Emerging Technologies for the Circular Economy

Lecture 6: IoT – Data Processing and BigData

Prof. Dr. Benjamin Leiding (Clausthal)
M.Sc. Arne Bochem (Göttingen)
M.Sc. Anant Sujatanagarjuna (Clausthal)

License

- This work is licensed under a **Creative Commons Attribution-ShareAlike 4.0 International License**. To view a copy of this license, please refer to https://creativecommons.org/licenses/by-sa/4.0/ .

- Updated versions of these slides will be available in our Github repository.

Examination Info

Göttingen

- Oral examination
- 15-20min Q&A
- Registration until 25.07.2022
- Dates:
  - 01.08.2022 → Afternoon
  - 03.08.2022 → All day

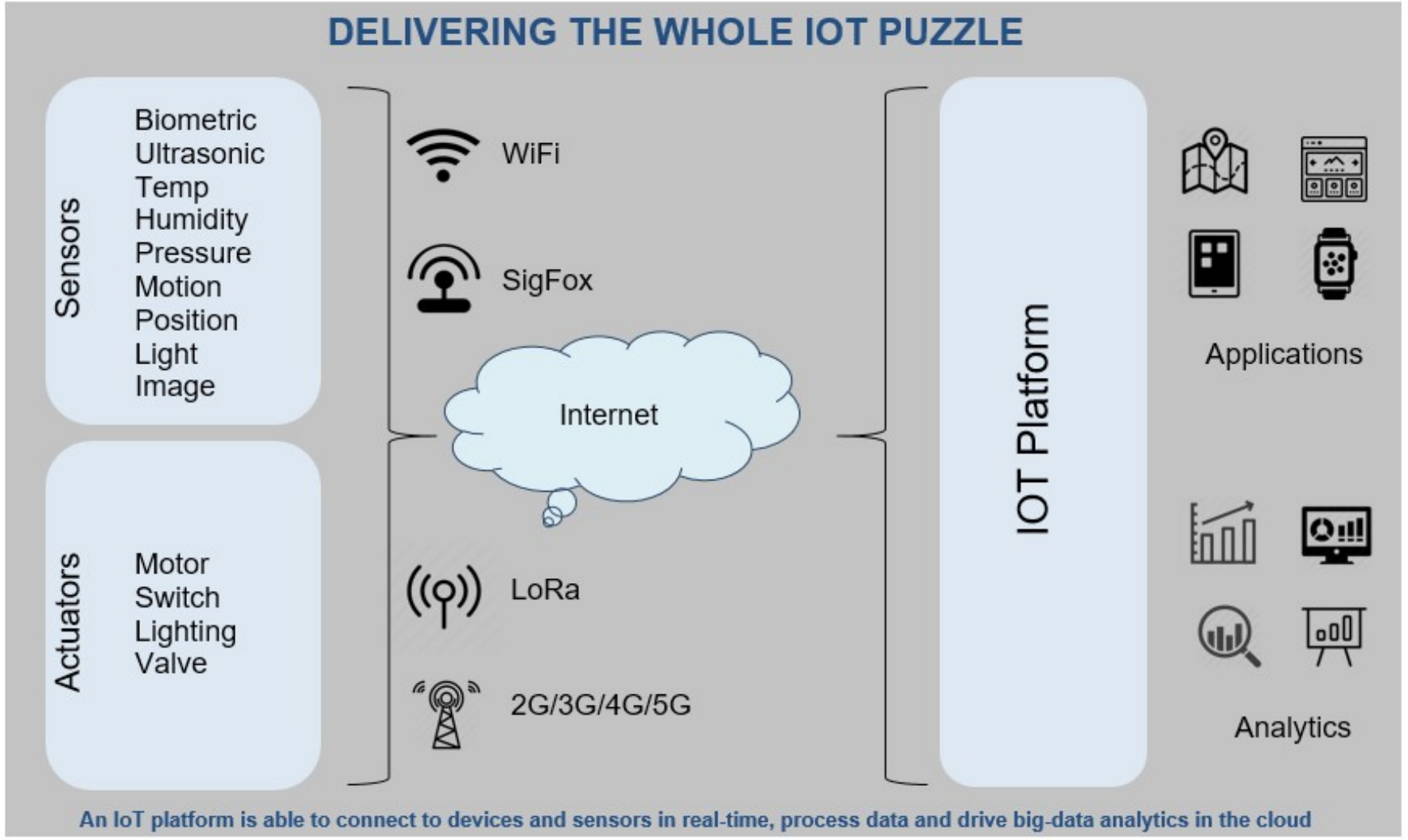
Examination Info

Clausthal

- Written Moodle-based examination
- 120min
- When? → 10.08.2022
  - 14:00 – 15:00 → Identity check
  - 15:00 – 17:00 → Exam

MOTIVATION

Overview


DELIVERING THE WHOLE IOT PUZZLE
Sensors
Biometric
Ultrasonic
Temp
Humidity
Pressure
Motion
Position
Light
Image
Actuators
Motor
Switch
Lighting
Valve
WiFi
SigFox
Internet
LoRa
2G/3G/4G/5G
IOT Platform
Applications
Analytics
An IoT platform is able to connect to devices and sensors in real-time, process data and drive big-data analytics in the cloud

CLOUDS

Cloud Computing – Definition

“Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.”

National Institute of Standards and Technology (NIST), “The NIST Definition of Cloud Computing”, 2011.

Overview

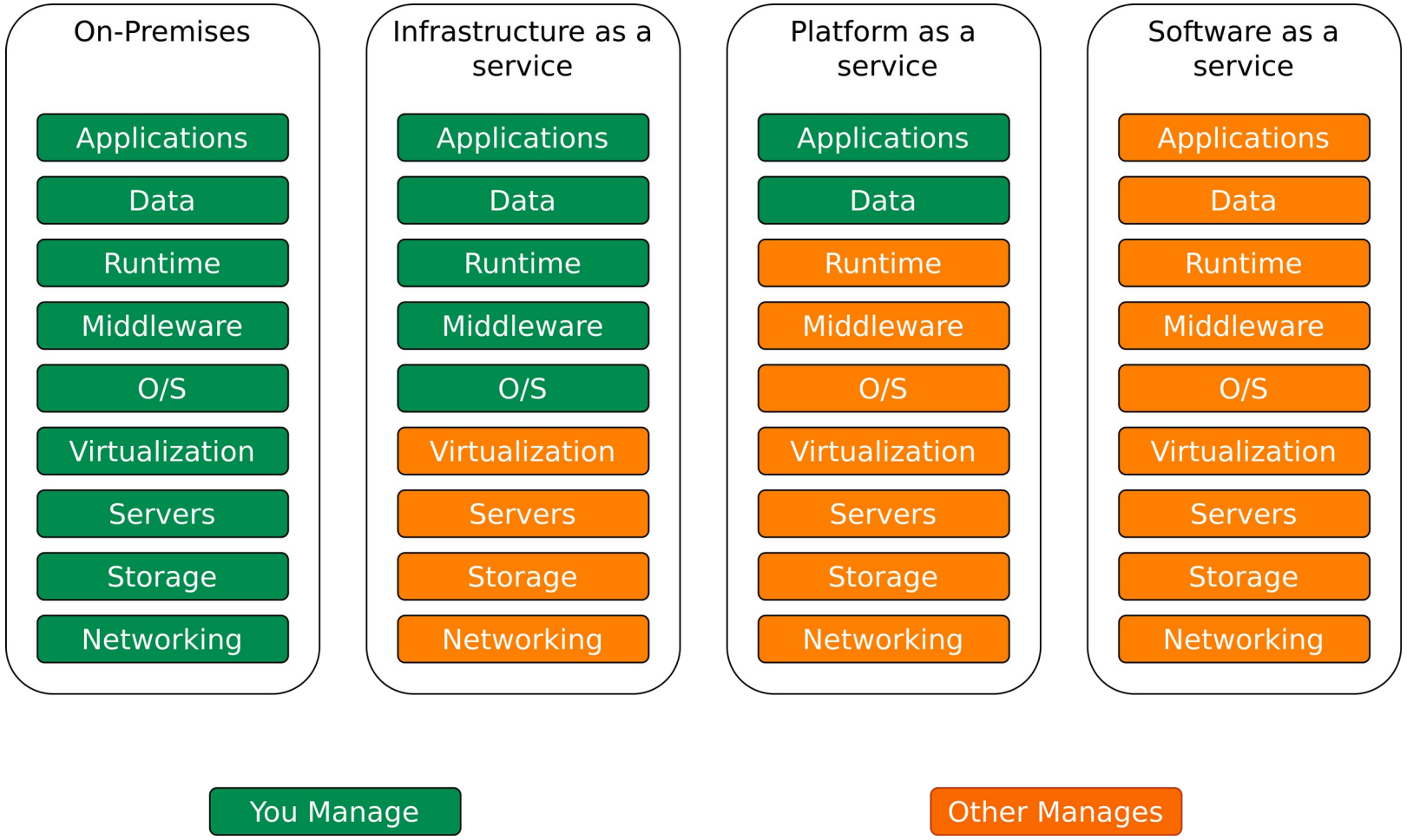


Image recreated from https://blogs.bmc.com

Cloud Service Models – Examples

IaaS:

- Amazon Web Services (AWS)
- Microsoft Azure
- Google Compute Engine

Cloud Service Models – Examples

IaaS:

- Amazon Web Services (AWS)
- Microsoft Azure
- Google Compute Engine

PaaS:

- Google App Engine
- Heroku, OpenShift
- AWS Elastic Beanstalk

Cloud Service Models – Examples

IaaS:

- Amazon Web Services (AWS)
- Microsoft Azure
- Google Compute Engine

PaaS:

- Google App Engine
- Heroku, OpenShift
- AWS Elastic Beanstalk

SaaS:

- Google Apps
- Dropbox
- Cisco Webex
- Salesforce
- GoToMeeting
- Zoom

Cloud Billing Models

- Pay-per-use (z.B. AWS S3, AWS ECU, etc.)
- Base fee + Pay-per-use
- Pay-per-user (Dropbox)
- Pay-per-transaction
- Pay-per-bandwidth/troughput

Cloud Platforms Market Shares

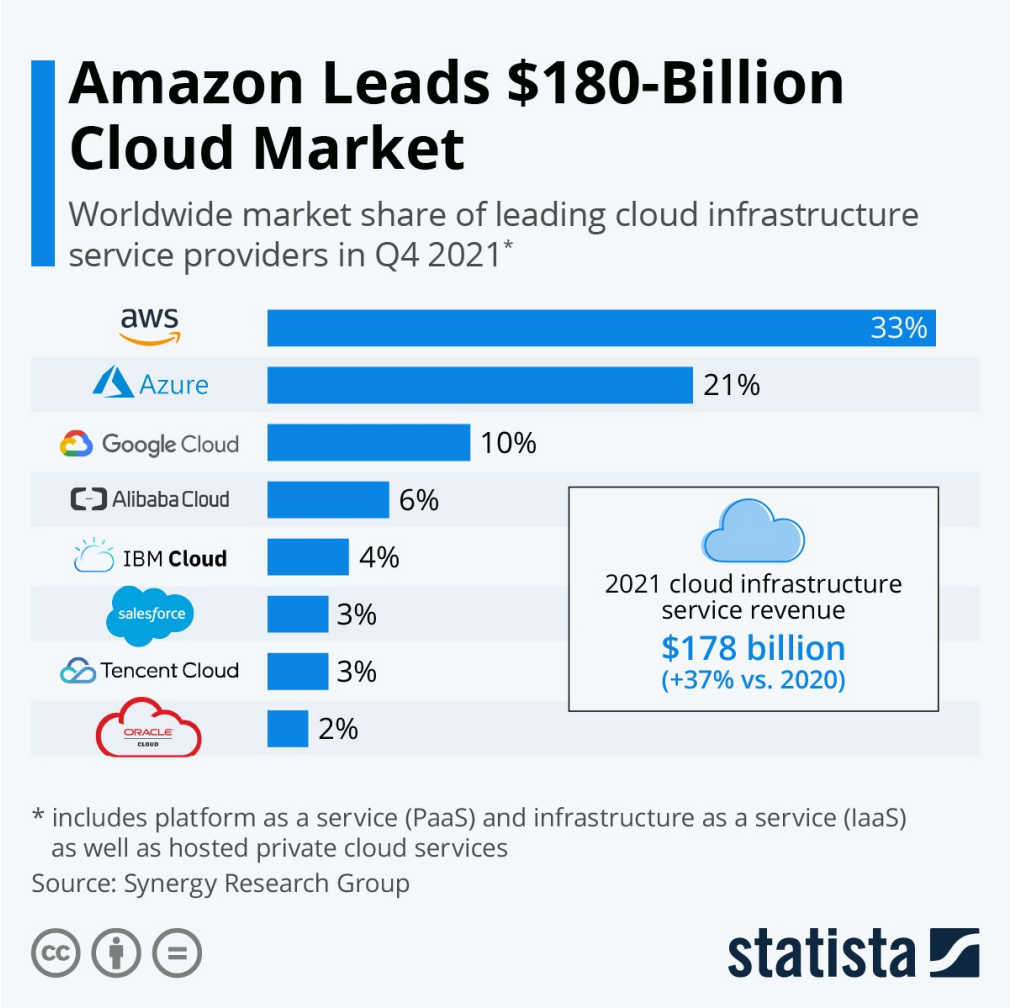


Image from https://www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers/ is licensed under CC BY-ND 3.0 (https://creativecommons.org/licenses/by-nd/3.0/).

