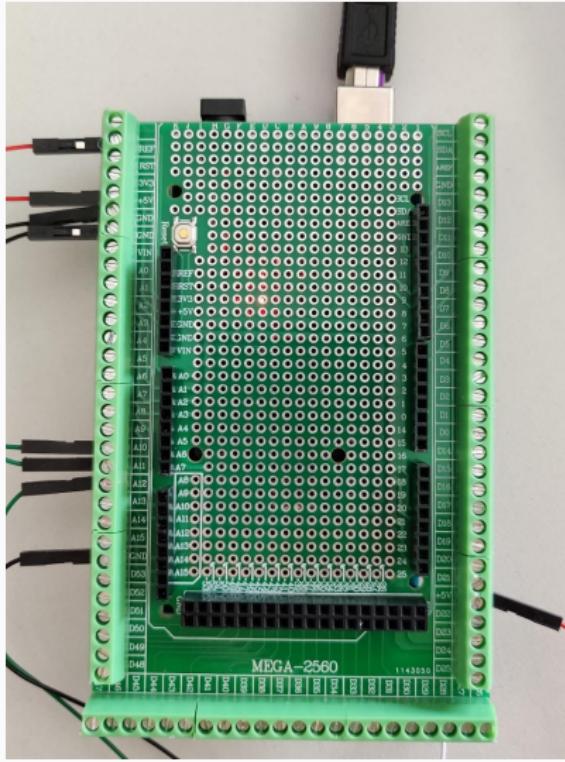
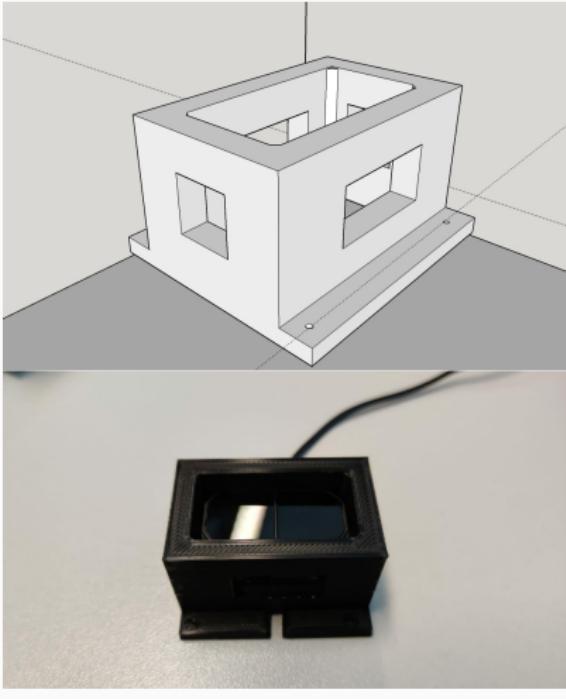
- Tabellenmodell
- 3 Tabellen
- MSSQL
- SiteSpecificValues-Table, SensorSpacing



# Implementierung

---

# Lasersensor und Arduino



Entwicklung von PaMesAn

└ Implementierung

└ Lasersensor und Arduino

2023-01-09

- Implementierung
- Lasersensor
- 3D Druck
- Arduino mit Shield, gute und sichere Verkabelung
- Script des Herstellers zum Auslesen der Sensoren

