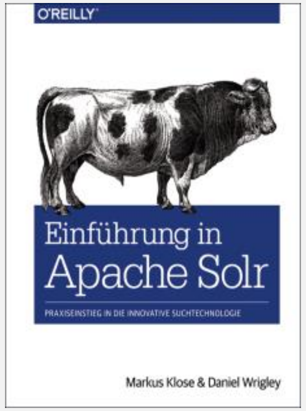
Actionable Insights with Real-time Streaming Analytics of Customer Reviews

Big Data Conference
Vilnius – November 28, 2019
Daniel Wrigley

Daniel Wrigley

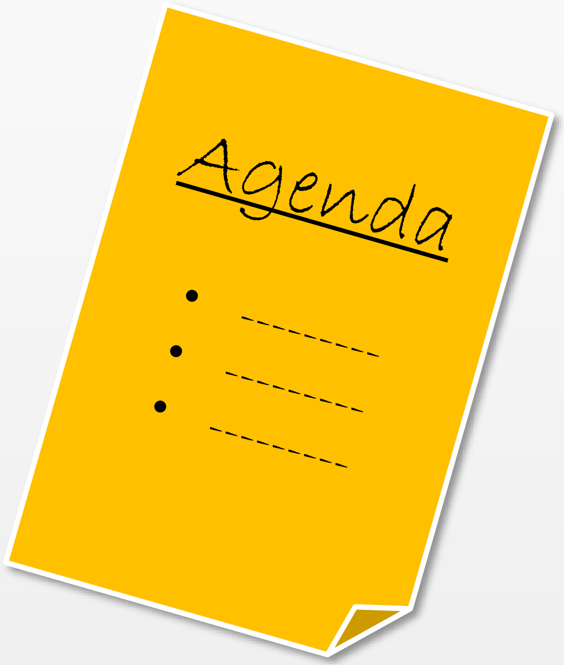
- Senior Consultant Search & Analytics
- Certified Apache Solr Trainer
- Author of „Einführung in Apache Solr“

- daniel.wrigley@shi-gmbh.com
- @wrigley_dan



Agenda

- Unstructured Data
- Requirements of a Real-time Streaming Analytics Platform
- Blueprint – Reference Architecture
- Open Source Software Components and their Value
- Demo Time!
- Recap
- Future Improvements



Unstructured Data

- Data is being created at any time in any company
- Huge amount is unstructured data: E-mails, reviews, documentation, presentations etc.
- Data is stored across many silos

