Technische Hochschule Rosenheim
Technical University of Applied Sciences

Modul
- Internet of Things (IoT) -

04-Vorlesung

Prof. Dr. Marcel Tilly

Fakultät Informatik, Cloud Computing

Überblick

21. März	Einführung in das Internet der Dinge
28. März	IoT Architekturen
4. April	Things und Sensoren
11. April	From Device to Cloud
18. April	Vorlesungsfrei – Ostern
25. April	IoT Analytics
02. Mai	Big Data in IoT
9. Mai	Data Exploration
16. Mai	IoT Platformen
23. Mai	Entwicklung einer IoT Lösung
30. Mai	Vorlesungsfrei; Christi Himmelfahrt
05. Juni	opt. Gastvortrag – Digitalisierung
13. Juni	Data Science in IoT
20. Juni	Vorlesungsfrei – Fronleichnam
27. Juni	Intelligente Cloud und intelligente Edge
04. Juli	PStA Abschlusspraesentationen

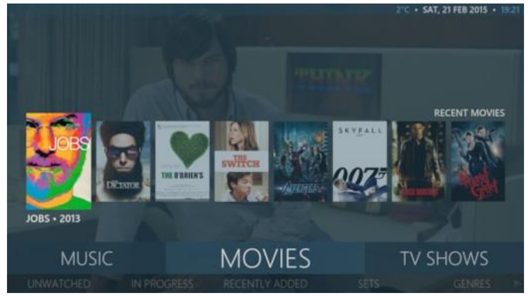
PStA

What is Raspberry Pi ?

- Arguably the most popular single board computer (SBC)
  - Easy to get started with because basically every problem is documented
- Add a computer with a OS to practically anything
  - NOTE: Do not expect it to perform as well as your laptop
- Support for a vast array of peripherals (thanks to the Linux kernel)
  - USB devices, networking, displays, cameras, audio etc.

What can you do with a Pi? (not so cool)

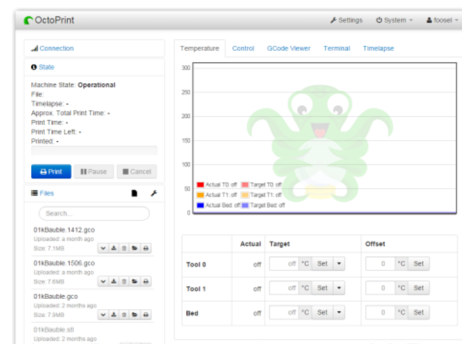
Technische Hochschule Rosenheim
Technical University of Applied Sciences



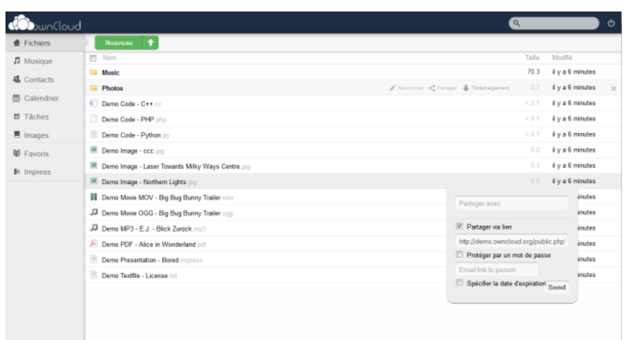
http://mymediaexperience.com/raspberry-pi-xbmc-with-raspbmc/

https://www.raspberrypi.org/magpi/magic-mirror/

https://learn.adafruit.com/pigrrl-2/overview



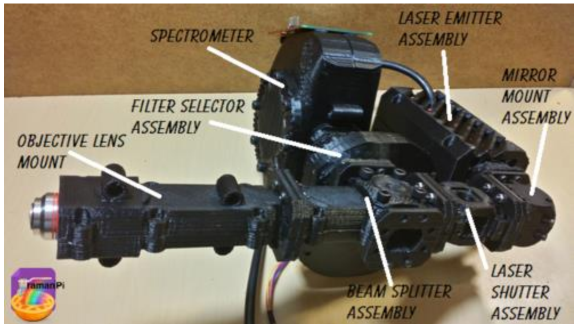
https://github.com/foosel/OctoPrint



https://vadelmapii.com/blogi/yllapida-omaa-dropbox-kloonia-raspberry-pilla-kayttaen-owncloudia

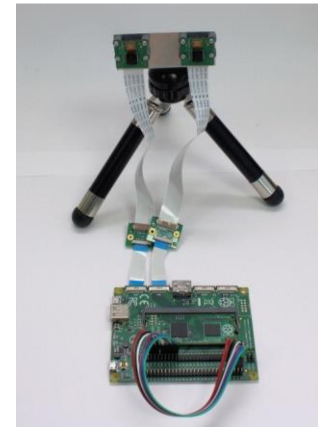
What can you do with a Pi? (cool)