Lasersensor und Arduino



# Umsetzung des Sensorträgers



Entwicklung von PaMesAn

└ Implementierung

└ Umsetzung des Sensorträgers

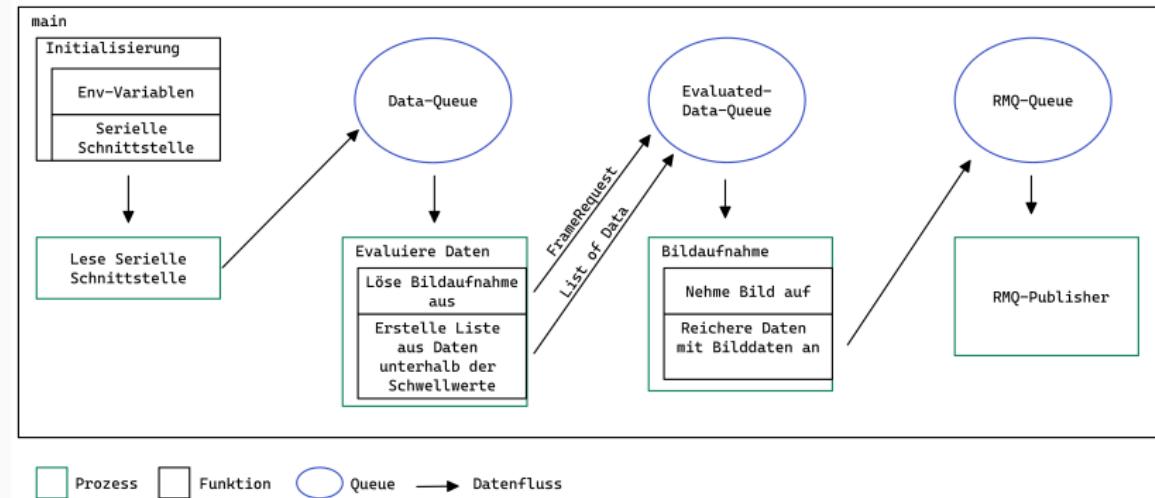
2023-01-09

- Haustechnik baut Sensorträger
- Kamera leicht nach hinten versetzt
- Lasersensor
- Arduino-Box und ThinClient hinter Foerderband

Umsetzung des Sensorträgers



# Implementierung des Ausleseservice



## Entwicklung von PaMesAn

### Implementierung

#### Implementierung des Ausleseservice

2023-01-09

Implementierung des Ausleseservice



- Python
- Multiprocessing anstatt Multithreading wegen Global Interpreter Lock
- 4 parallele Prozesse
- Initialisierung
- Serienller Prozess
- Evaluierungs Prozess, Schwellwert unterschreiten
- Bildaufnahme Prozess
- RabbitMQ-Publisher Prozess

# 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:
```

⋮

## Entwicklung von PaMesAn

### └ Implementierung

#### └ Implementierung des Ausleseservice – Pika

2023-01-09

- Beispiel RMQ-Publisher
- Initialisierung
- Verbindungsdaten ueber Umgebungsvariablen
- Verbindung ueber Programmbibliothek Pika
- While schleife mit 500ms Abfrage

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             time.sleep(0.5)
88         except Exception as ex:
89             ...
```

# 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     [Image] varbinary(MAX) NULL,
10    CONSTRAINT PK_RawData PRIMARY KEY (UUID),
11    CONSTRAINT RawData_FK FOREIGN KEY (Site)
12    REFERENCES PAMESAN.dbo.SiteSpecificValues(Id)
13
14 );
```

## Entwicklung von PaMesAn

### └ Implementierung

#### └ Implementierung der Datenbanktabellen

2023-01-09

- Tabellen zum Speichern
- SQL
- Auszug fuer rawData

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     [Image] varbinary(MAX) NULL,
9     CONSTRAINT PK_RawData PRIMARY KEY (UUID),
10    CONSTRAINT RawData_FK FOREIGN KEY (Site)
11    REFERENCES PAMESAN.dbo.SiteSpecificValues(Id)
12 );
13
```

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

2

### └ Implementierung

#### └ Implementierung des Datenspeicherservice

2023-01-09

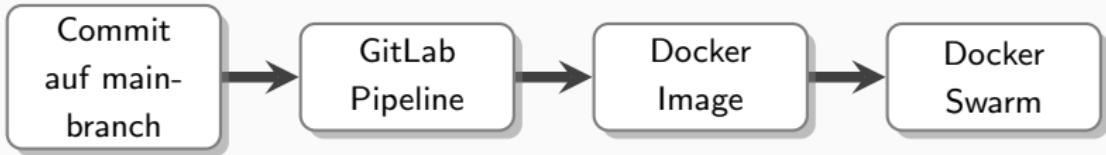
1. C#
2. Database first Ansatz
3. EntityFrameworkCore

```
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
```

