
# PyctureStream

Experts for distributed Image Processing

Imaginary Sales Pitch
by Holger Büch & Marcus Fixel

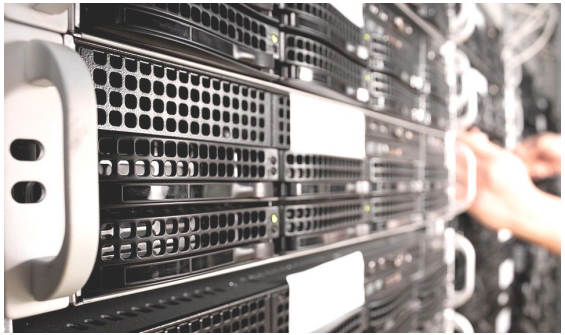
Master „Data Science and Business Analytics“
Hochschule der Medien, Stuttgart
Modul „BI- and Big-Data-Architectures“
Prof. Dr. Peer Küppers

AGENDA

# The next 30 Minutes

1. Introduction
2. Business Case
3. Designing the Architecture
4. Reference Architecture & Implementation
5. Demo of the Prototype
6. Wrap Up

# PyctureStream - Who are we?



## EXPERTS FOR BIG DATA

Two Founders & Consultancy-Team

Academical Research Projects

Experience in building
Real World Applications

## SPECIALIZED ON VISUAL DATA

Streaming Images or Videos

Applying advanced Analytics
like Deep Learning

Scalable distributed Architectures

## YOUR PARTNER FOR YOUR NEEDS

Project Planning

Requirements Engineering

Architecture &
Implementation

# What we can offer?

**ESCORTING YOUR BUSINESS**

We don’t leave the ship, when it’s set up. We help you sailing to success.

**CUSTOMIZATIONS FOR YOUR NEEDS**

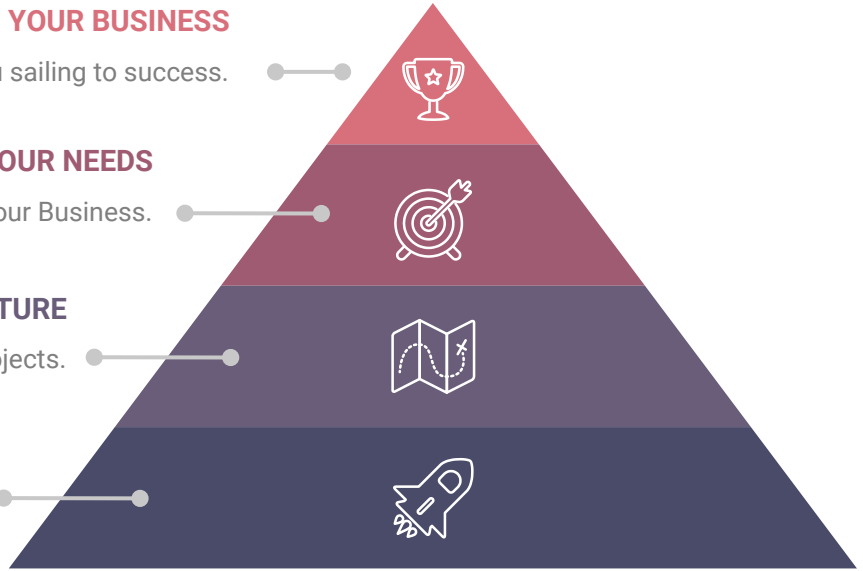
We identify your requirements and adept our Solution to your Business.