→ Limited access and readability

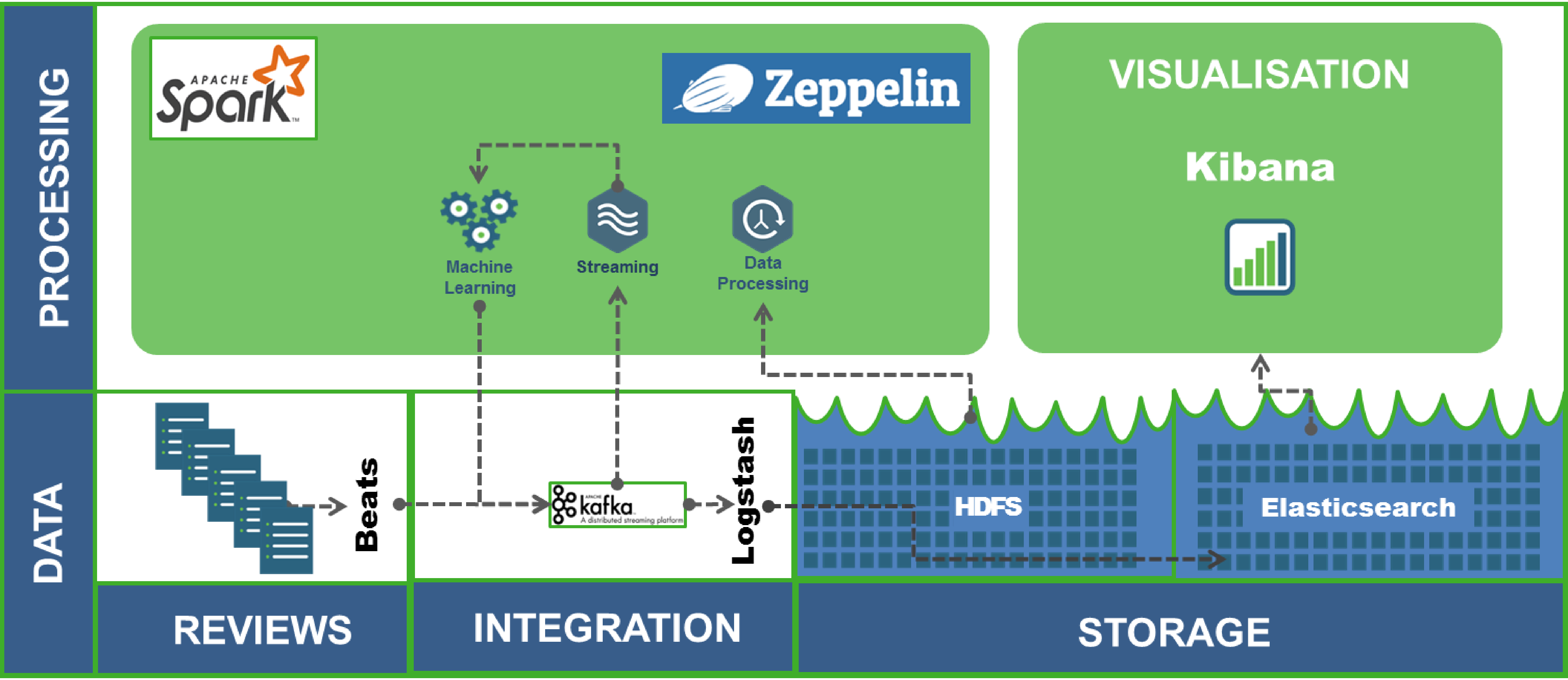
https://www.flickr.com/photos/pauldineen/4529213795/

SEARCH
ANALYTICS
BIG DATA

Requirements

- Performance:
  - Low Latency & High Throughput
- Manageability:
  - Easy Deployment
  - APIs & KPIs for Monitoring
- Scalability & Resiliency:
  - Start Small → Grow Tall
  - Resilient to Partial Outages
- Data Protection & Mulit-tenancy:
  - End-to-End Governance Process
  - Different Consumers with Different Data Usages
- Compatibility & Expansion:
  - Scale Across Multiple Data Centers
  - Implementation of New Use Cases & Extending Existing Ones

Blueprint – Reference Architecture



Data Transport – Why Beats?

- Easy to configure for simple data shipment to several systems: Elasticsearch, Logstash, Kafka, …
- Acts as an agent on servers
- Downside:
  - Does not scale well
  - No fault tolerance
- Alternative:
  - Apache NiFi/MiNiFi

- Shippers for all data
  - Filebeat for log files
  - Metricbeat for metrics
  - Packetbeat for network data
  - Winlogbeat for windows event logs
  - Auditbeat for audit data
  - Heartbeat for uptime monitoring
  - Functionbeat for serverless shipment of cloud service data

Data Integration – Why Kafka?

- Publish & subscribe streaming platform
- Seamless integration with other open source frameworks: Beats, Logstash, Spark, …
- Low latency & high scalability
- Independent of message structure

Data Integration – Why Logstash?

- Easy to configre data shipment
- Also connects to technologies beyond the Elastic related ones
- Downside:
  - Does not scale well
  - No fault tolerance
- Alternative
  - Apache NiFi
- Offers a variety of inputs
  - Beats
  - Redis
  - Kafka
  - …
- Filters parse and transform the data
- Stores data in multiple outputs
  - Elasticsearch
  - HDFS
  - Slack
  - …

SEARCH
ANALYTICS
BIG DATA

Data Storage – Why Elasticsearch?

- Blazingly fast NoSQL storage system with rich query DSL
- Scalable to tremendous amounts of data
- Obvious: Seemlessly connects to other Elastic products like Logstash, Beats & Kibana
- Management and monitoring of clusters via Kibana
- Client libraries for all relevant programming languages
- Rich RESTful APIs offer easy integration

Data Processing – Why Spark?