2

# 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
- 2



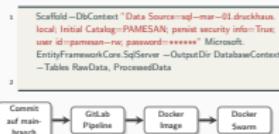
## Entwicklung von PaMesAn

### └ Implementierung

#### └ Implementierung des Datenspeicherservice

2023-01-09

Implementierung des Datenspeicherservice



1. C#
2. Database first Ansatz
3. EntityFrameworkCore
4. Deployment Prozess

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

## Entwicklung von PaMesAn

### └ Implementierung

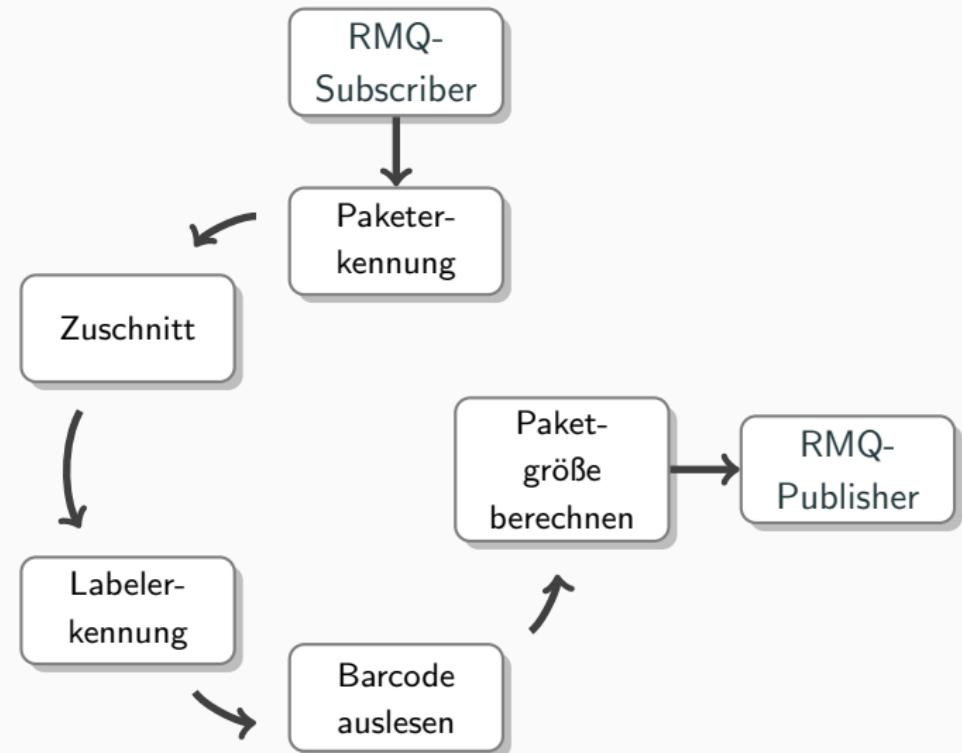
#### └ Implementierung des Datenspeicherservice

2023-01-09

- Code Beispiel fuer rawData
- Dependency Injektion Datenbankkontext
- Deserialisierung des JSON-Objekts
- Speichern in Datenbank
- Fehler zur Aufrufenden Methode durchreichen

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

# Implementierung des Datenverarbeitungsservice



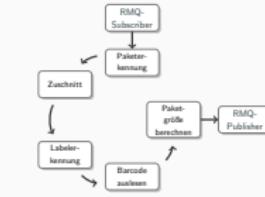
## Entwicklung von PaMesAn

### Implementierung

#### Implementierung des Datenverarbeitungsservice

2023-01-09

- Datenverarbeitungsservice in Python
- RMQ-Subscriber fragen Daten ab
- Paketerkennung
- Erkannte Paket zuschneiden
- Labelerkennung mit OpenCV
- Barcode auslesen mit pyzbar
- Paketgröße berechnen
- Mit RabbitMQ publishen



# Problemstellung Rollen und YOLOv7



Entwicklung von PaMesAn

└ Implementierung

└ Problemstellung Rollen und YOLOv7

2023-01-09

1. OpenCV Problem
2. Spiegelende Rollen
3. Lösung: Machine Learning





## YOLOv7

- Fully Convolutional Neural Network

Entwicklung von PaMesAn

└ Implementierung

└ Problemstellung Rollen und YOLOv7

2023-01-09

1. OpenCV Problem
2. Spiegelende Rollen
3. Lösung: Machine Learning
4. YOLOv7: You Only Look Once, Objektdetektor
5. FCNN: Neuronales Netzwerk, das nur aus Filtern besteht





## YOLOv7

- Fully Convolutional Neural Network
- Erkennung von Objekten durch Semantische Segmentierung