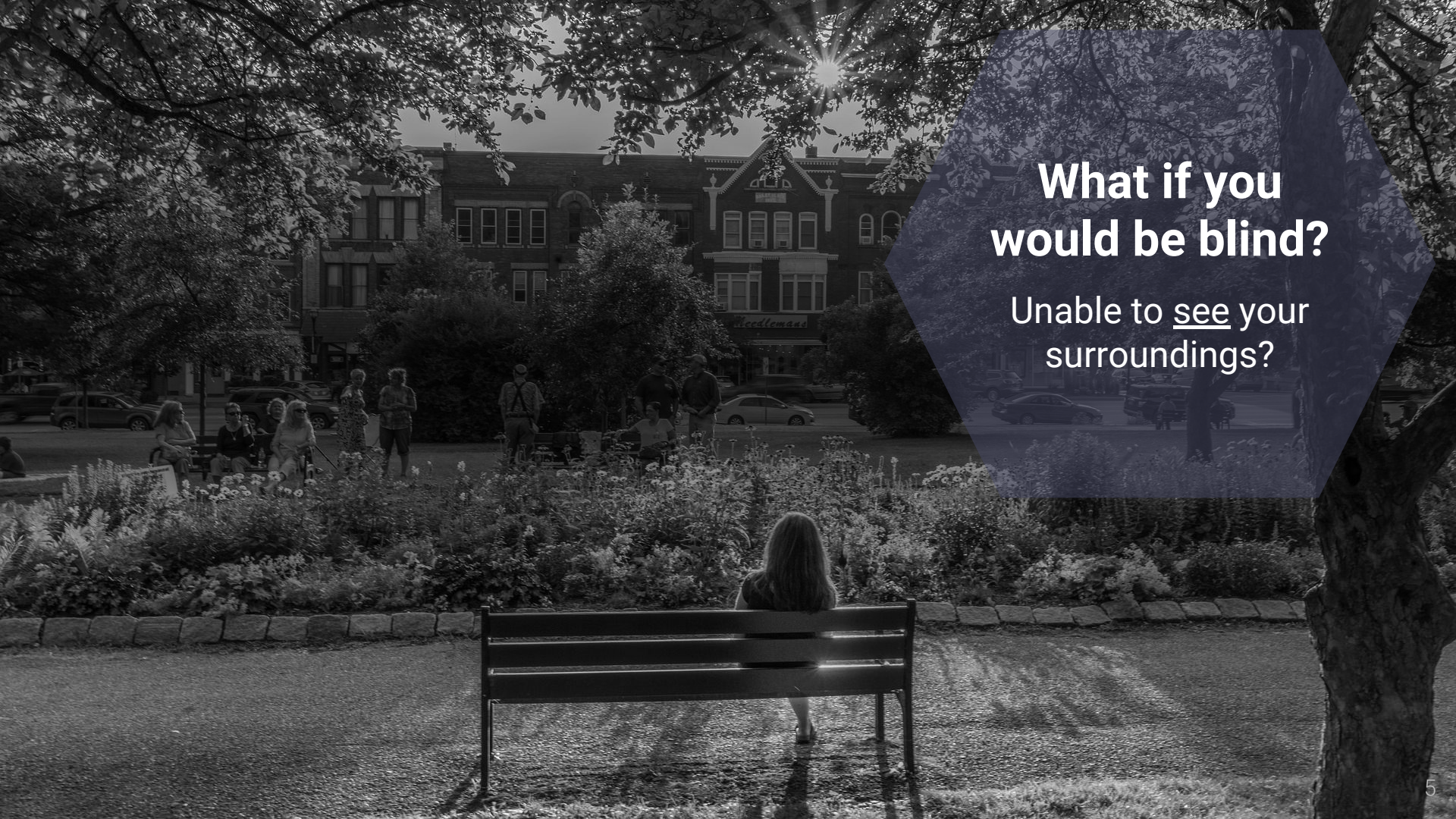
**REFERENCE ARCHITECTURE**

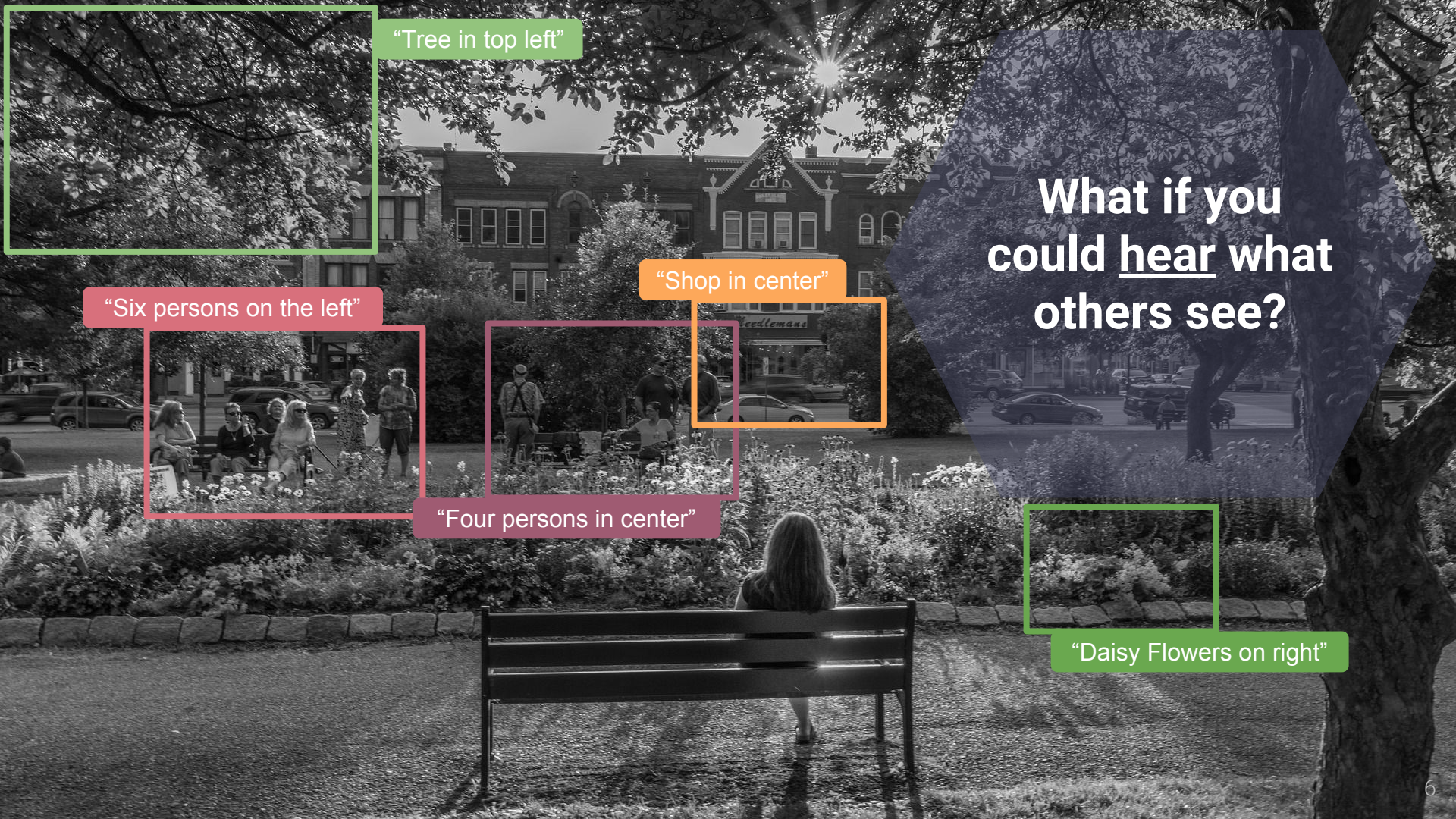
The Blueprint for your Big Data Architecture, proofen in multiple Projects.

**EXPERIENCED BIG DATA ARCHITECTS & DEVELOPERS**

We have Data Scientists and Data Engineers with the right Skill Set.




What if you would be blind?
Unable to see your surroundings?


“Tree in top left”
“Six persons on the left”
“Shop in center”
“Four persons in center”
“Daisy Flowers on right”
What if you could hear what others see?