Cloud Billing Models – AWS Glacier Example

- Storage pricing
  - $0.0045 per GB / Month
- Retrieval pricing

Retrieval Time	Data Retrievals
Expedited	$0.036 per GB
Standard	$0.012 per GB
Bulk	$0.003 per GB

- Retrieval request pricing

Retrieval Time	Retrieval Requests
Expedited	$12.00 per 1,000 requests
Standard	$0.036 per 1,000 requests
Bulk	$0.03 per 1,000 requests

https://aws.amazon.com/glacier/pricing/

Cloud Billing Models – AWS Calculator

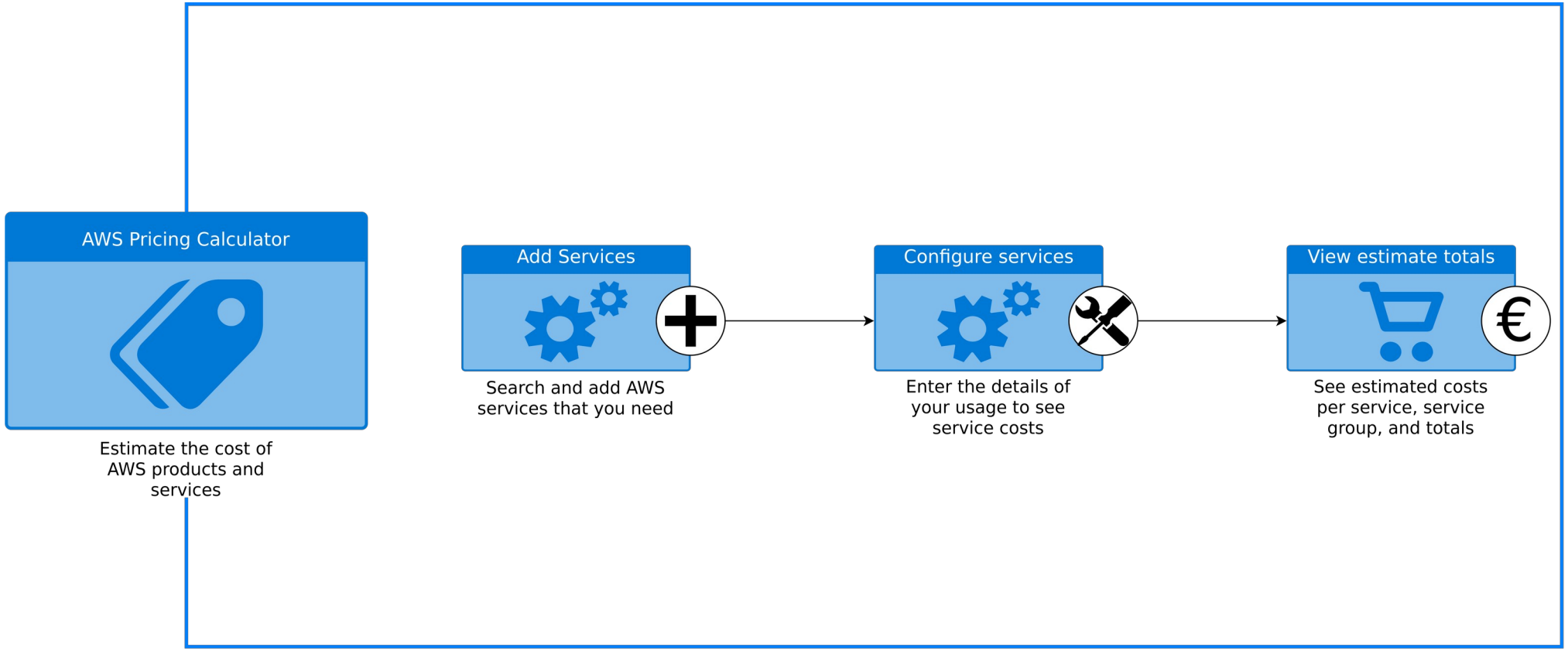


Image recreated based on https://calculator.aws/#/addService

Public, Hybrid and Private Clouds

Parameters/ Type	Public cloud	Private cloud	Hybrid cloud	Community cloud
Description	Services are available for public users	Built up with existing private infrastructure. This type of cloud has some authentic users who can dynamically provision the resources	A heterogeneous distrubited system, resulting from a private cloud, which incorporates different types of services and resources from public clouds	Different types of cloud are integrated together to meet a common or particular need for some organizations
Scalability	Very High	Limited	Very High	Limited
Reliability	Moderate	Very High	Medium to High	Very High
Security	Totally depends on the service provider	High class security	Secure	Secure
Performance	Low to medium	Good	Good	Very Good
Cost	Cheaper	High Cost	Costly	Costly
Examples	Amazon EC2, Google AppEngine	VMWare, Microsoft KVM, Xen	IBM, HP, VMWare vCloud, Eucalyptus	SolaS Community Cloud, VMWare

Table reacreated from Pal, Debranjan, Sourav Chakraborty, and Amitava Nag. "Cloud Computing: A Paradigm Shift in IT Infrastructure“, 2015.

Public, Hybrid and Private Clouds

Parameters/ Type	Public cloud	Private cloud	Hybrid cloud	Community cloud
Description	Services are available for public users	Built up with existing private infrastructure. This type of cloud has some authentic users who can dynamically provision the resources	A heterogeneous distrubited system, resulting from a private cloud, which incorporates different types of services and resources from public clouds	Different types of cloud are integrated together to meet a common or particular need for some organizations
Scalability	Very High	Limited	Very High	Limited
Reliability	Moderate	Very High	Medium to High	Very High
Security	Totally depends on the service provider	High class security	Secure	Secure
Performance	Low to medium	Good	Good	Very Good
Cost	Cheaper	High Cost	Costly	Costly
Examples	Amazon EC2, Google AppEngine	VMWare, Microsoft KVM, Xen	IBM, HP, VMWare vCloud, Eucalyptus	SolaS Community Cloud, VMWare

?

?

Table reacreated from Pal, Debranjan, Sourav Chakraborty, and Amitava Nag. "Cloud Computing: A Paradigm Shift in IT Infrastructure“, 2015.

Cloud – Data Protection and Privacy

Cloud – Data Protection and Privacy

- Am I allowed to transfer the data to the cloud? (e.g. patient data)

Cloud – Data Protection and Privacy

- Am I allowed to transfer the data to the cloud? (e.g. patient data)
- Does the data have to be encrypted and/or anonymized/pseudonymized?

Cloud – Data Protection and Privacy

- Am I allowed to transfer the data to the cloud? (e.g. patient data)
- Does the data have to be encrypted and/or anonymized/pseudonymized?
- Am I allowed to transfer data to foreign servers?

Cloud – Data Protection and Privacy

- Am I allowed to transfer the data to the cloud? (e.g. patient data)
- Does the data have to be encrypted and/or anonymized/pseudonymized?
- Am I allowed to transfer data to foreign servers?
- Liability issues in the event of data loss and data manipulation

Cloud – Data Protection and Privacy

- Am I allowed to transfer the data to the cloud? (e.g. patient data)
- Does the data have to be encrypted and/or anonymized/pseudonymized?
- Am I allowed to transfer data to foreign servers?
- Liability issues in the event of data loss and data manipulation
- Access to the data on the part of the cloud provider, third parties or secret services

Cloud – Data Protection and Privacy

- Am I allowed to transfer the data to the cloud? (e.g. patient data)
- Does the data have to be encrypted and/or anonymized/pseudonymized?
- Am I allowed to transfer data to foreign servers?
- Liability issues in the event of data loss and data manipulation
- Access to the data on the part of the cloud provider, third parties or secret services
- Cloud service is temporarily unavailable (SLAs – Service Level Agreements)