## Entwicklung von PaMesAn

### └ Implementierung

#### └ Problemstellung Rollen und YOLOv7

2023-01-09

1. OpenCV Problem
2. Spiegelende Rollen
3. Lösung: Machine Learning
4. YOLOv7: You Only Look Once, Objektdetektor
5. FCNN: Neuronales Netzwerk, das nur aus Filtern besteht
6. Suche nach Bedeutung in Bildern, pixelweise



YOLOv7  
• Fully Convolutional Neural Network  
• Erkennung von Objekten durch Semantische Segmentierung



## YOLOv7

- Fully Convolutional Neural Network
- Erkennung von Objekten durch Semantische Segmentierung
- Leichtes Training durch Bag of Freebies und Transfer Learning

## Entwicklung von PaMesAn

### └ Implementierung

#### └ Problemstellung Rollen und YOLOv7

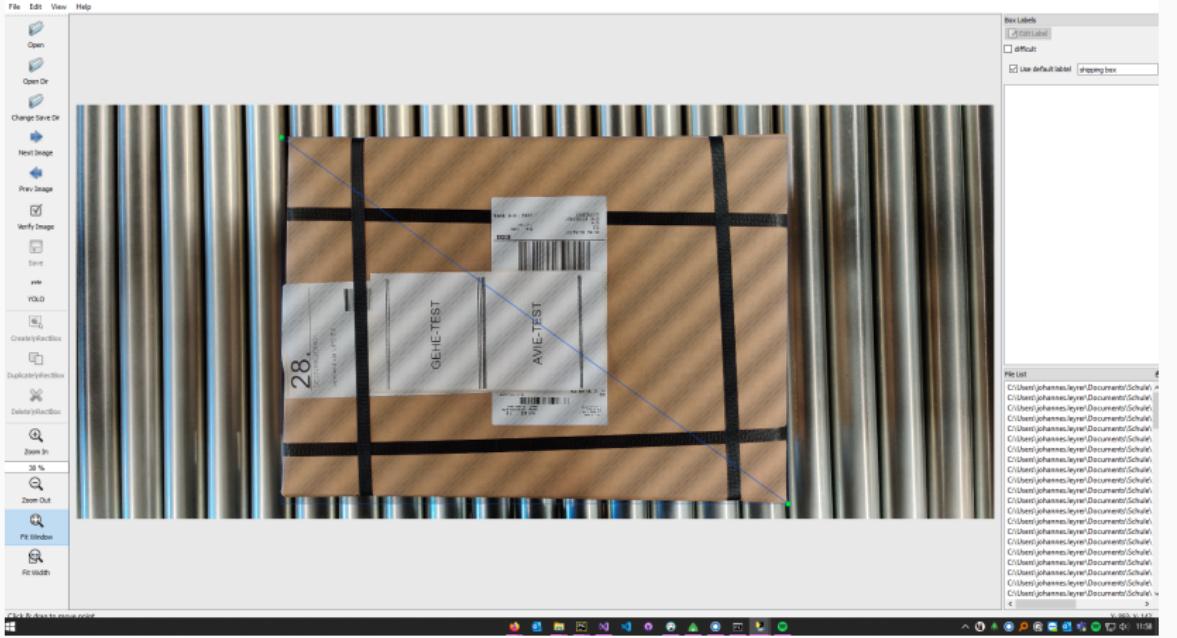
2023-01-09

1. OpenCV Problem
2. Spiegelende Rollen
3. Lösung: Machine Learning
4. YOLOv7: You Only Look Once, Objektdetektor
5. FCNN: Neuronales Netzwerk, das nur aus Filtern besteht
6. Suche nach Bedeutung in Bildern, pixelweise
7. Bag of Freebies: Methoden, die die Performance eines Modells erhöhen, ohne die Trainingszeit zu verlängern



YOLOv7  
• Fully Convolutional Neural Network  
• Erkennung von Objekten durch Semantische Segmentierung  
• Leichtes Training durch Bag of Freebies und Transfer Learning

# Labeln der Bilder



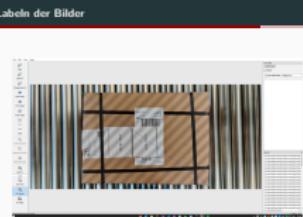
## Entwicklung von PaMesAn

### └ Implementierung

#### └ Labeln der Bilder

2023-01-09

- Labeln der Bilder
- LabellImg
- 100 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