# Business Model Canvas

## Key Partners

Sponsors

Technology Company

Health Insurance

Nonprofit Organizations

## Key Activities

Image Analysis

## Key Resources

Infrastructure

Development

Experts

## Value Proposition

More Information about Surroundings

More Independence

Fast & easy to use

Data Protection

## Customer Relations

Subscriptions

Technical Support

## Channels

Smartphone App

Online Media

## Customer Segment

Visually impaired People

Blind People

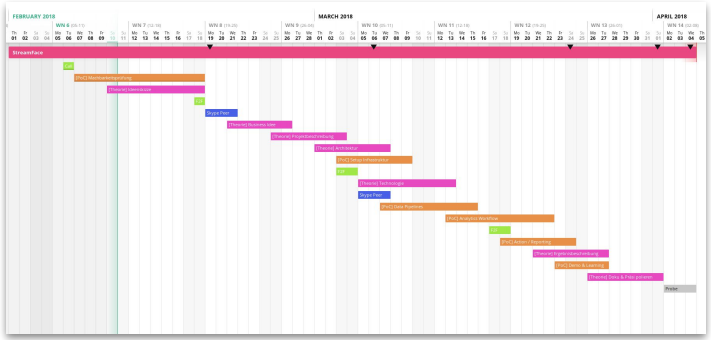
## Cost structure

Operation of Data Center
Development Costs
Sales and Marketing Costs

## Revenue Stream

Donations
Subscription Fees
Partners

# Project Planning




## Schedule-oriented Planning

e.g. using Gantt-Charts

## Activity-oriented Planning

e.g. using Work Breakdown Structures

**Pycture Stream**
**Bildverarbeitung in Micro Batches**

| 1.1 Kickoff | 1.2 Business Case | 1.3 Anforderungs-management | 1.4 User Experience | 1.5 Architektur | 1.6 Entwicklung | 1.7 Testing and Quality assurance | 1.8 Präsentation | 1.9 Projekt-bewertung |
|---|---|---|---|---|---|---|---|---|
| 1.1.1 Projekt erfassen | 1.2.1 Brainstorm | 1.3.1 Prozess-beschreibung | 1.4.1 User Interviews | 1.5.1 Information Flow | 1.5.1 Einrichtung der Infrastruktur | 1.6.1 Testen | 1.8.2 Dokumentation | 1.8.1 Controlling |
| 1.1.2 Projektplan erstellen | 1.2.2 Case Auswahl | 1.3.2 Anforderungen definieren | 1.4.2 User Interface | 1.5.2 Development View | 1.5.2 Setup Software Komponente | 1.6.2 Fehlerbehebung | 1.8.3 Präsentation erstellen | 1.8.1 Projektnach-besprechung |
| 1.1.3 Arbeitsablauf definieren | 1.2.3 Ausarbeitung Business Case | 1.3.3 FRD | 1.4.3 Usability Test | 1.5.3 Deployment View | 1.5.3 Implentierung | 1.6.3 Abnahme | 1.8.4 Präsentation | |
| 1.1.4 Teamrollen definieren | | | | | 1.5.4 Reporting | 1.6.4 Roll out | | |

# Business Requirements

**CAMERA-STREAM PER DAY**

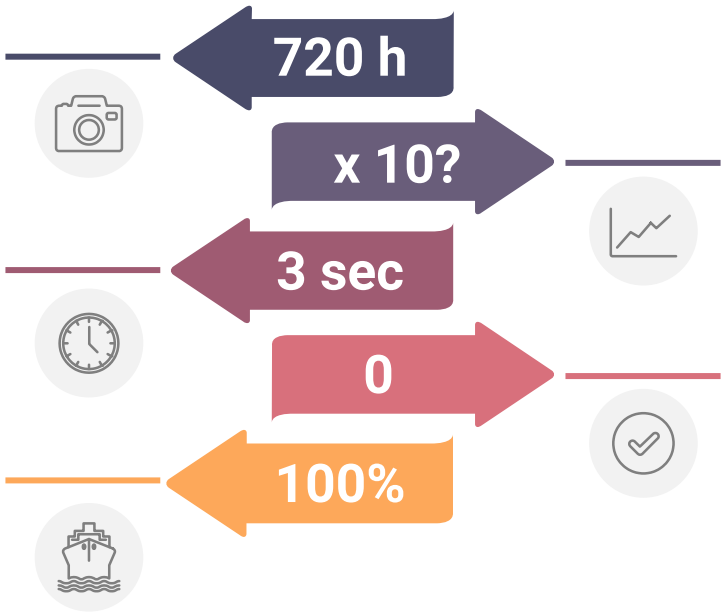
~4000 Customers using the App for 10 min per Day

**INSTANT RESULTS**

Timespan from taking the picture to hearing the results

**ENABLEMENT**

Enable the Company to drive the System and avoid Vendor Lock In.




**SCALABILITY**

Flexible Scaling in the first three Years, depending on Usage.

**PRIVACY CONCERNS**

No images or videos of customers should be used or stored. Ever.