Self-Hosted Storage Clouds

www.nextcloud.com
www.owncloud.com
www.syncthing.net

Self-Hosted Computing Clouds

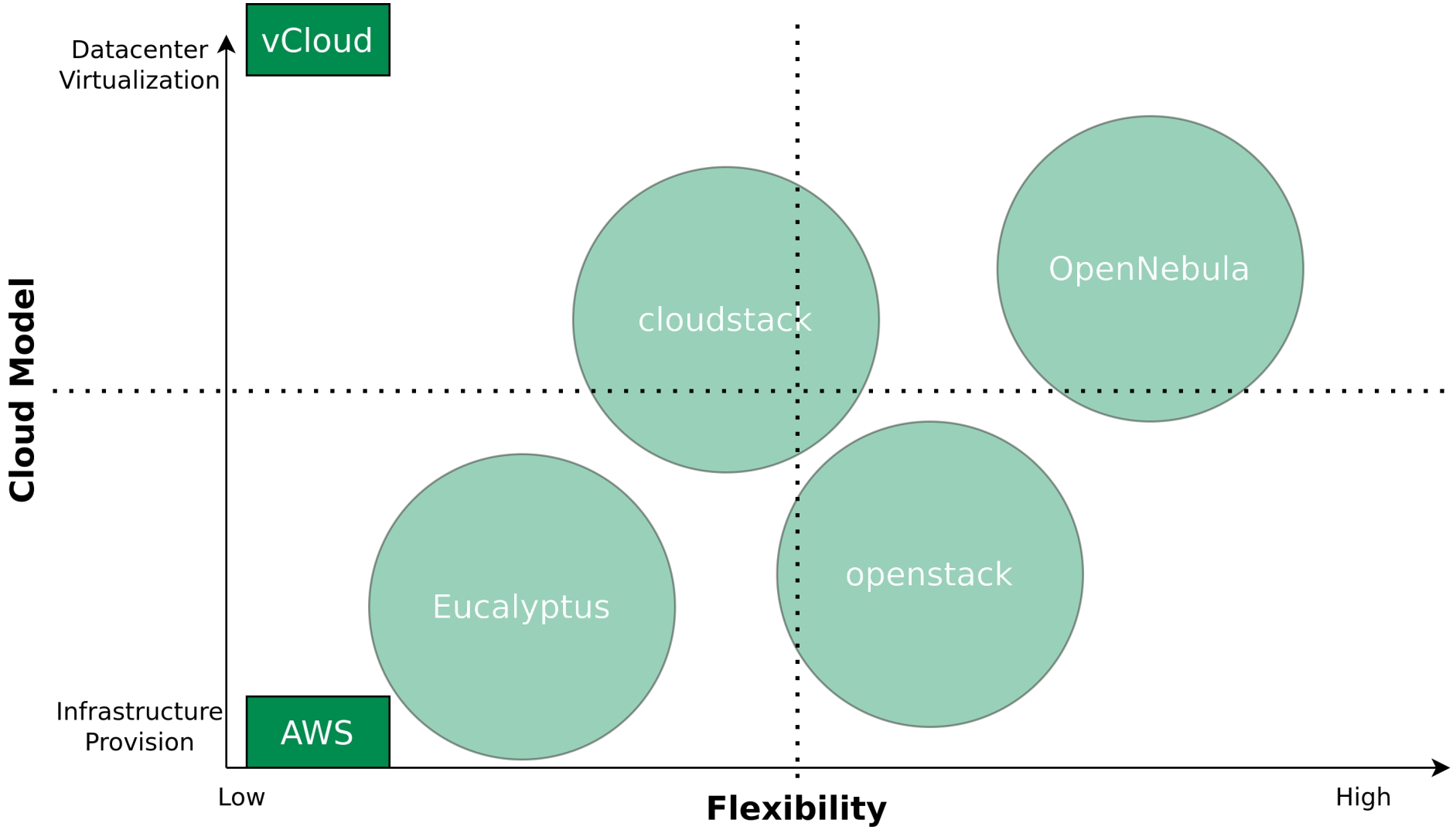


Image recreated from https://opennebula.org

IOT CLOUDS

Domain-specific Clouds

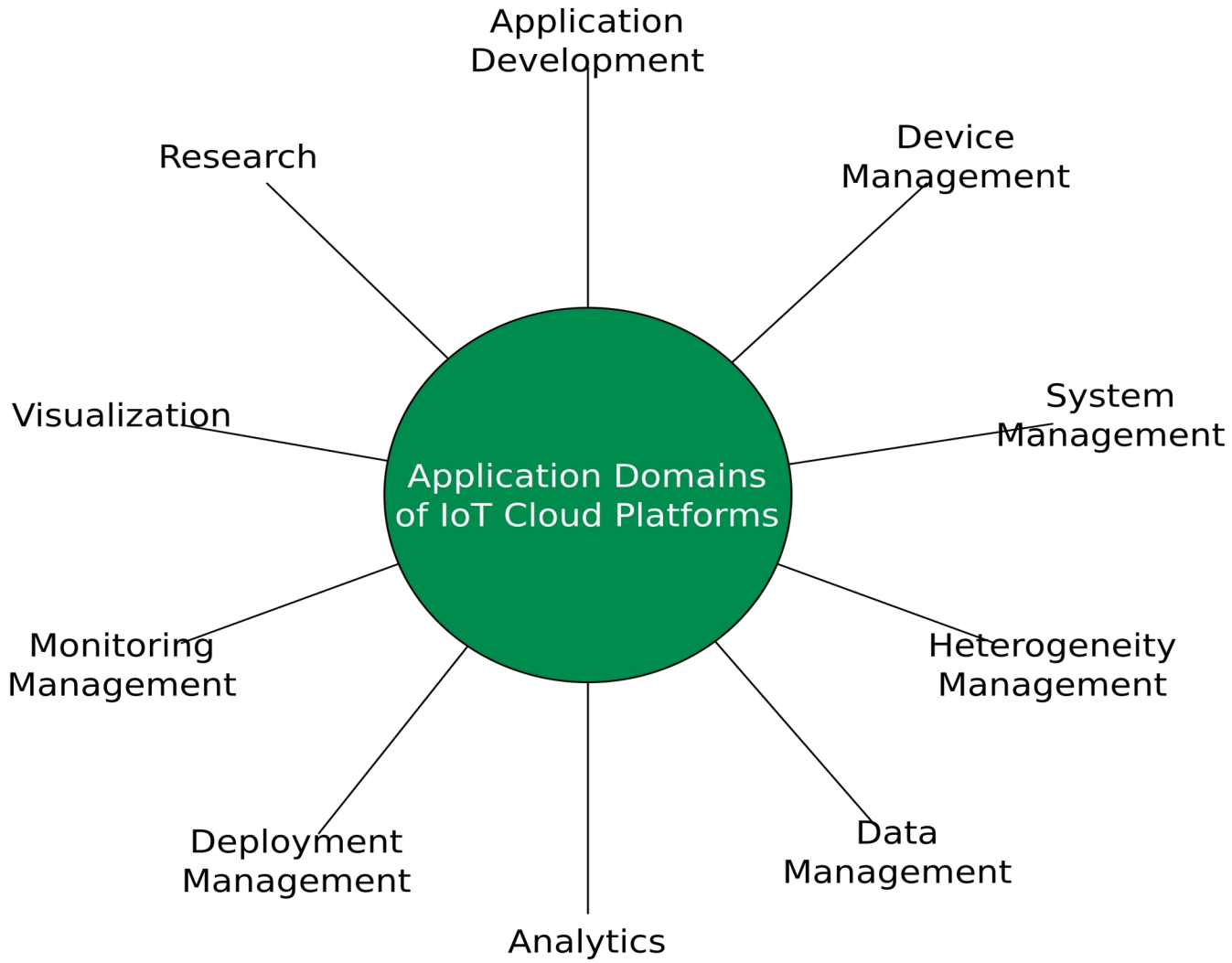


Image recreated from P. P. Ray, “A Survey of IoT Cloud Platforms,” Future Computing and Informatics Journal, vol. 1, no. 1-2, pp. 35–46, 2016.

So what is an IoT Platform exactly?

IoT platforms are the support software that connects everything in an IoT system, thereby facilitating communication, data flow, device management, and the functionalities of applications.

So what is an IoT Platform exactly?

IoT platforms are the support software that connects everything in an IoT system, thereby facilitating communication, data flow, device management, and the functionalities of applications.

- Connect hardware, such as sensors and devices
- Handle different hardware and software communication protocols
- Provide security and authentication for devices and users
- Collect, visualize, and analyze data the sensors and devices gather
- (External) web-service or application integration

IoT Clouds

- AWS IoT
- ThingSpeak
- Samsung Artik Cloud
- Microsoft Azure IoT Platform
- IBM Watson IoT

IoT Clouds – Comparison

Platform	FIWARE	OpenMTC	SiteWhere	Webinos	AWS IoT	IBM Watson IoT Platform	MS Azure IoT Hub	Samsung SmartThings
Sensor/ Actuator	-*	Sensors & Actuators	-*	-*	-*	-*	-*	Sensors & Actuators & Devices & Users & Things
Device	Device/ NGSI Device	-*	Data from Devices + Commands to Devices	PZP: Policy + Session + Discovery + Context Manager	Things	-*	Device	Sensors & Actuators & Devices & Users & Things + Clients (-Devices)
Gateway	IoT Edge	Front-End: Core Features + Connectivity + Back-End: Connectivity	-*	PZP: Sync + Messaging Manager + PZH: Sync Manager	-*	Connect	Cloud Protocol Gateway + Field Gateway	Hub and Client Connectivity + Device Type Handlers
IoT Integration Middleware	IoT Back-End + Data Context Broker	Back-End: Connectivity + Core Features + Application Enablement	SiteWhere Tenant Engine	PZH: User Authentication + Policy Repository + Policy Enforcement + Web APIs	Message Broker + Thing Shadows + Thing Registry + Rules Engine + Security & Identity	Analytics + Risk Management + Connect + Information Management + Bluemix Open Standards Based Services + Flexible Deployment	IoT Hub + Event Processing and Insight + Device Business Logic, Connectivity Monitoring + Application Device Provisioning and Management	Application Management System + Subscription Processing
Application	-*	Applications + Other M2M Platform	Integration- + REST-components	Third Party Applications	Amazon Services + IoT Applications	IoT Industry Solutions + Third Party Apps	Application Device Provisioning and Management	Event Stream + Web UI + Core APIs + External System + Physical Graph

Table recreated from Guth, Jasmin, Uwe Breitenbücher, Michael Falkenthal, Paul Fremantle, Oliver Kopp, Frank Leymann, and Lukas Reinfurt. "A detailed analysis of IoT platform architectures: concepts, similarities, and differences." In Internet of everything, pp. 81-101. Springer, Singapore, 2018.