## Entwicklung von PaMesAn

### └ Implementierung

#### └ Training des YOLOv7-Models

2023-01-09

```
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

- Datensatz in 80% Trainingsdaten und 20% Validierungsdaten aufteilen
- Traingsbefehl hier zu sehen
- batch size wie viele Bilder gleichzeitig
- 100 Durchgaengen (-epochs) vor und zurueck

# Ergebnis des YOLOv7-Models



Entwicklung von PaMesAn

└ Implementierung

└ Ergebnis des YOLOv7-Models

2023-01-09

- Ergebnis YOLOv7
- Model 94% sicher, dass es ein Versandkarton ist
- Falls Zeit ist, MiniDemo nach der Praesentation



# Auslesen des Versandlabels und Barcodes



Entwicklung von PaMesAn

└ Implementierung

└ Auslesen des Versandlabels und Barcodes

2023-01-09

Auslesen des Versandlabels und Barcodes



- Versandlabelerkennung mit OpenCV
- Barcodeerkennung mit pyzbar

# Berechnung der Paketgröße

```
8 def calculate_height(top_bottom: int, top_cm: int) ->
   int:
9     return top_bottom - top_cm
10
11 def calculate_width(left_right: int, left_cm: int,
12                     right_cm) -> int:
13     return left_right - left_cm - right_cm
14
15 def calculate_length(sensor_spacing: float, right_in:
16                      int, left_in: int, left_out: int) -> int:
17     speed = sensor_spacing / (right_in - left_in)
18     length = int((left_out - left_in) * speed)
19
20     return length
```

## Entwicklung von PaMesAn

### └ Implementierung

2023-01-09

### └ Berechnung der Paketgröße

- Hoehe: Bekannter Abstand Sensor–Foerderband minus gemessener Abstand
- Breite: Bekannter Abstand der Sensoren minus gemessener Abstand
- Laenge: Berechnung durch Geschwindigkeit und bekanntem Versatz der seitlichen Sensoren

Berechnung der Paketgröße

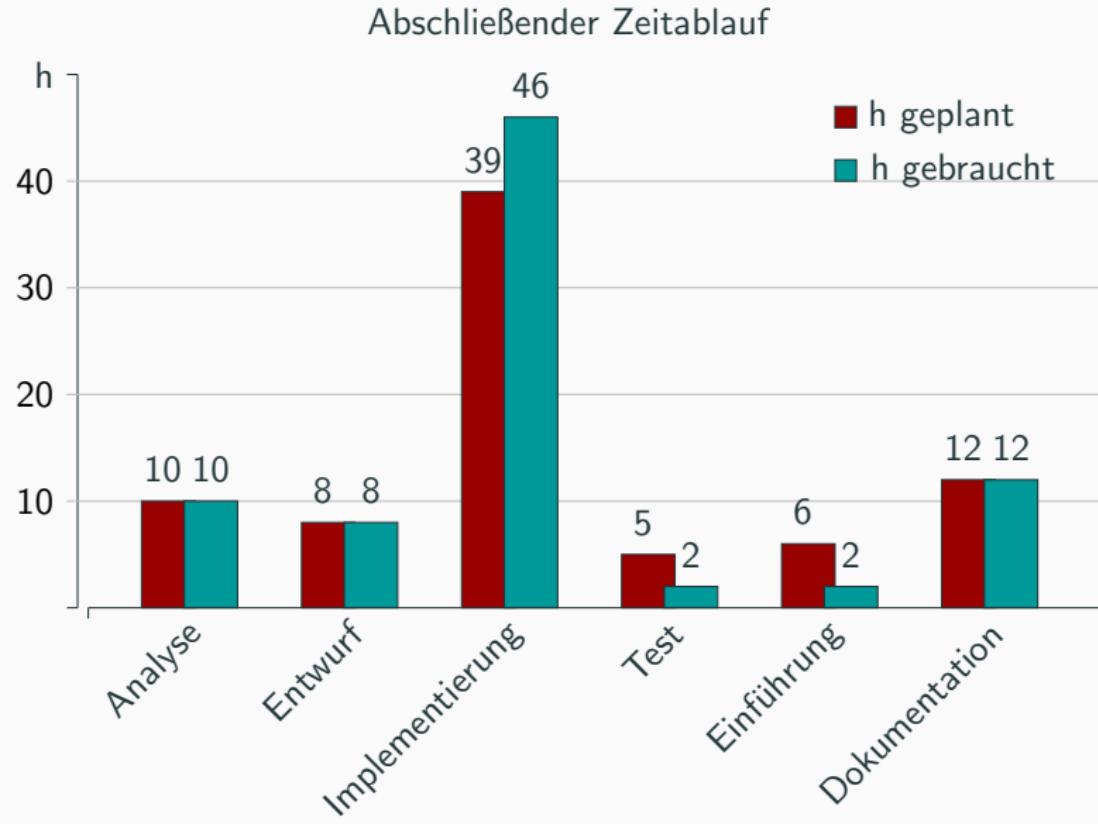
```
8 def calculate_height(top_bottom: int, top_cm: int) ->
   int:
9     return top_bottom - top_cm
10
11 def calculate_width(left_right: int, left_cm: int,
12                     right_cm) -> int:
13     return left_right - left_cm - right_cm
14
15 def calculate_length(sensor_spacing: float, right_in:
16                      int, left_in: int, left_out: int) -> int:
17     speed = sensor_spacing / (right_in - left_in)
18     length = int((left_out - left_in) * speed)
19
20     return length
```

2023-01-09

## Fazit

---

# Abschließender Zeitablauf



## Entwicklung von PaMesAn

### Fazit

#### Abschließender Zeitablauf

- Abschließender Zeitablauf
- Anfang gut
- Implementierung durch Paketerkennung verzoegert