# Technical Requirements


Business Requirements

&

Technical Expertise

Reference Architecture

# Basic Design Decisons

## Big Data Architecture?

- Volume, Velocity, Veracity, Variety

## Distributed Architecture?

- Flexible Scaling, Preconditions

# Choosing the Platform

## Cloud

78.000 $ in three Years

### PRO

- Leverage Autoscaling

### CON

- Expensive on larger Use-Cases
- Privacy hard to guarantee

### NEUTRAL

- Managable Risk of Vendor-Lock-In

## On-Premise

75.000 $ in three Years

### PRO

- Full Control

### CON

- More initial costs & effort
- Less flexible

### NEUTRAL

- Depends a lot on the Company

# Deciding on Architecture Model

## Lambda

PRO

- Enabling Re-Use of Images

CON

- More complex & effort
- Batch Layer not needed

NEUTRAL

- Enables Batch & Stream Approaches

## Kappa

PRO

- Focused on Streaming
- Single Technology Stack

CON

- Less widely used

NEUTRAL

- Skills for Streaming Analytics required

# Selecting the Software

## Kafka, Spark & TensorFlow

### PRO

- Widely known, used & supscriptported
- Very flexible
- Solid Choice

### NEUTRAL

- Lots of Features not needed in all Use-Cases

## Possible Alternatives

Streaming

- RabbitMQ, Apache ActiveMQ

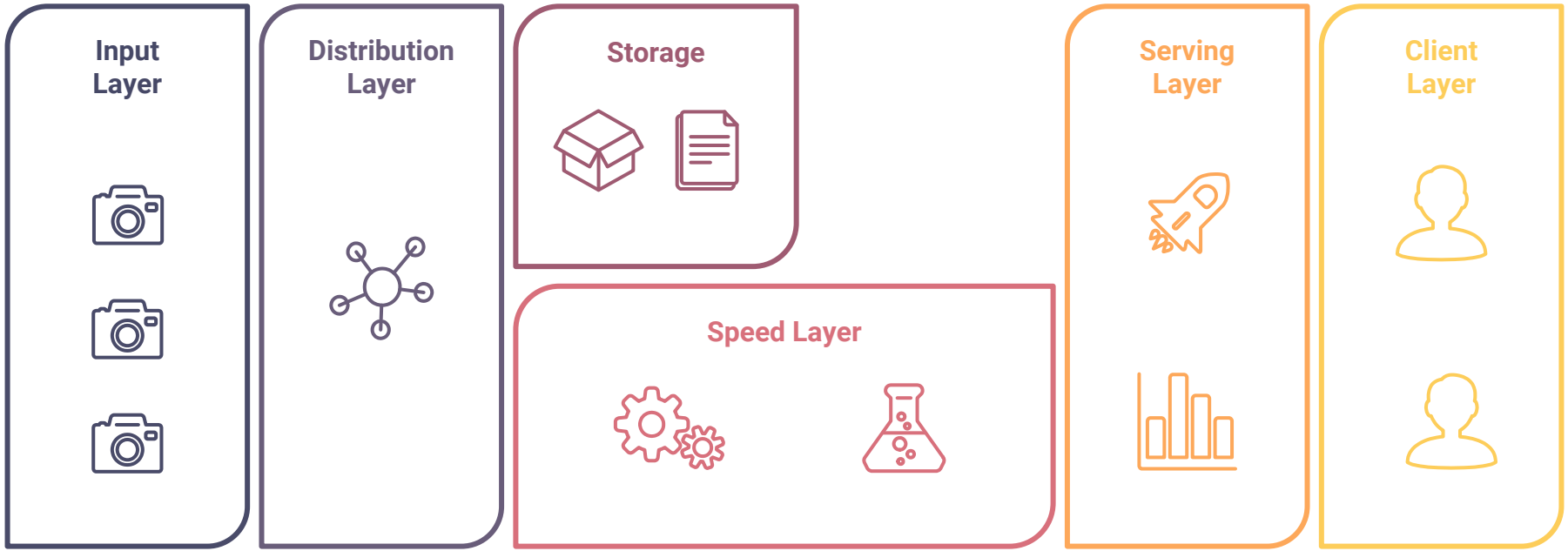
Processing

- Apache Storm

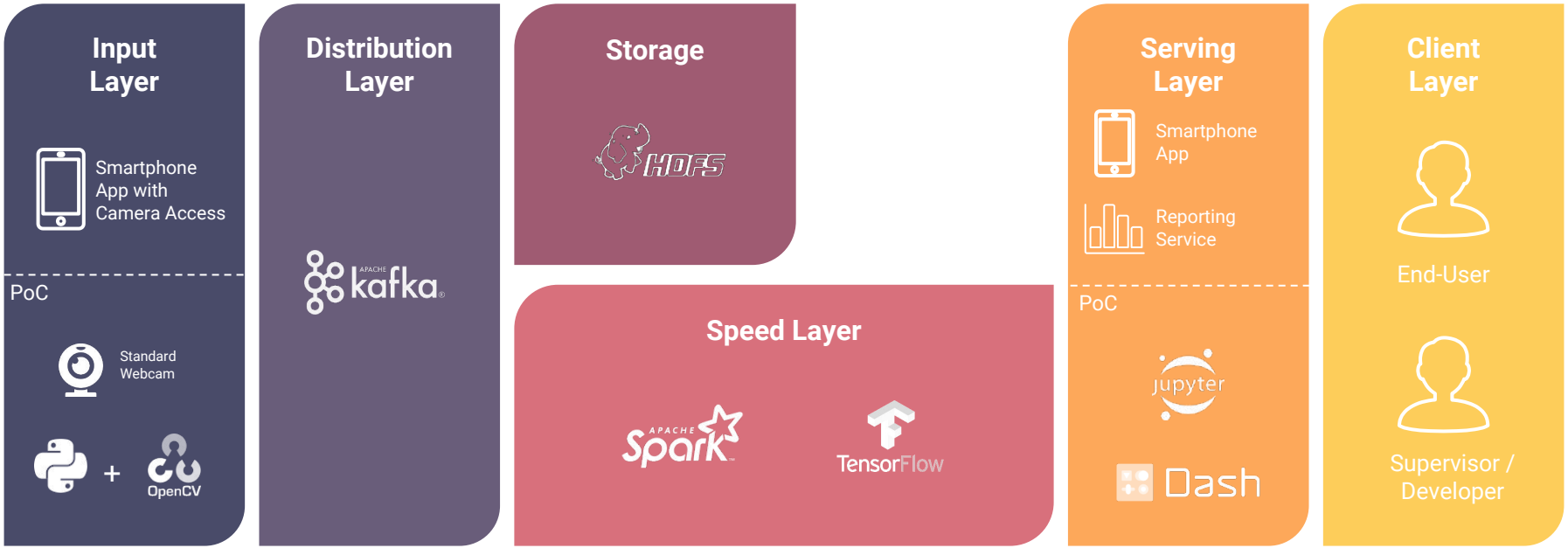
Analytics

- YOLO

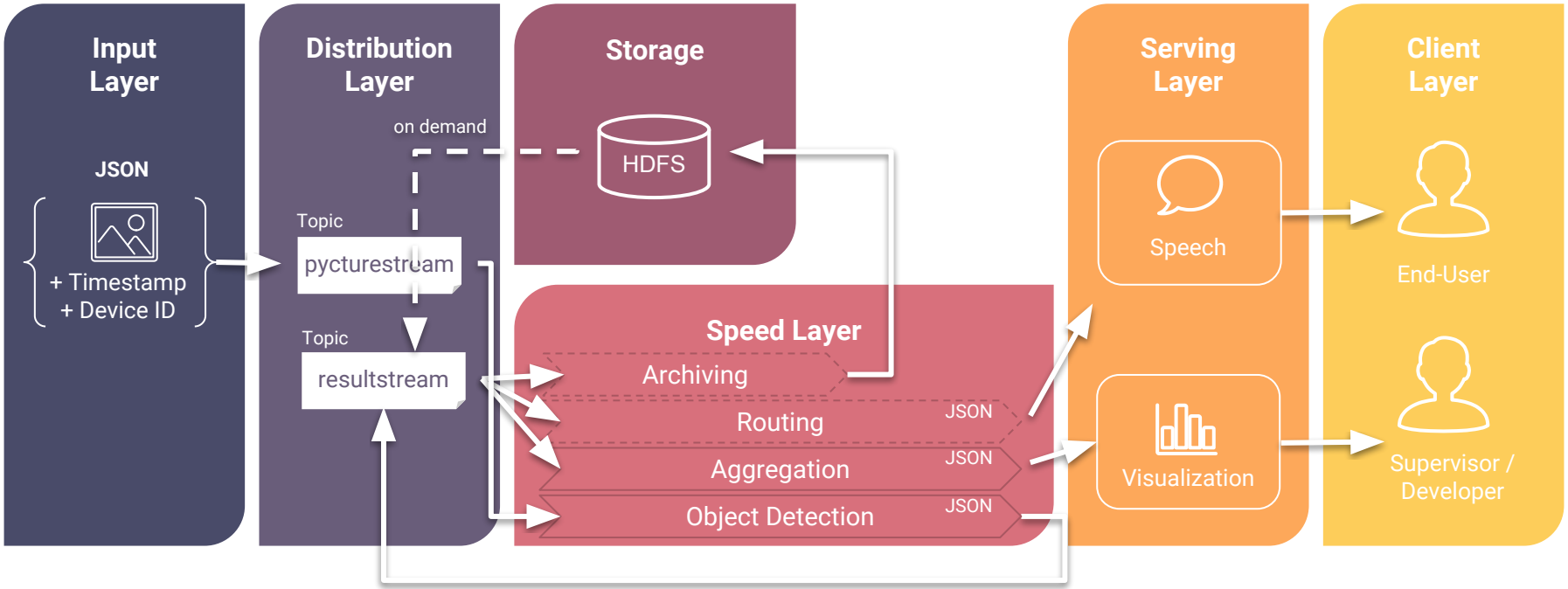
# Kappa Layer Model



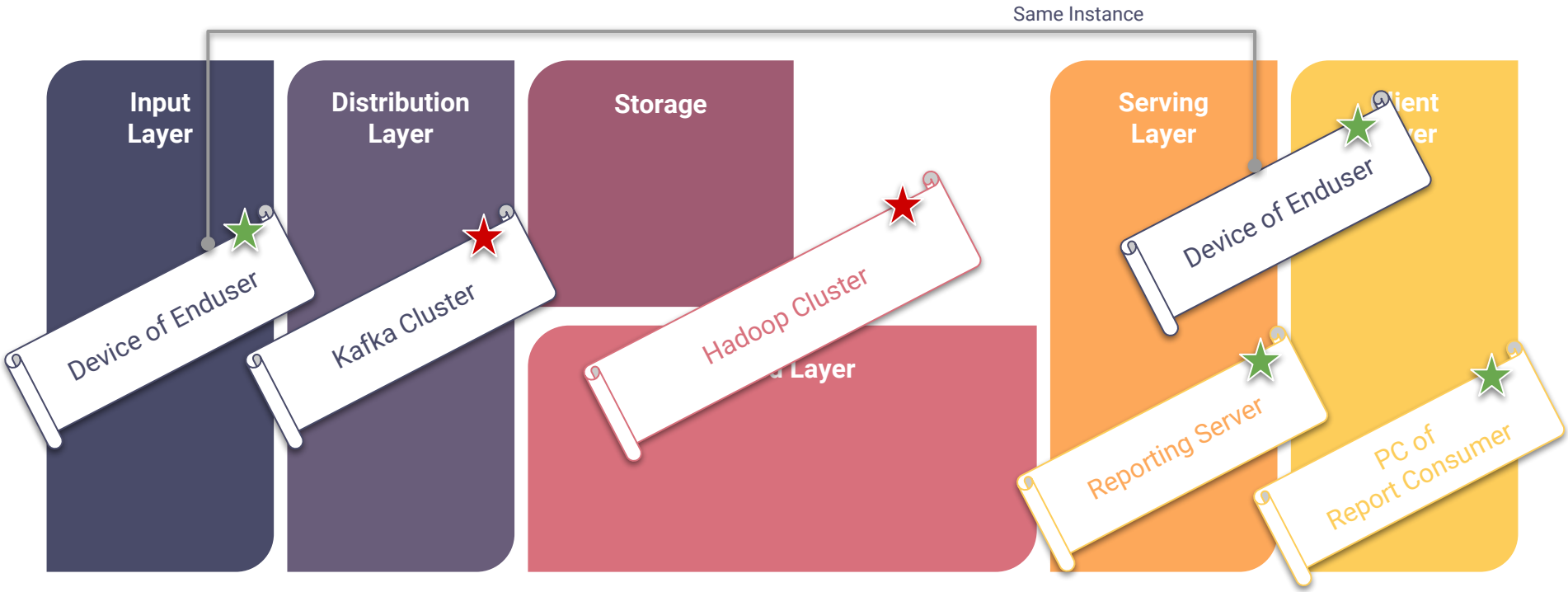