AWS IoT for Industrial Applications

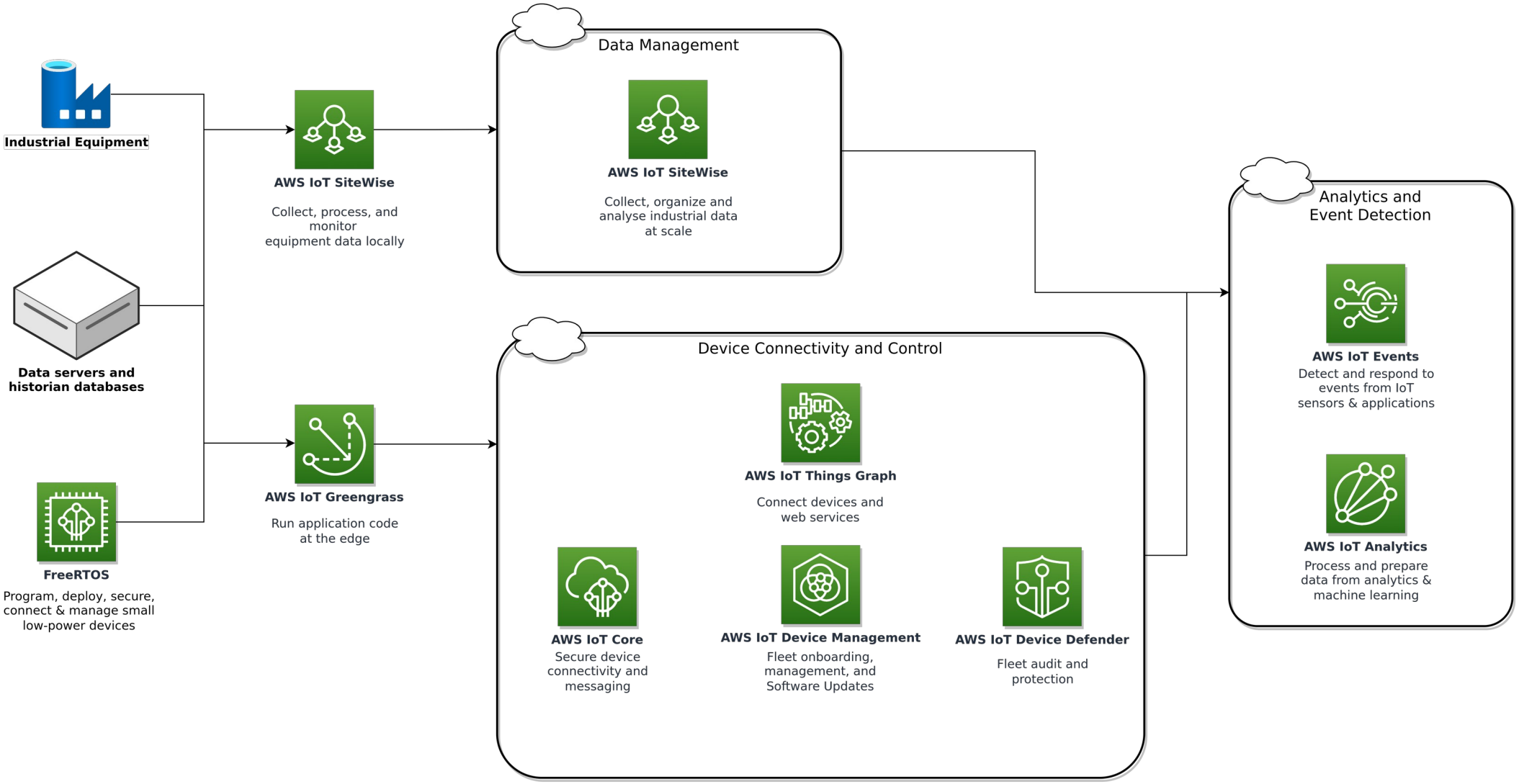


Image recreated from https://aws.amazon.com/iot/solutions/industrial-iot

AWS IoT for the Connected Homes

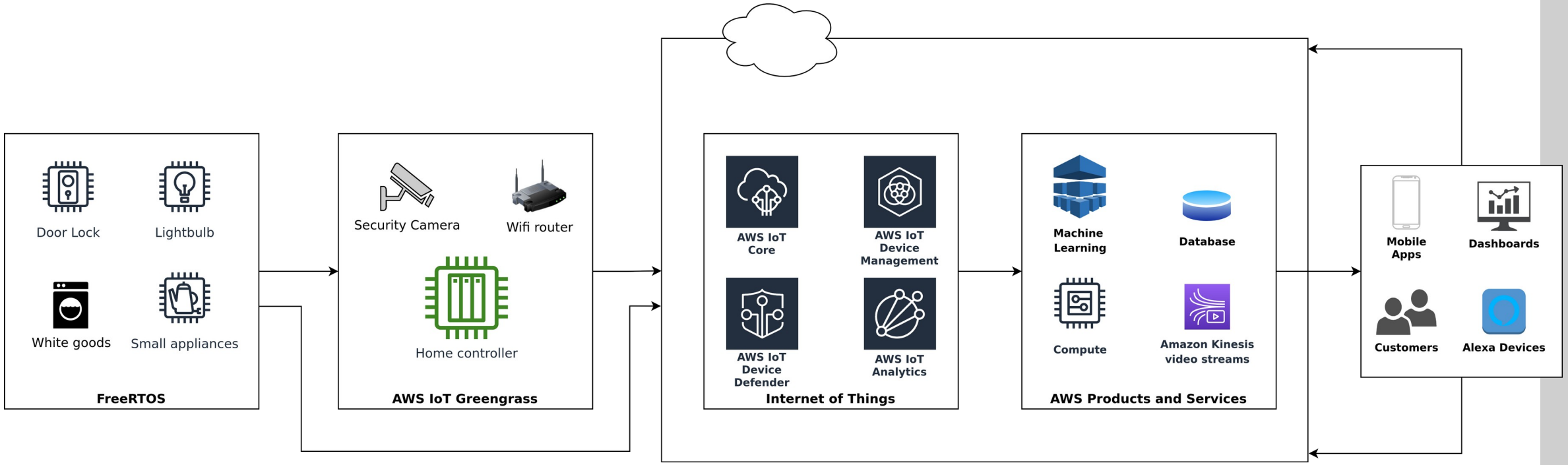


Image recreated from https://aws.amazon.com/iot/solutions/connected-home

EDGE/FOG COMPUTING

Motivation

Why do we need Edge/Fog computing if we have a Cloud?

Motivation

Why do we need Edge/Fog computing if we have a Cloud?

- Data volume to large
- Network latency
- Costs (data transport constraints/limitations)

Motivation

Why do we need Edge/Fog computing if we have a Cloud?

- Data volume to large
- Network latency
- Costs (data transport constraints/limitations)

→ Edge computing shifts the computing capacity from central servers or the cloud closer to the IoT devices themselves. Thus, allowing for faster data processing, shorter response times, and saving bandwidth.

Edge/Fog for IoT

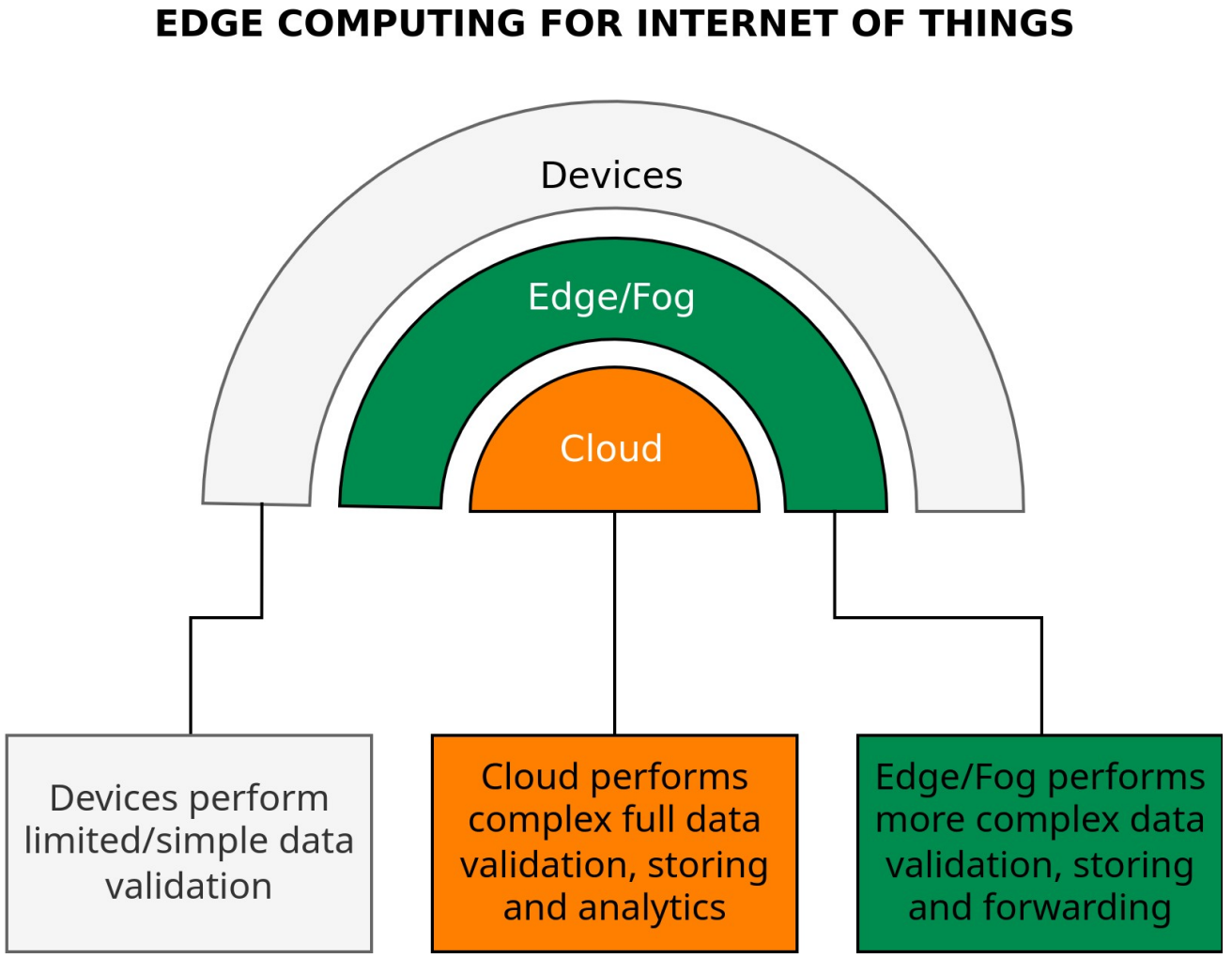
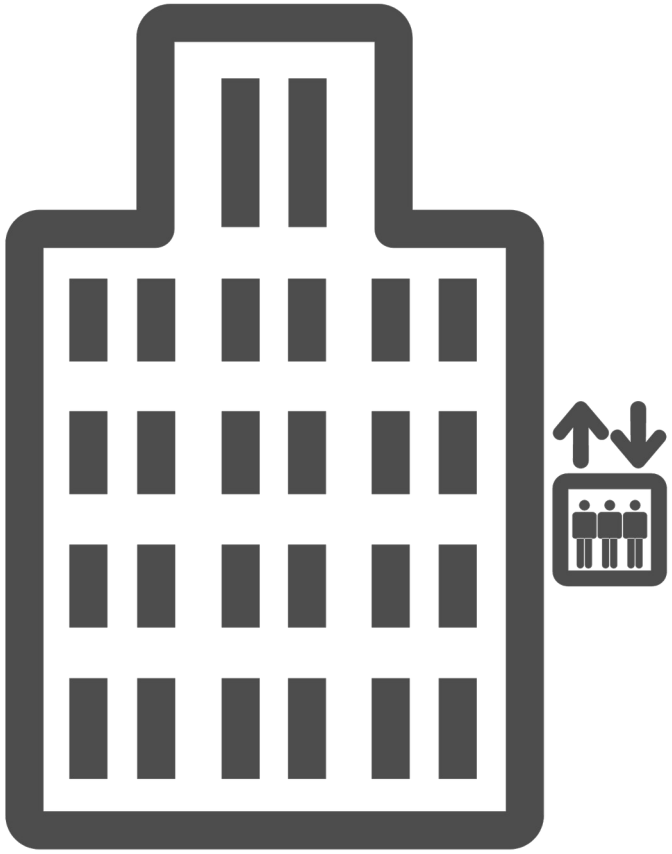


Image recreated from https://thecustomizewindows.com/2019/09/edge-computing-vs-cloud-computing/

FogHorn – Example

SMART BUILDINGS



OPTIMIZING ELEVATOR PERFORMANCE

50+ ML Models on tiny controllers deliver predictive maintenance

Challenge

1. Monitor 1.5M+ Elevators / Escalators deployed globally
2. Limited communications / compute resources
3. Mine sensor information for actionable insights
4. Reduce inspection / repair fees of ~$2K / event

Foghorn Solution

1. Foghorn installed on existing motion sensor kits, < 1 Gb footprint
2. CEP time-aligns state and activity data in < 20 lines of c
3. 40+ ML models generate predictive maintenance alerts

Benefits

Smart, not scheduled maintenance

Reduce costly repair and servicing

New managed service revenue

Image recreated from http://web.stanford.edu/class/ee392b/lecture/may22/Foghorn.pdf

Edge Computing – Definition

“Edge computing refers to the enabling technologies al-lowing computation to be performed at the edge of the network, on downstream data on behalf of cloud services and upstream data on behalf of IoT services”

Shi, Weisong, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. "Edge computing: Vision and challenges." IEEE internet of things journal 3, no. 5 (2016): 637-646.

Fog Computing – Definition

“Fog computing is a geographically distributed computing architecture with a resource pool consists of one or more ubiquitously connected heterogeneous devices (including edge devices) at the edge of network and not exclusively seamlessly backed by cloud services, to collaboratively provide elastic computation, storage and communication (and many other new services and tasks) in isolated environments to a large scale of clients in proximity.”

Yi, Shanhe, Zijiang Hao, Zhengrui Qin, and Qun Li. "Fog computing: Platform and applications." In 2015 Third IEEE workshop on hot topics in web systems and technologies (HotWeb), pp. 73-78. IEEE, 2015.

Edge and Fog Computing – What's the Difference?