Technische Hochschule Rosenheim
Technical University of Applied Sciences



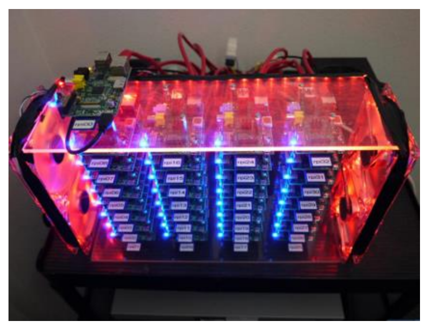
https://hackaday.io/project/1279-ramanpi-raman-spectrometer

https://www.raspberrypi.org/blog/real-time-depth-perception-with-the-compute-module/

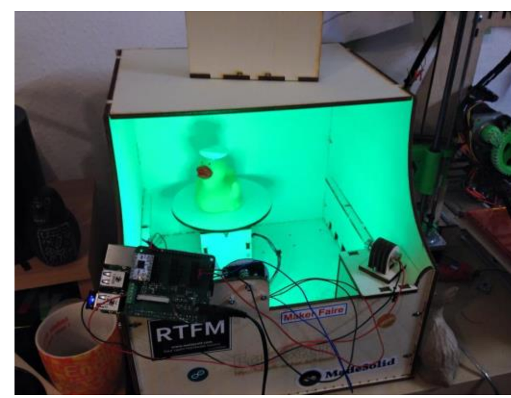


https://www.raspberrypi.org/blog/real-time-depth-perception-with-the-compute-module/

https://hackaday.io/project/1269-mashberry-beer-brewing-with-raspberry-pi



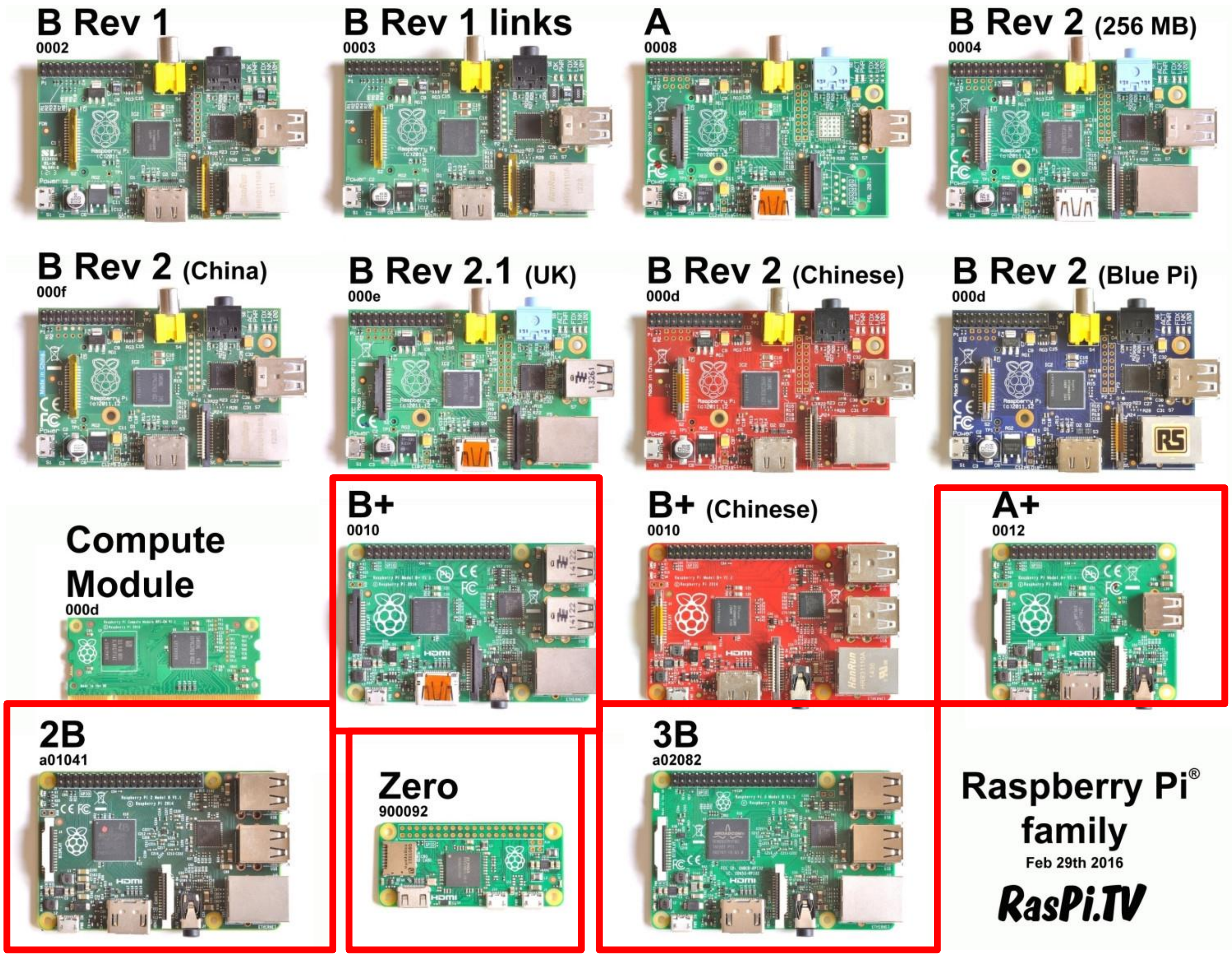
http://www.zdnet.com/article/build-your-own-supercomputer-out-of-raspberry-pi-boards/