# Component View



# Information Flow



# Deployment View



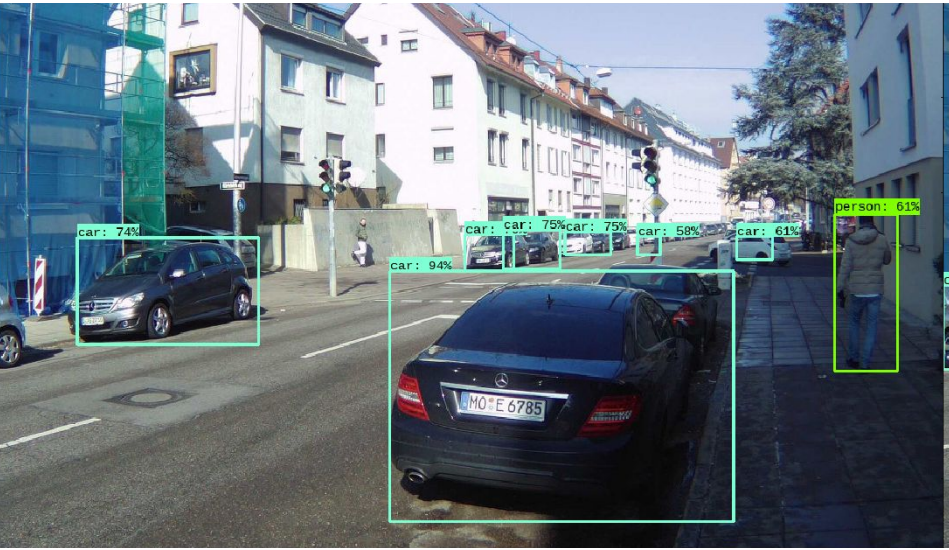
# Client-App

Using OpenCV to capture Webcam Image

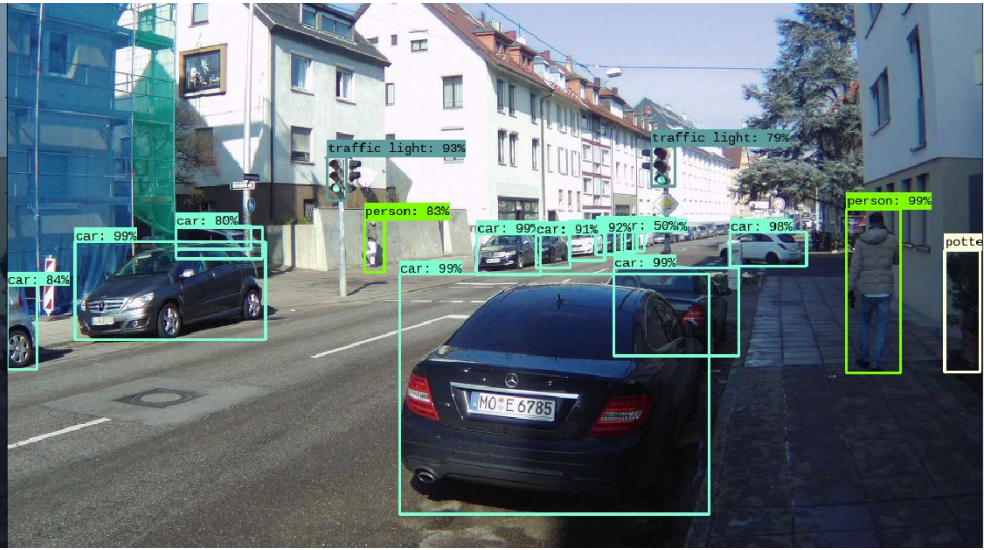
Serialize Image and send to Kafka using kafka-python

```
        logger.info('-'*50)
        self.stream_video()

    def stream_video(self):
        """Start streaming video frames to Kafka forever."""
        logger.info(f'Start capturing frames every {self.interval} sec.')
        while True:
            vidcap = cv2.VideoCapture(self.video_source)
            vidcap.set(3,1280)
            vidcap.set(4,720)
            success, image = vidcap.read()
            timestamp = dt.datetime.now().isoformat()
            vidcap.release()
            if success is True:
                # Base64 encode image for transfer in json
                jpg = cv2.imencode('.jpg', image)[1]
                jpg_as_text = base64.b64encode(jpg).decode('utf-8')
                # Build object and send to Kafka
                result = {
                    'image': jpg_as_text,
                    'timestamp': timestamp,
                    'camera_id': self.camera_id
                }
                self.send_to_kafka(result)
                self.save_image(image)
            else:
                logger.error(f'Could not read image from {self.video_source}!')
            # Sleep interval, before next capture
            time.sleep(self.interval)

    def send_to_kafka(self, data):
        """Send JSON payload to topic in Kafka."""
```