- Depends on who you ask

Architecture – Edge/Fog

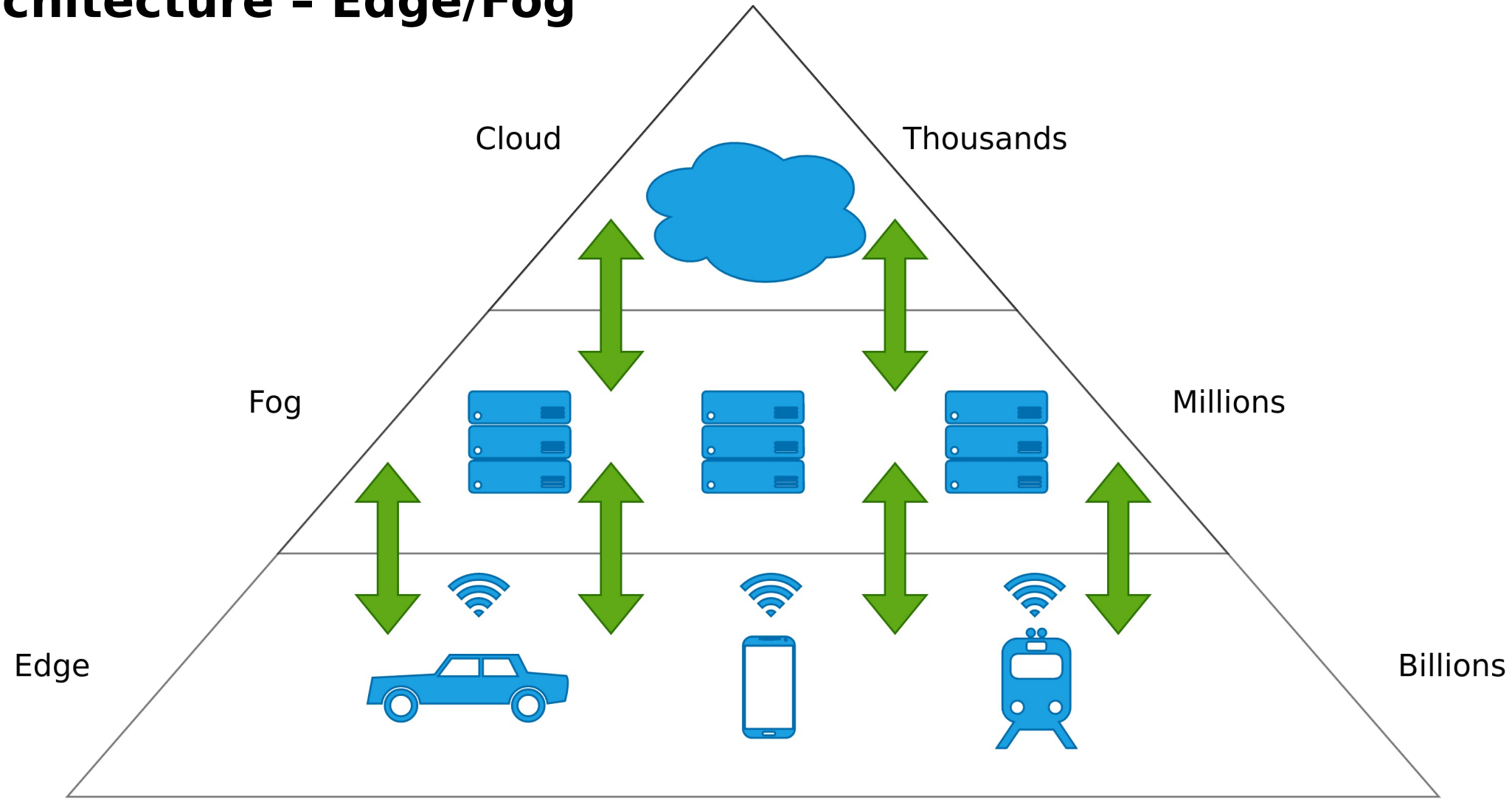


Image recreated from https://blog.octo.com/en/edge-computing-learn-to-delegate/

Amazon Lambda – Serverless Computing

- Event-driven (PUT to S3, API Call, etc.), serverless compute service that lets you run code without provisioning or managing servers
- Allows for automated back-end provision services triggered by custom HTTP requests
- Spin down services when not in use

- Difference to AWS EC2 instances?
  - EC2 is priced by the hour but metered by the second
  - Lambda is metered by rounding up to the nearest millisecond with no minimum execution time

Amazon Lambda@Edge

- Extension of AWS Lambda
- Execution of the AWS Lambda Code in AWS Edge Locations (entry points into the AWS ecosystem, e.g. CloudFront).

- Difference between Lambda and Lambda@Edge?
  - Lambda are regional services
  - Lambda@Edge is a global service
  - Lambda@Edge allows you to execute the logic across multiple location

Amazon Greengrass

- IoT open source edge runtime and cloud service that helps you build, deploy, and manage device software
- Enables AWS Cloud functions to be transferred to local devices
- Capture and analyze data/events close to the source
- Create AWS Lambda (serverless code) functions in the cloud and then deploy them to devices to run applications locally

Pro:

- Fast and direct event processing
- Works offline
- Simplifies IoT development and reduces costs (BUT you still have to pay!)

AWS Snowcone | Snowball Edge | Snowmobile

	AWS Snowcone	AWS Snowball Edge Storage Optimized	AWS Snowball Edge Compute Optimized	AWS Snowmobile
Usage Scenario	Edge computing, Data transfer, Edge storage	Data transfer, Edge storage	Edge computing, Data transfer	Data transfer
Usable HDD Storage	8 TB	80 TB	42 TB	100 PB
Usable SDD Storage	No	1 TB	7.68 TB	No
Usable vCPUs	2 vCPUs	40 vCPUs	52 vCPUs	N/A
Usable Memory	4 GB	80 GB	208 GB	N/A
GPU	No	No	Nvidia V100 (Optional)	No
Onboard Computing Options	AWS IoT Greengrass Amazon EC2 AMIs	AWS IoT Greengrass Amazon EC2 AMIs	AWS IoT Greengrass Amazon EC2 AMIs	N/A
DataSync	Yes	No	No	No
Transfers via S3 API	No	Yes	Yes	No
Device Size	9 in x 6 in x 3 in	28.3 in x 10.6 in x 15.5 in	28.3 in x 10.6 in x 15.5 in	N/A
Device Weight	2.1 kg	22.3 kg	22.3 kg	N/A
Encryption	256-Bit	256-Bit	256-Bit	256-Bit
Portability	Battery-Based operation	No	No	No
Wireless	Wi-Fi	No	No	No
Storage Clustering	No	Yes, 5-10 nodes	Yes, 5-10 nodes	N/A
HIPAA Compliant	Yes, eligible	Yes, eligible	Yes, eligible	No
Typical Job Lifetime	Offline or online data transfer: days-weeks Edge compute: weeks - years	Offline data transfer: days-weeks	Edge compute: weeks-years	Data-Migration: months

https://aws.amazon.com/snow/#Feature_comparison

BIG DATA

Big Data

More data ≠ More knowledge

More is not always better!

BigData – Definition

“BigData represents the information assets characterized by such a high volume, velocity and variety to require specific technology and analytical methods for its transformation into value”

De Mauro, Andrea, Marco Greco, and Michele Grimaldi. "What is big data? A consensual definition and a review of key research topics." In AIP conference proceedings, vol. 1644, no. 1, pp. 97-104. American Institute of Physics, 2015.

BigData – Definition

“BigData represents the information assets characterized by such a high volume, velocity and variety to require specific technology and analytical methods for its transformation into value”

→ Large and complex data sets that are so massive that classic data processing software cannot manage them, but whose information value offers new solutions for existing business problems.

De Mauro, Andrea, Marco Greco, and Michele Grimaldi. "What is big data? A consensual definition and a review of key research topics." In AIP conference proceedings, vol. 1644, no. 1, pp. 97-104. American Institute of Physics, 2015.

BigData – 4 V’s

BigData – 4 V’s

- **V**olume

BigData – 4 V's

- **V**olume
- **V**elocity

BigData – 4 V's

- **V**olume
- **V**elocity
- **V**ariety

BigData – 4 V's

- **V**olume
- **V**elocity
- **V**ariety
- **V**eracity

BigData – Volume

- Large amounts of structured and unstructured data
- Various heterogeneous sources
- Several TBs up to hundreds of PBs

BigData – Velocity

- Rate at which new data is generated
- Speed at with which data is transferred and analyzed

BigData – Velocity

- Rate at which new data is generated
- Speed at with which data is transferred and analyzed

- Best case:
  - Data processing is faster than data generation
  - Process data while it is being generated without having to write it into a database

BigData – Variety

- Heterogeneity of data
- Structured vs. unstructured
- Data type and structure partly unknown (text, audio, video, etc.)

- Who generates the data?
  - Humans
  - Sensors
  - Machines
    - Hardware
    - Software

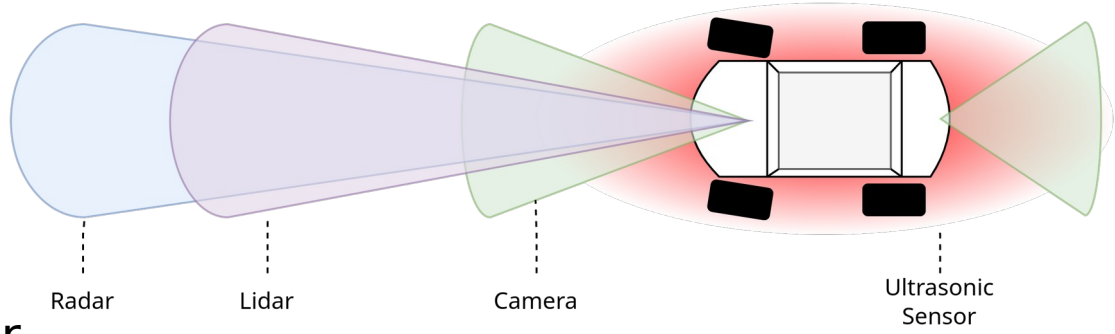
BigData – Veracity

- How accurate is the collected data → Can I trust it?
- Lower quality can be compensated for by higher volume if necessary
- Challenges:
  - Tweets with spelling mistakes or abbreviations.

  → e.g. u→you, thr→there, teh→the
- Bias of data (e.g. political).
- Detect and filter noise and abnormalities

Challenges – Data Volumes

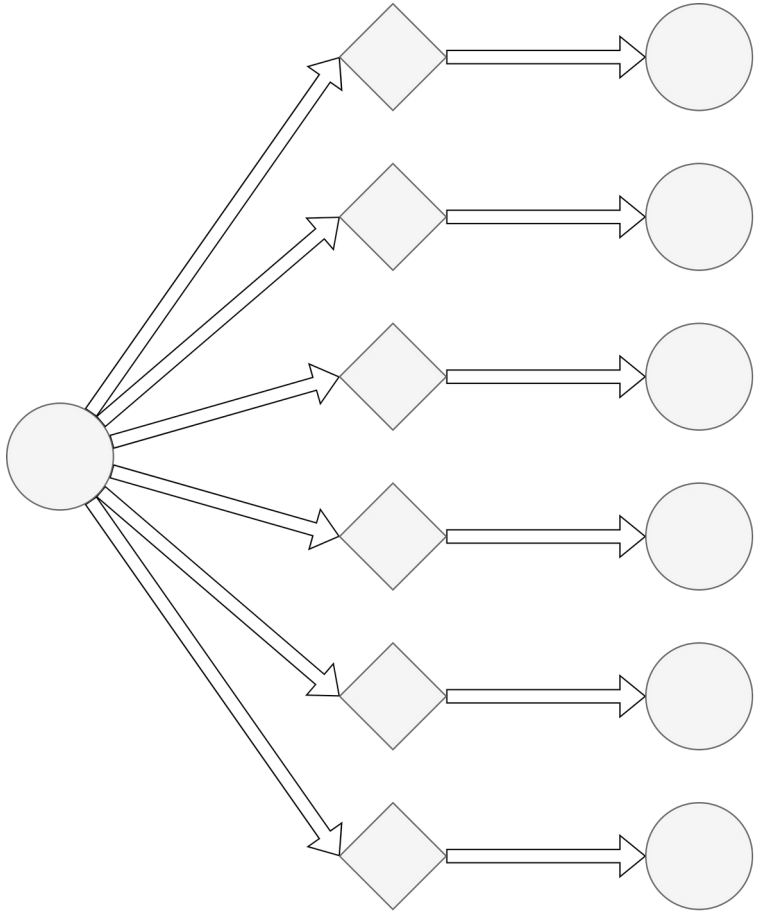
- RADAR
  - 4-6 Sensors: 0.1 – 15 Mbit/s per Sensor
- LIDAR
  - 1-5 Sensors: 20 – 100 Mbit/s per Sensor
- CAMERA
  - 6-12 Sensors: 500 - 3500 Mbit/s per Sensor
- ULTRASONIC
  - 8-16 Sensors: < 0.01 Mbit/s per Sensor
- VEHICLE MOTION, GNSS, IMU
    - < 0.1 Mbit/s per Sensor