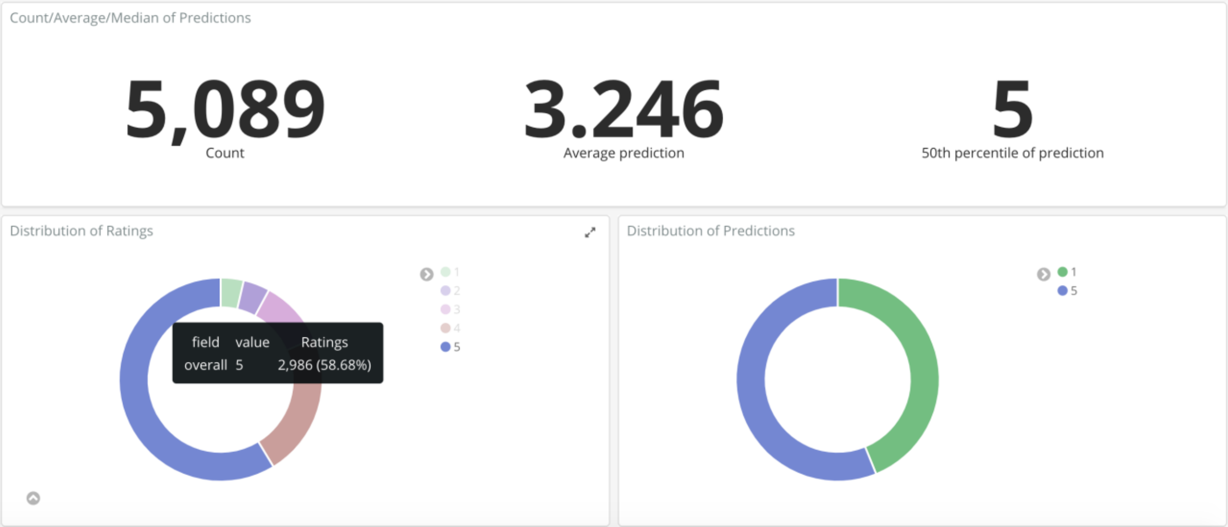
- Well, why not?
- General data processing framework with (structured) streaming and machine learning features
- Faster than traditional MapReduce options
- Scala, Python, Java, R support

Data Processing – Why Zeppelin?

- Collaborative data science platform with visualisation capabilites
- Excellent for „playing around“ with and analysing your data
- Full Spark support
- Other, more stable solutions for this purpose may exist (e.g. Jupyter)

Data Visualisation – Why Kibana?

- If you use Elasticsearch, you use Kibana
- Rich visualisation options for data stored in Elasticsearch
- Drill-down options
- Leverages fast querying capabilities of Elasticsearch
- Administration capabilites for Elasticsearch



Demo Setting – The Data

- 37,000+ Movie Reviews from Amazon
- Data Structure:

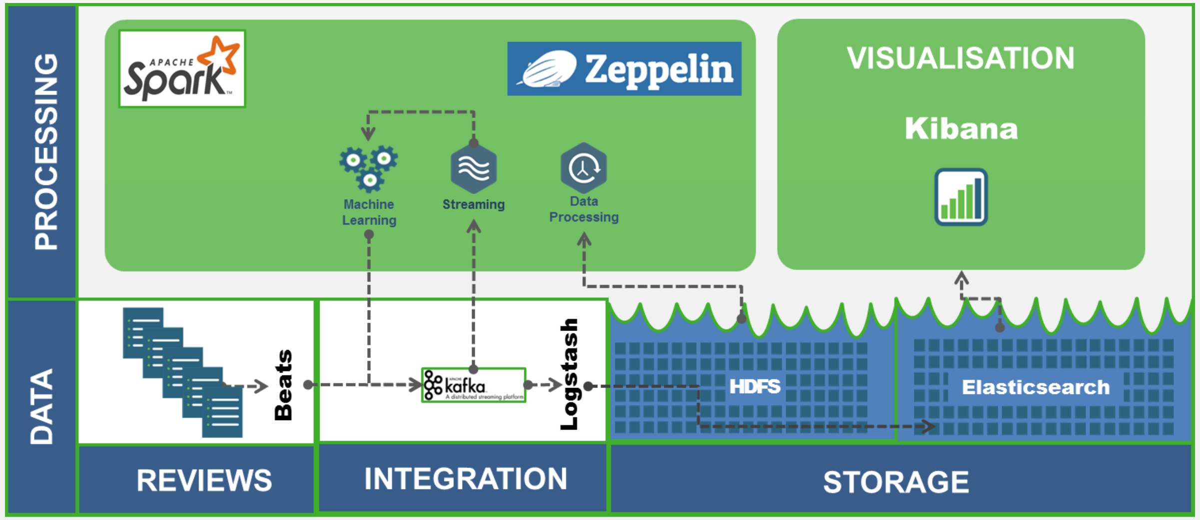
```
root
 |-- asin: string (nullable = true)
 |-- helpful: array (nullable = true)
 |    |-- element: long (containsNull = true)
 |-- overall: double (nullable = true)
 |-- reviewText: string (nullable = true)
 |-- reviewTime: string (nullable = true)
 |-- reviewerID: string (nullable = true)
 |-- reviewerName: string (nullable = true)
 |-- summary: string (nullable = true)
 |-- unixReviewTime: long (nullable = true)
```

- Machine Learning Part:
  - Simple ML model
    - Trained in Spark with 1000 „pure good“ (5*) and 1000 „pure bad“ (1*) reviews
    - Input → Tokenization → Model Computation
    - Based on TF/IDF

SEARCH
ANALYTICS
BIG DATA

Demo Setting – Workflow

- Simulation of a continuous data stream via a simple script
- Beats reads data from disk and sends it to a Kafka topic (*reviews*)
- Spark Streaming subscribes to Kafka topic *reviews*, classifies the data, writes the result to another Kafka topic (*reviews_analysed*)
- Logstash picks up data from Kafka topic (*reviews_analysed*), sends it to Elasticsearch
- Kibana visualises indexed data



Recap – Requirements met?

- Performance
  - Kafka, Spark, Elasticsearch as central components can handle an immense number of events per second
- Manageability
  - Containerisation possible
  - All components provide APIs
- Scalability & Resiliency
  - Cluster-mode
  - Data replication
- Data Protection & Mulit-tenancy
  - Topic-based architecture of Kafka offers multi-tenancy capabilities
  - Data lineage and provenance can be tracked (NiFi)
- Compatibility & Expansion
  - Implementing new use cases or extending existing ones is often just a question of scaling
  - Multi data center can be a challenge

Streaming Analytics Use Cases

- E-Commerce
  - Search result optimsation
  - Reward top reviewers
  - Reward „bad“ reviewers
  - Automatic filtering of reviews
- Non-E-Commerce
  - Fraud Detection
  - Real-time classification of transactions
  - Named Entity Recognition
  - Social Media Monitoring
  - Sensor Data Anomaly Detection

Possible Future Improvements

- Exclude (some) Stopwords from Reviews
  Careful: *very funny* & *not funny* become *funny*!
- Word Embeddings/word2vec
- Summary vs. Whole Review
- Quality Measurements
  - High Frequency Reviewers vs. One Time Reviewers
  - Review Length