# Model Evaluation

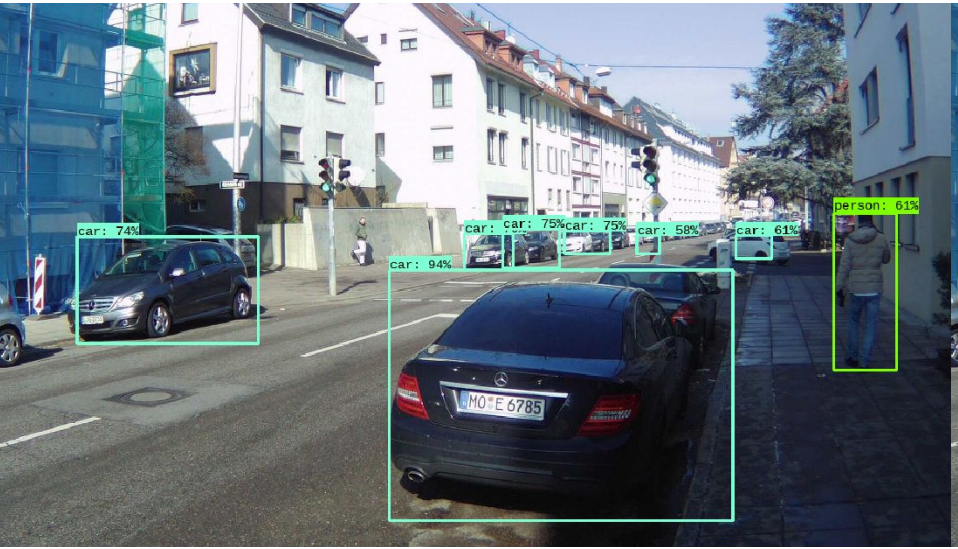



ssd_mobilenet_v1_coco (4.2 sec)

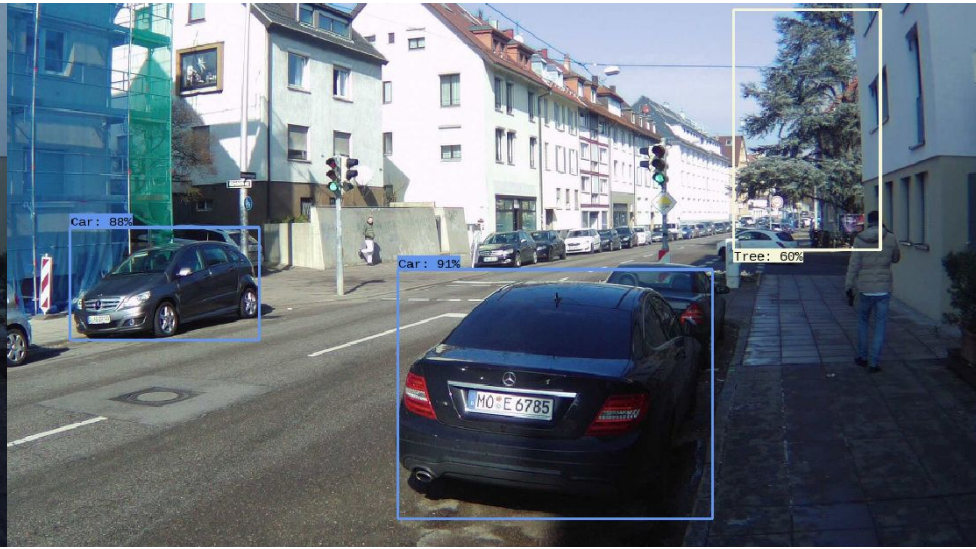



faster_rcnn_resnet101_coco (17.9 sec)

# Model Evaluation




ssd_mobilenet_v1_coco (4.2 sec)




faster_rcnn_inception_resnet_v2_atrous_oid (83.7 sec)

# Processing

Create SparkContext and broadcast Detection Model to Worker Nodes

Listen to Kafka Topic using Spark Streaming

Run TensorFlow with Object Detection API on Images

```
        # Load Spark Context
        sc = SparkContext(appName='PyctureStream')
        self.ssc = StreamingContext(sc, interval)  # , 3)

        # Load Frozen Network Model & Broadcast to Worker Nodes
        with tf.gfile.FastGFile(model_file, 'rb') as f:
            model_data = f.read()
        self.model_data_bc = sc.broadcast(model_data)

        # Load Graph Definition
        self.graph_def = tf.GraphDef()
        self.graph_def.ParseFromString(self.model_data_bc.value)

    def start_processing(self):
        """Start consuming from Kafka endpoint and detect objects."""
        kvs = KafkaUtils.createDirectStream(self.ssc,
                                            [self.topic_to_consume],
                                            {'metadata.broker.list': self.kafka
                                            )

        kvs.foreachRDD(self.handler)
        self.ssc.start()
        self.ssc.awaitTermination()
```

```
    def detect_objects(self, event):
        """Use TensorFlow Model to detect objects."""
        # Load Network Graph
        tf.import_graph_def(self.graph_def, name='')  # ~ 2.7 sec

        # Runs a tensor flow session
        with tf.Session() as sess:
            # Get handles to input and output tensors
            ops = tf.get_default_graph().get_operations()
```

# Demonstrating the Pipeline

# Considerations

**CLOUD vs. ON-PREMISE**

Tough decision. Has to be decided based on individual Circumstances.

**PROTOTYPE ONLY**

The Demo ran on a single Machine with lots of Overhead. Performance would scale!

**STREAMS ARE DIFFERENT**

Stream Processing needs a special Skill Set.
Like the one our Experts have.