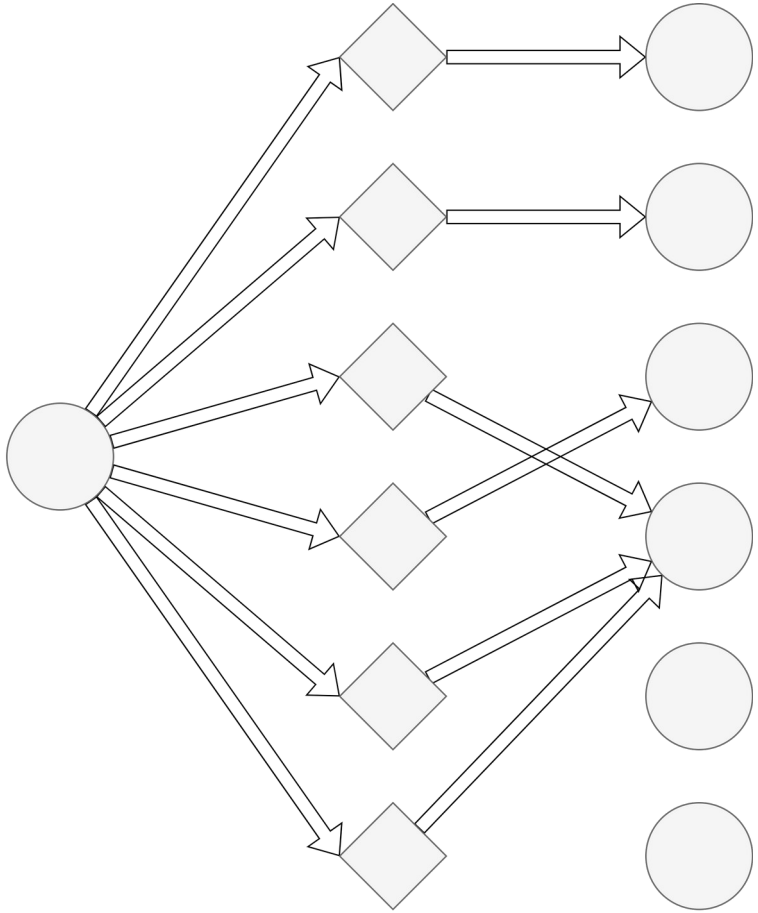
- Total Sensor Bandwidth = 3 Gbit/s (~ 1.4 TB/h) **or** 40 Gbit/s (~ 19 TB/h)

Image recreated from Heinrich, Stephan – Lucid Motors, „Flash Memory in the emerging age of autonomy“, Flash Memory Summit 2017 – https://www.flashmemorysummit.com/English/Collaterals/Proceedings/2017/20170808_FT12_Heinrich.pdf

Challenges – Scalability



Theory



Practice

MACHINE LEARNING

Overview

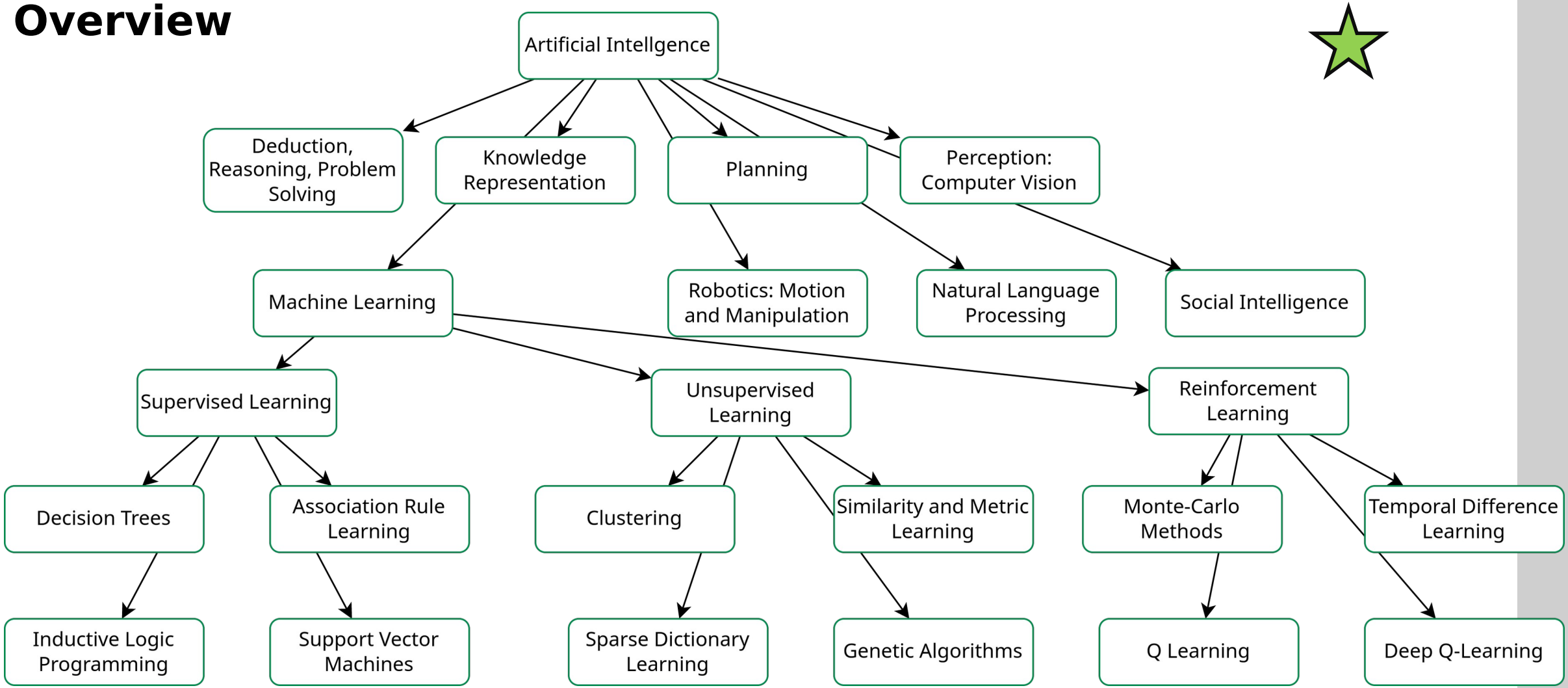


Image recreated based on Ajit Nazre and Rahul Garg, “A Deep Dive in the Venture Landscape of Artificial Intelligence and Machine Learning”, 2015.

Supervised Learning

The objective is to assign data to a class/grouping that is specified by the user. The key challenge is to build a model with the help of sample data, which then takes over the assignment independently.

Advantage:

- Leads to good results even with small amounts of data.

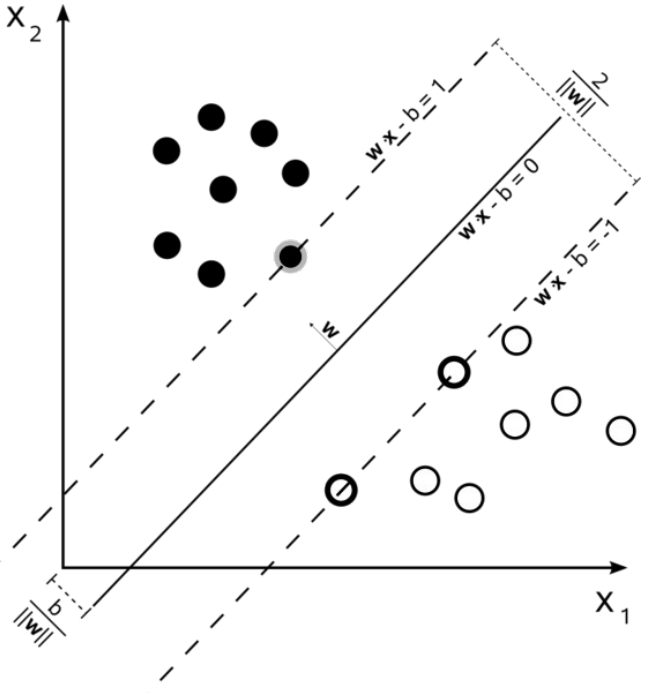
Disadvantage:

- Data must be labelled.

Supervised Learning – Examples

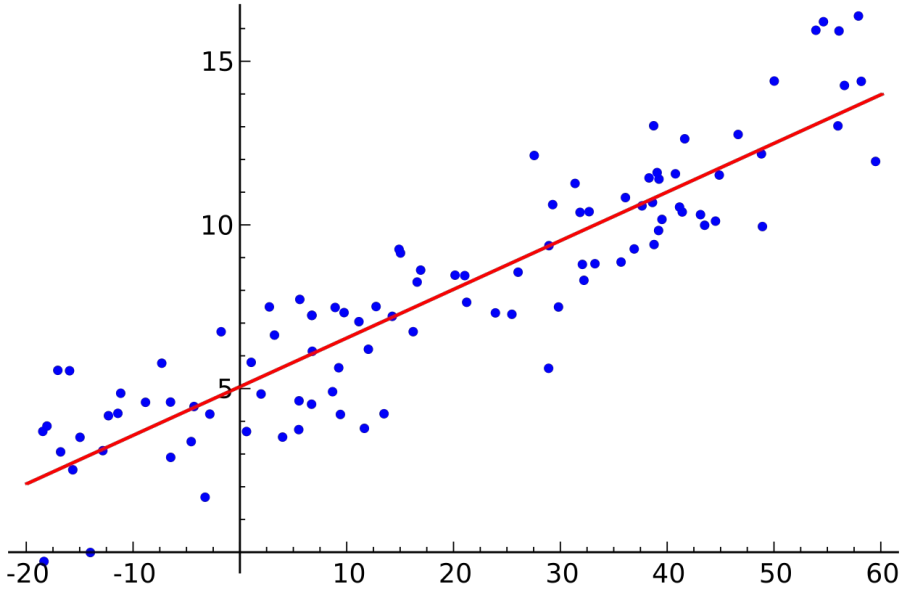
Classification problems:

- Patient has disease X or not
- E-mail is spam or not



Regression problems:

- Prediction of stock prices
- Prediction of real estate prices



„Linear Regression" by Sewaqu is sourced from the public domain.
"Graphic showing the maximum separating hyperplane and the margin." by Cyc is sourced from the public domain.

Unsupervised Learning

The objective is to identify unknown structures and relationships amongst data. The grouping and the division of the individual data sets is unknown.

Advantage:

- No labels necessary

Disadvantage:

- Requires large amounts of data

Unsupervised Learning – Examples

- Text analysis
- Speech analysis
- Image recognition and processing

→ Results are used, for example, in voice assistant systems or social bots

Reinforcement Learning

An algorithm learns through trial and error about its environment and which actions it can perform. The executed actions of the algorithm are either "rewarded" or "punished" → reinforcement learning.