- Test und Einführung stark abgeschwaecht
- Dokumentation im Zeitplan



## Lessons Learned

- YOLOv7 ist ein mit geringem Vorwissen leicht einsetzbarer Objektdetektor

## Entwicklung von PaMesAn

### Fazit

#### Lessons Learned und Ausblick

2023-01-09

1. Was habe ich aus dem Projekt mitgenommen?
2. YOLOv7 privat genutzt und gut einsetzbar

### Lessons Learned

- YOLOv7 ist ein mit geringem Vorwissen leicht einsetzbarer Objektdetektor

## Lessons Learned

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

### Fazit

#### Lessons Learned und Ausblick

2023-01-09

#### Lessons Learned

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

## Lessons Learned

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

## Entwicklung von PaMesAn

### Fazit

### Lessons Learned und Ausblick

2023-01-09

1. Was habe ich aus dem Projekt mitgenommen?
2. YOLOv7 privat genutzt und gut einsetzbar
3. Mehr Pufferzeit
4. Budget besser setzen

#### Lessons Learned

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

## Lessons Learned

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

## Ausblick

- Verwendung KEYENCE-Scanner und Umschreiben auf C#

### Fazit

### Lessons Learned und Ausblick

2023-01-09

1. Was habe ich aus dem Projekt mitgenommen?
2. YOLOv7 privat genutzt und gut einsetzbar
3. Mehr Pufferzeit
4. Budget besser setzen
5. Ausblick bzw. hier und jetzt
6. KEYENCE, Umschreiben auf C#, Wegfall von YOLOv7

#### Lessons Learned

- YOLOv7 ist ein 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

- YOLOv7 ist ein 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

## Entwicklung von PaMesAn

### Fazit

### Lessons Learned und Ausblick

2023-01-09

Lessons Learned und Ausblick

#### Lessons Learned

- YOLOv7 ist ein 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

- YOLOv7 ist ein 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

## Entwicklung von PaMesAn

### Fazit

### Lessons Learned und Ausblick

2023-01-09

1. Was habe ich aus dem Projekt mitgenommen?
2. YOLOv7 privat genutzt und gut einsetzbar
3. Mehr Pufferzeit
4. Budget besser setzen
5. Ausblick bzw. hier und jetzt
6. KEYENCE, Umschreiben auf C#, Wegfall von YOLOv7
7. Bekannte Kartonage
8. Aufbau an anderen Standorten

#### Lessons Learned

- YOLOv7 ist 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?**

**Danke für die Aufmerksamkeit!**

## Entwicklung von PaMesAn

2023-01-09

Danke für die Aufmerksamkeit!