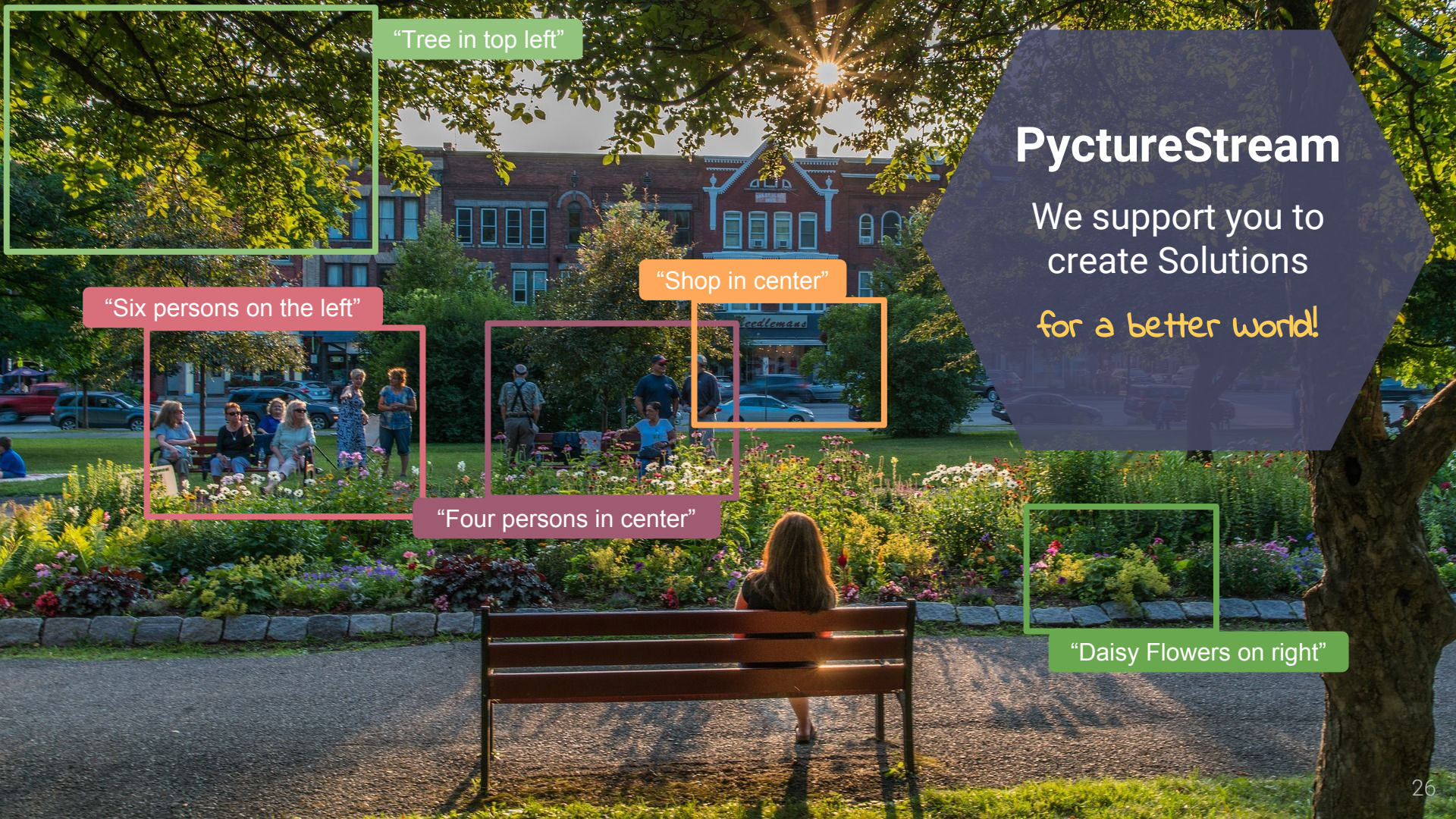


**END-TO-END**

We help you to implement a complete solution.
From Idea to shipped Product.

# PyctureStream

We support you to create Solutions


“Tree in top left”
“Six persons on the left”
“Shop in center”
“Four persons in center”
“Daisy Flowers on right”
PyctureStream
We support you to create Solutions
for a better world!

# Q & A

Please ask your Questions now! :-)

# References


**Aichele** C. (2006): Intelligentes Projektmanagement. Stuttgart: Kohlhammer Verlag.

**Be My Eyes** (2018): Be My Eyes – Hilfe für Blinde und Sehbehinderte. Android App im Google Play Store. URL: https://play.google.com/store/apps/details?id=com.bemyeyes.bemyeyes. Zugriff: 25.03.2018.

**Büch** H., *alias* dynobo (2018). PyctureStream – PoC for image processing using Kafka, Spark Streaming & TensorFlow. GitHub repository. URL: https://github.com/dynobo/PyctureStream. Zugriff: 26.02.2018.

**BMI** – Business Models Inc. (2018): The Business Model Canvas – Tool to help you understand a business model. URL: https://www.businessmodelsinc.com/tools-skills/tools/business-model- canvas. Zugriff: 25.03.2018.

**Cloudera** Inc. (2018). QuickStarts for CDH 5.12 – Virtualized clusters for easy installation on your desktop. URL: https://www.cloudera.com/downloads/quickstart_vms/5-12.html. Zugriff: 26.02.2018.

**Dell** Inc. (2017). Solution Overview – Dell EMC Ready Bundles for Hadoop. URL: https:// www.emc.com/collateral/solution-overview/ready-bundles-for-hadoop-solution-overview.pdf. Zugriff: 13.03.2018.

**Google** LLC (2018): Cloud Vision API – Quotas and Limits. URL: https://cloud.google.com/vision/quotas. Zugriff: 22.03.2018.

**Hunter** T. (2016): Deep Learning with Apache Spark and TensorFlow. URL: https://databricks.com/blog/2016/01/25/deep-learning-with-apache-spark-and-tensorflow.html. Zugriff: 28.02.2018.

**LeaseWeb** (2018): Private Rack Colocation – Set your business free with powerful private racks. URL: https://www.leaseweb.com/colocation/private-rack. Zugriff: 20.03.2018

**Pathirage** M. (2017): Kappa Architecture – Where everything is a stream. URL: http://www.kappa-architecture.com. Zugriff: 23.03.2018.

# References

**Mallot** H. A. (2006): Visuelle Wahrnehmung. In J. Funke & P. A. Frensch (Hrsg.): Handbuch der Allgemeinen Psychologie – Kognition. Göttingen: Hogrefe Verlag.