Advantage:

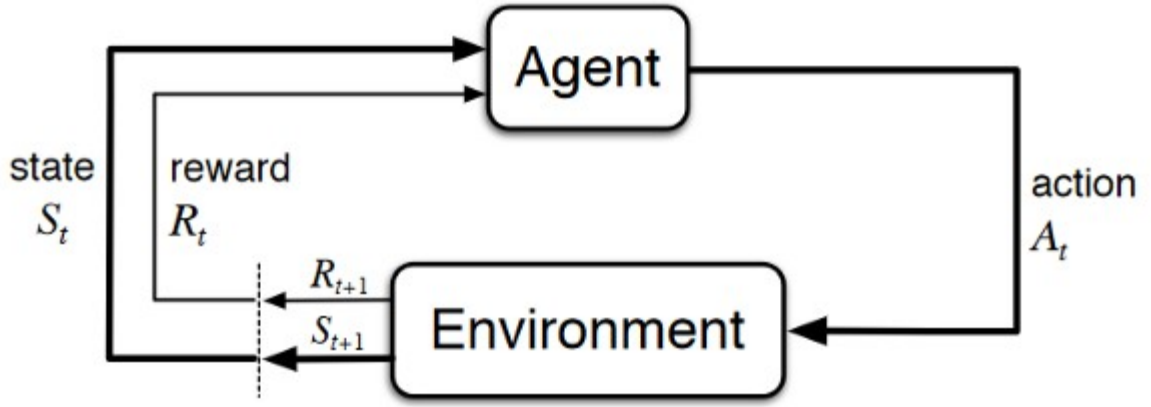
- Allows the programming of autonomously acting systems

Disadvantage:

- Complex structure and training process.

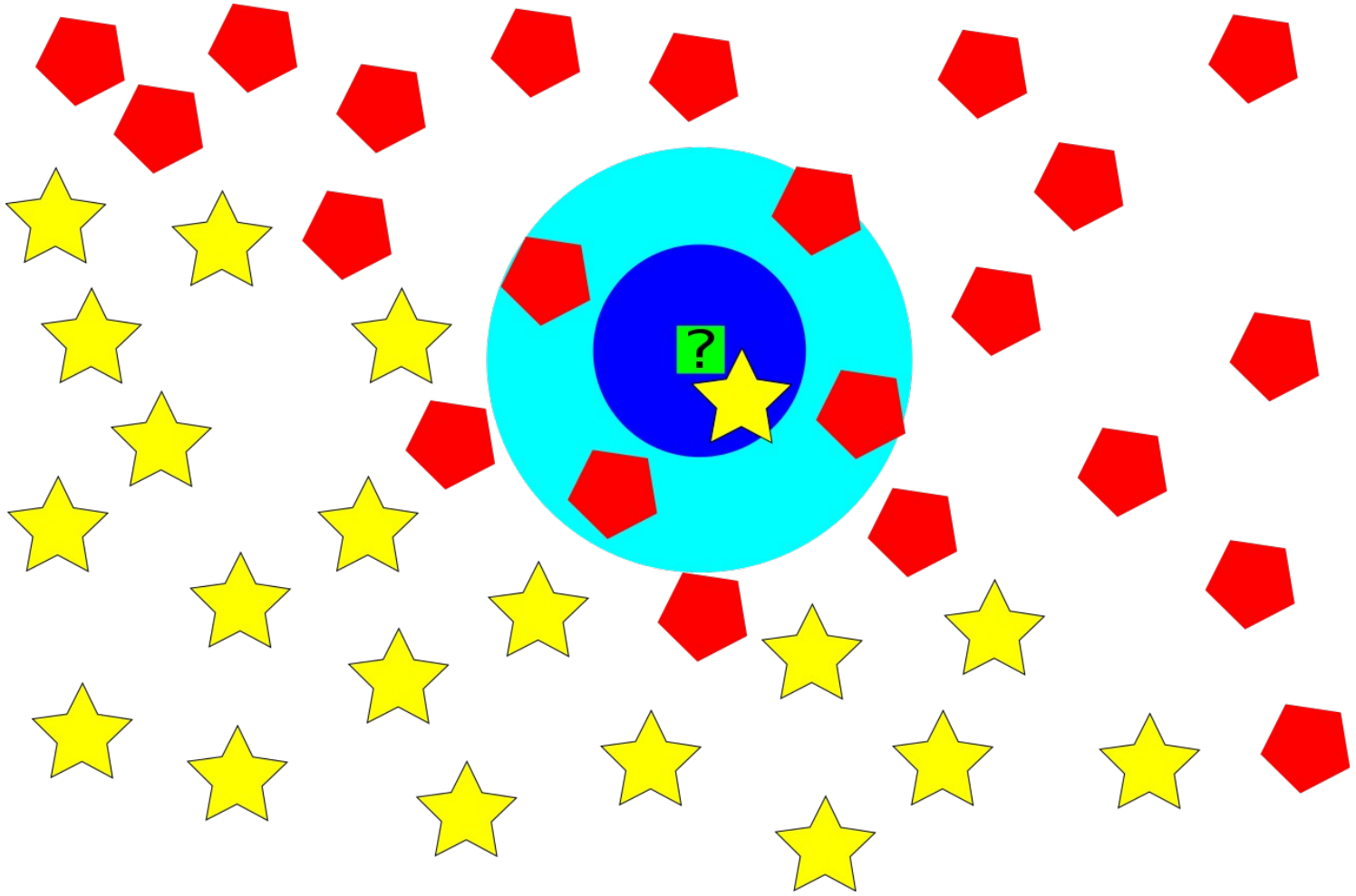
Reinforcement Learning – Examples

- Robotics
- Chemistry (optimization of chemical reactions)
- Recommendation systems (Amazon)



"Reinforcement learning diagram of a Markov decision process based on a figure from 'Reinforcement Learning An Introduction' second edition by Sutton and Barto." by EbattleP is licensed under CC BY-SA 4.0 (https://creativecommons.org/licenses/by-sa/4.0/)

K-Nearest Neighbor



Drichel, Alexander, „Klassifikation im R^2 durch das k-nearest-neighbor Verfahren (k=1, k=5)“ – https://de.wikipedia.org/wiki/Datei:KNNClass.svg is licensed under CC BY-SA 2.0 (https://creativecommons.org/licenses/by-sa/2.0/de/legalcode)

DATA ANALYTICS

Overview

- Includes the process of inspection, cleaning/preparation, transforming and modelling data for the purpose of extraction of useful information.
- Extracted information is the basis for subsequent decision-making processes
- Often in conjunction with data visualization

Overview

- Includes the process of inspection, cleaning/preparation, transforming and modelling data for the purpose of extraction of useful information.
- Extracted information is the basis for subsequent decision-making processes
- Often in conjunction with data visualization

⇒ Value of collected data depends on their interpretation and the decisions that are made on the basis of the data.

A Data-Driven Smart CE Framework

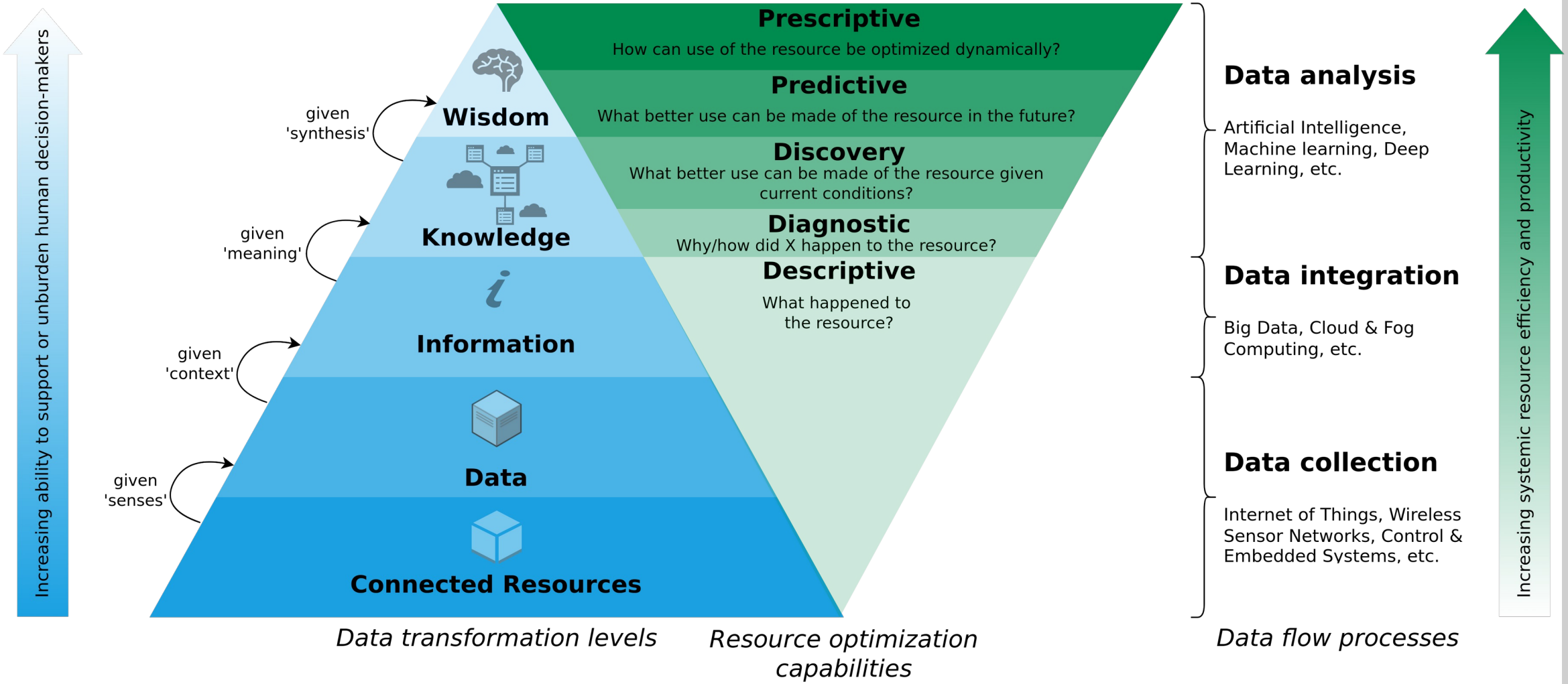


Image recreated from Kristoffersen et al. (2020) – The smart circular economy: A digital-enabled circular strategies framework for manufacturing companies.