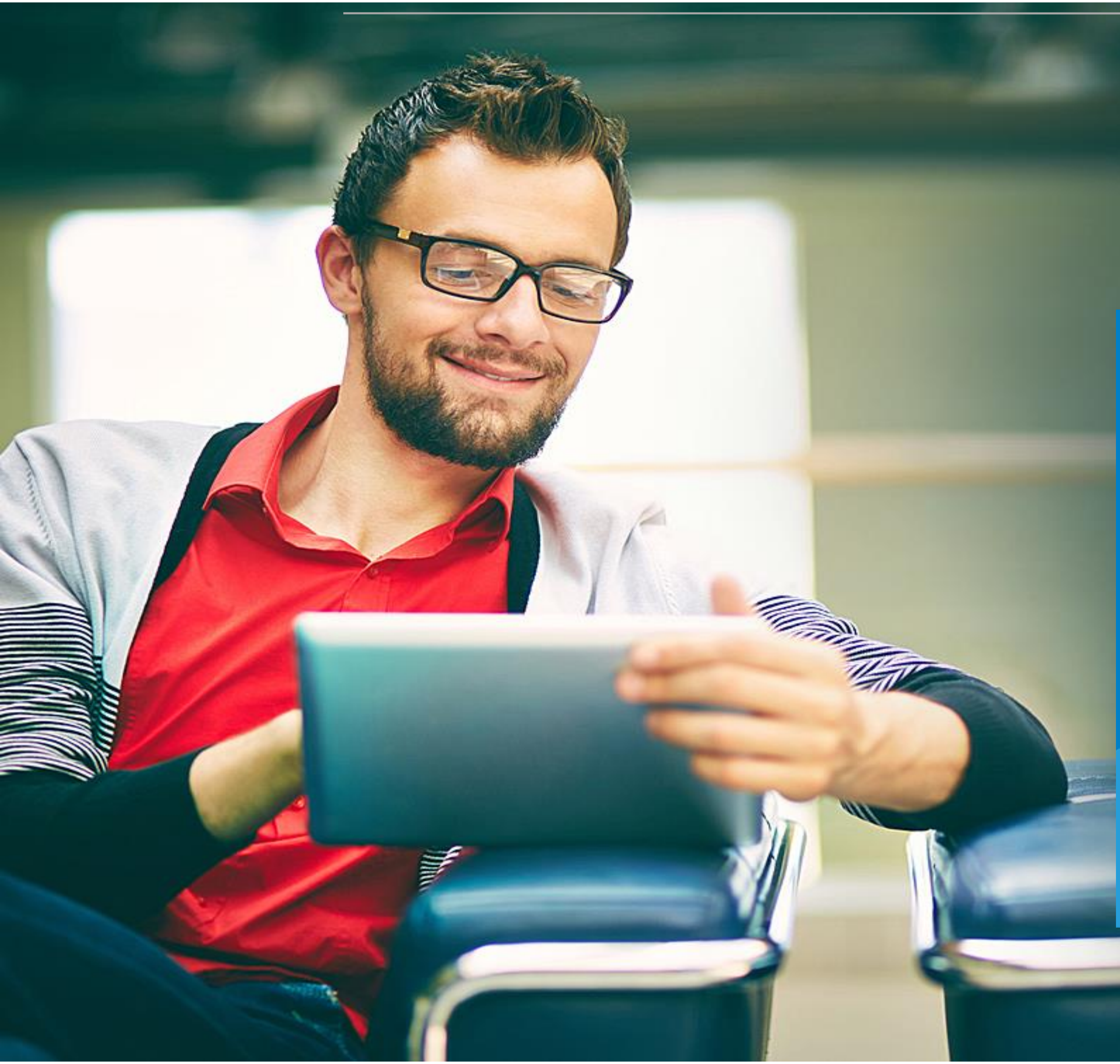
https://www.flickr.com/photos/157270154@N05/27175302347/

SEARCH ANALYTICS BIG DATA

References

- Amazon Reviews Data Reference:
  - R. He, J. McAuley. Modeling the visual evolution of fashion trends with one-class collaborative filtering. WWW, 2016
  - J. McAuley, C. Targett, J. Shi, A. van den Hengel. Image-based recommendations on styles and substitutes. SIGIR, 2015
- Streamlio – Evaluating Streaming Data Solutions: https://info.streaml.io/evaluating-streaming-solutions-webcast
- Photo Credits:
  - https://www.flickr.com/photos/pauldineen/4529213795/
  - https://www.flickr.com/photos/157270154@N05/27175302347/

    License: https://creativecommons.org/licenses/by/2.0/
- Elastic{ON}, Elasticsearch, Kibana, Logstash, Beats and Metricbeat are trademarks of Elasticsearch BV, registered in the U.S. and in other countries.



Contact Us

SHI GmbH
Konrad-Adenauer-Allee 15
D - 86150 Augsburg

info@shi-gmbh.com
+49 821 - 74 82 633 - 0
@SHIEngineers