**Moffat** R. (2017): Getting Started with Spark Streaming, Python and Kafka. URL: https://www.rittmanmead.com/blog/2017/01/getting-started-with-spark-streaming-with-python-and-kafka/. Zugriff: 05.03.2018.

**Plotly** Inc. (2018): Dash by Plotly – Build beautiful web-based interfaces in Python. URL: https://plot.ly/products/dash/. Zugriff: 20.03.2018.

**Project Jupyter** (2018, 20. Februar): JupyterLab is Ready for Users. *Jupyter Blog – All the latest about Project Jupyter*. URL: https://blog.jupyter.org/jupyterlab-is-ready-for-users-5a6f039b8906. Zugriff: 25.02.2018.

**Oracle** Corp. (2018). Oracle VM VirtualBox. URL: https://www.virtualbox.org. Zugriff: 26.02.2018.

**TensorFlow** (2018a): Tensorflow Object Detection API. GitHub repository. URL: https://github.com/tensorflow/models/tree/master/research/object_detection. Zugriff: 26.02.2018.

**TensorFlow** (2018b): Tensorflow detection model zoo. GitHub repository. URL: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md. Zugriff: 26.02.2018.

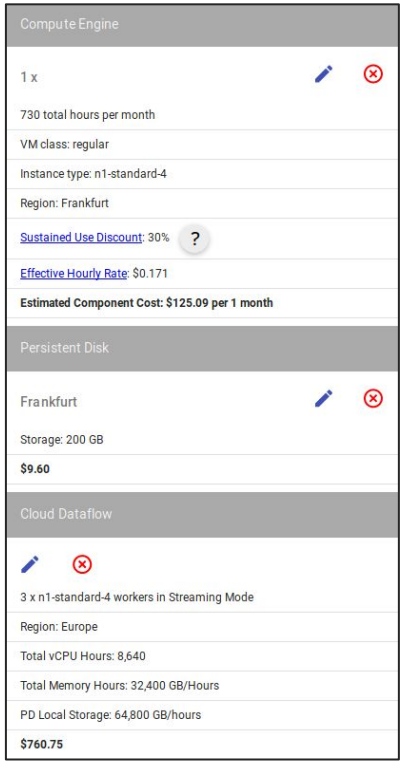
# Image Sources

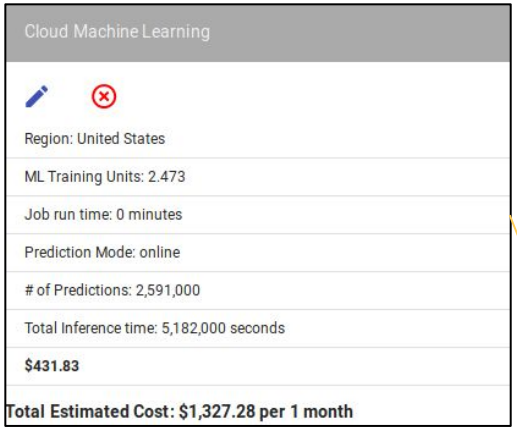
Photographs are from pixabay.com and licensed CC0
Icons are from SlidesCarnival.com and licensed CC BY 4.0
Charts & Screenshots are self-made

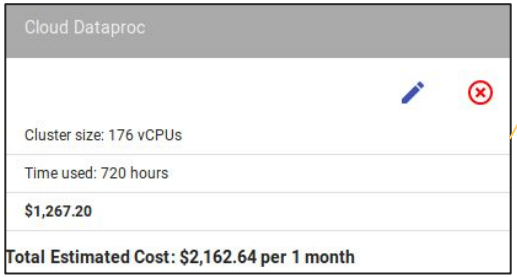
# BACKUP SLIDES

If enough is not enough...

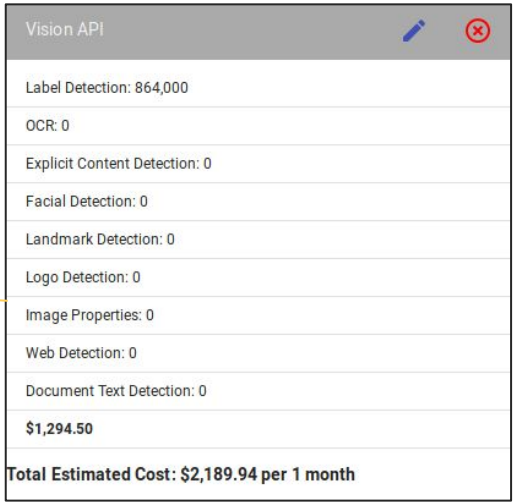
# Cost Calculation Cloud



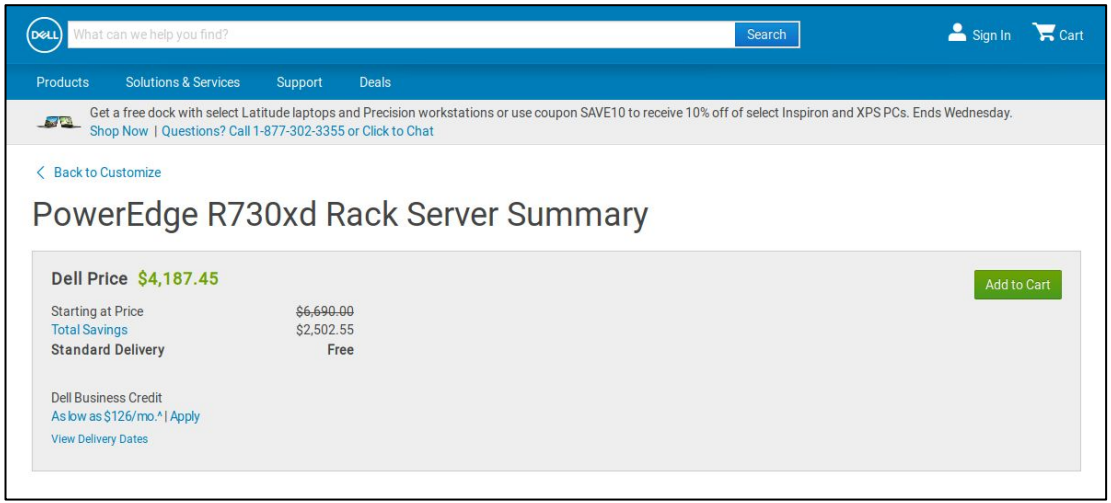




OR



# Cost Calculation On Premise




- 8 Server for Hadoop-Cluster
  (3 Name Nodes, 1 Edge Node, 4 Worker Nodes)
- 3 Server forKafka-Cluster
  (1 Zookeeper Node, 2 Broker Nodes)
- Reporting will run on Edge Node
- 700 $ Rent per month for Data Center
- No Licence Cost (Apache Distro)