Data Analytics – Overview

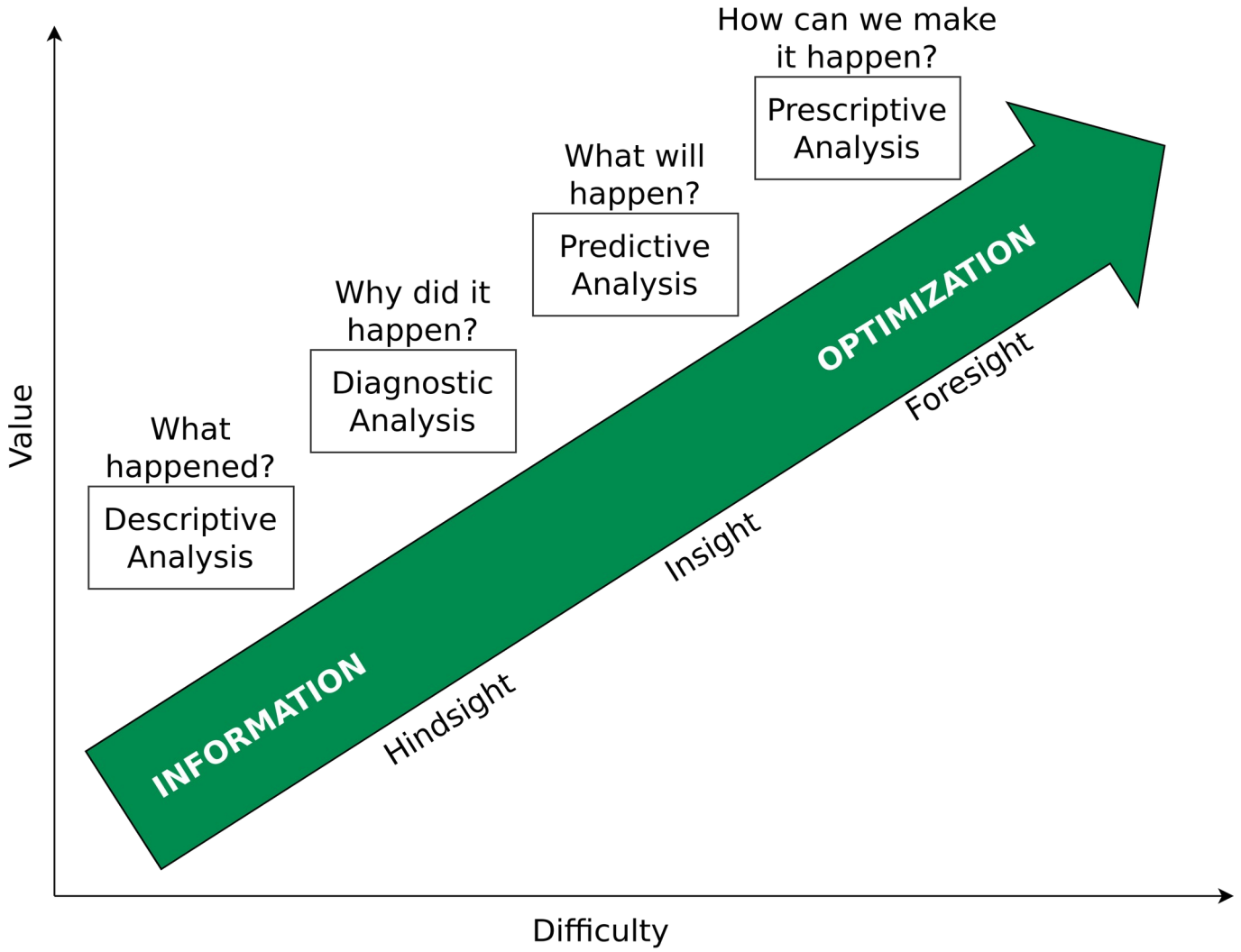


Image recreated from https://www.gartner.com/it-glossary/predictive-analytics/

Data Processing Example – Map Reduce

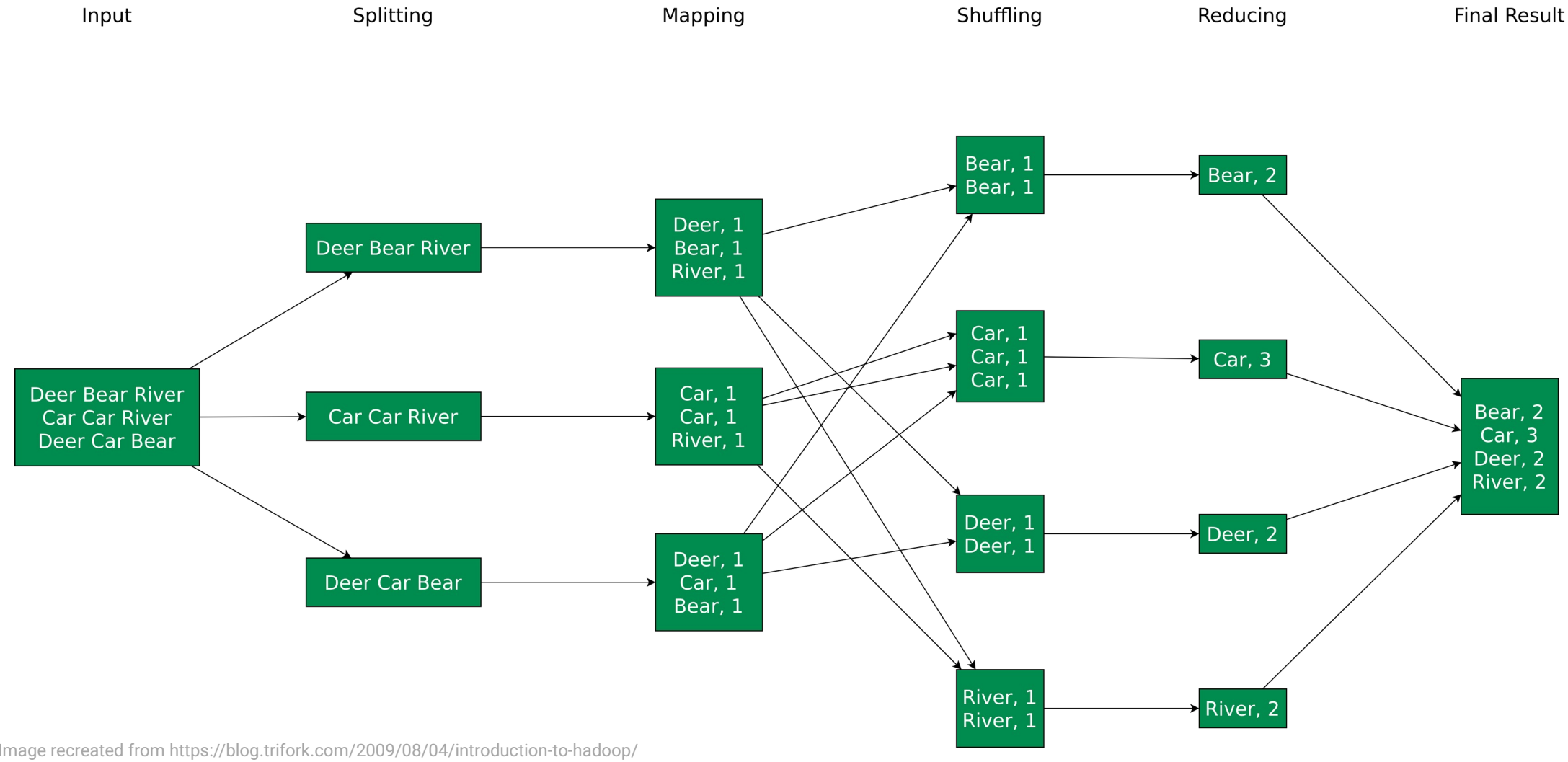
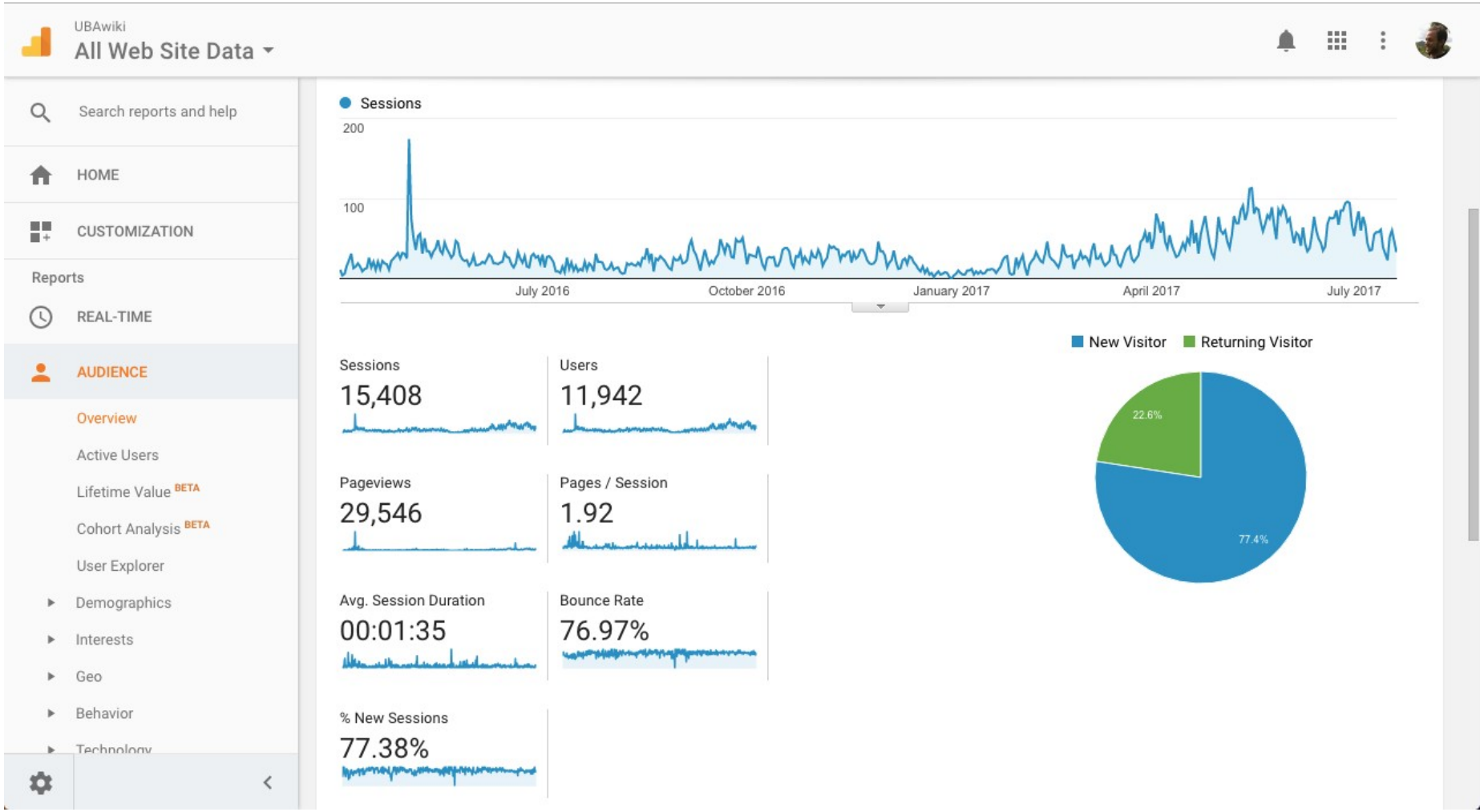


Image recreated from https://blog.trifork.com/2009/08/04/introduction-to-hadoop/

Data Visualisation Example – Google Analytics



Screenshot of the Google Analytics Audience screen for http://uba.wiki is licensed under CC BY-SA 4.0 (https://creativecommons.org/licenses/by-sa/4.0/deed.en)

Predictive Analytics

- Classification of new observations on the basis of training data from the past
- It is not a question of predicting the future!

  → Instead, to calculate the probability of the (non-)occurrence of an event with a certain degree of reliability.
- Possible applications in the creation of what-if scenarios, Risk management or outlier analyses

Predictive Analytics Example – Engine Vibrations

- Monitoring of the vibrations of specific engine parts
- Analysis of measured vibrations via Fast Fourier Transform (FFT)
- Observation over time reveals increasing deviations from normal values
- Predictions about quality and deterioration

„Crankshaft fatigue“ is licensed under CC BY-SA 3.0 Unported (https://creativecommons.org/licenses/by-sa/3.0/deed.en)

Real-Time Data Analytics

When do I analyze data?

Real-Time Data Analytics

When do I analyze data?

After an event has occurred? ⇒ Data at rest

or

Real-Time Data Analytics

When do I analyze data?

After an event has occurred? ⇒ Data at rest

or

While an event occurs? ⇒ Data in motion

Real-Time Data Analytics – Stream Processing

Problem:

- Processing of continuous data streams
- Answering question X without delay

Examples:

- E-commerce order processing
- Credit card fraud detection
- Spam detection
- etc.

Real-Time Data Analytics – Stream Processing Properties

- Results for question X based on current data
- Calculation on individual data set or for a small time window
- Optimized for low latency
- Calculations close to "real-time" and often very simple
- Example frameworks: Apache Storm or Amazon Kinesis

EXERCISE E05

E05 – IoT Security

1) In E03 and E04, you first gathered weather data from different sources (weather sensors/APIs) and aggregated them into a single data set. Subsequently, you processed the data to make predictions based on the results of the gathered information. However, so far, we have discarded the aspect of privacy and security when handling IoT-related data. Especially the transmission of data from the sensors to the processing entity (cloud, edge, etc.) is often prone to data manipulation. Moreover, the sensor data might contain sensitive information that is not meant to be public. Therefore, data in transit must be protected against manipulation and encrypted.

Questions?