https://www.raspberrypi.org/magpi/fabscan-pi-project-3d-scanning-for-all/

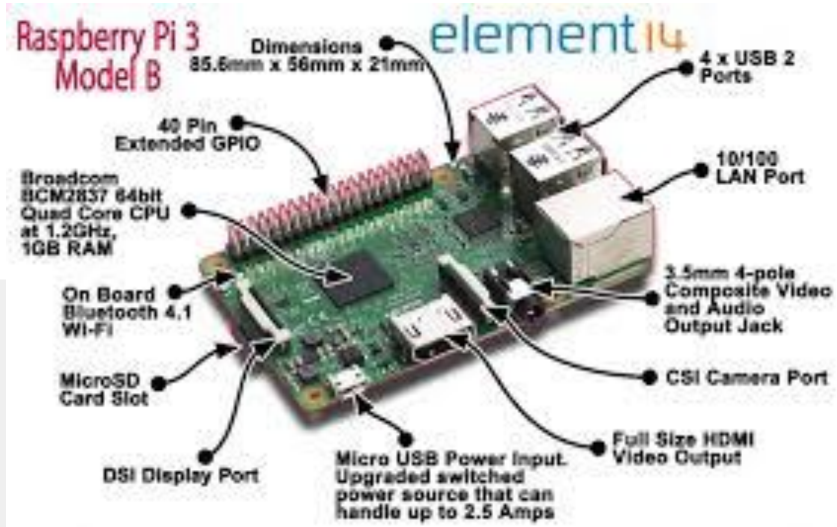
Raspberry Pi models

Technische Hochschule Rosenheim
Technical University of Applied Sciences



Plattform: Raspberry Pi

Technische Hochschule Rosenheim
Technical University of Applied Sciences



	Raspberry Pi 3 Model B	Raspberry Pi Zero	Raspberry Pi 2 Model B	Raspberry Pi Model B+
Introduction Date	2/29/2016	11/25/2015	2/2/2015	7/14/2014
SoC	BCM2837	BCM2835	BCM2836	BCM2835
CPU	Quad Cortex A53 @ 1.2GHz	ARM11 @ 1GHz	Quad Cortex A7 @ 900MHz	ARM11 @ 700MHz
Instruction set	ARMv8-A	ARMv6	ARMv7-A	ARMv6
GPU	400MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV
RAM	1GB SDRAM	512 MB SDRAM	1GB SDRAM	512MB SDRAM
Storage	micro-SD	micro-SD	micro-SD	micro-SD
Ethernet	10/100	none	10/100	10/100
Wireless	802.11n / Bluetooth 4.0	none	none	none
Video Output	HDMI / Composite	HDMI / Composite	HDMI / Composite	HDMI / Composite
Audio Output	HDMI / Headphone	HDMI	HDMI / Headphone	HDMI / Headphone
GPIO	40	40	40	40
Price	$35	$5	$35	$35

Raspberry Pi advantages and disadvantages

Technische Hochschule Rosenheim
Technical University of Applied Sciences

+

- Cheap (price per performance)
- Well documented
- Availability
  - Also in terms of add-ons (HATs)
- Versatile
- Compact (especially Zero)

−

- Scary linux (learning required)
- Not real time*
- No ADC (easy to add though)
- PWM possible but limited frequency

* It is possible to install a RTOS on the Pi

OS options

Technische Hochschule Rosenheim
Technical University of Applied Sciences

NOOBS

RASPBIAN

WINDOWS 10 IOT CORE

RISC OS

A non-Linux distribution

with Pixel or Lite

RetroPie

archlinux ARM

pidora

OSMC

LIBREELEC

UBUNTU MATE

SNAPPY UBUNTU CORE

Installation process

Get SD card (micro on newer Pi’s)
Format to FAT32 (for example with SD card formatter)

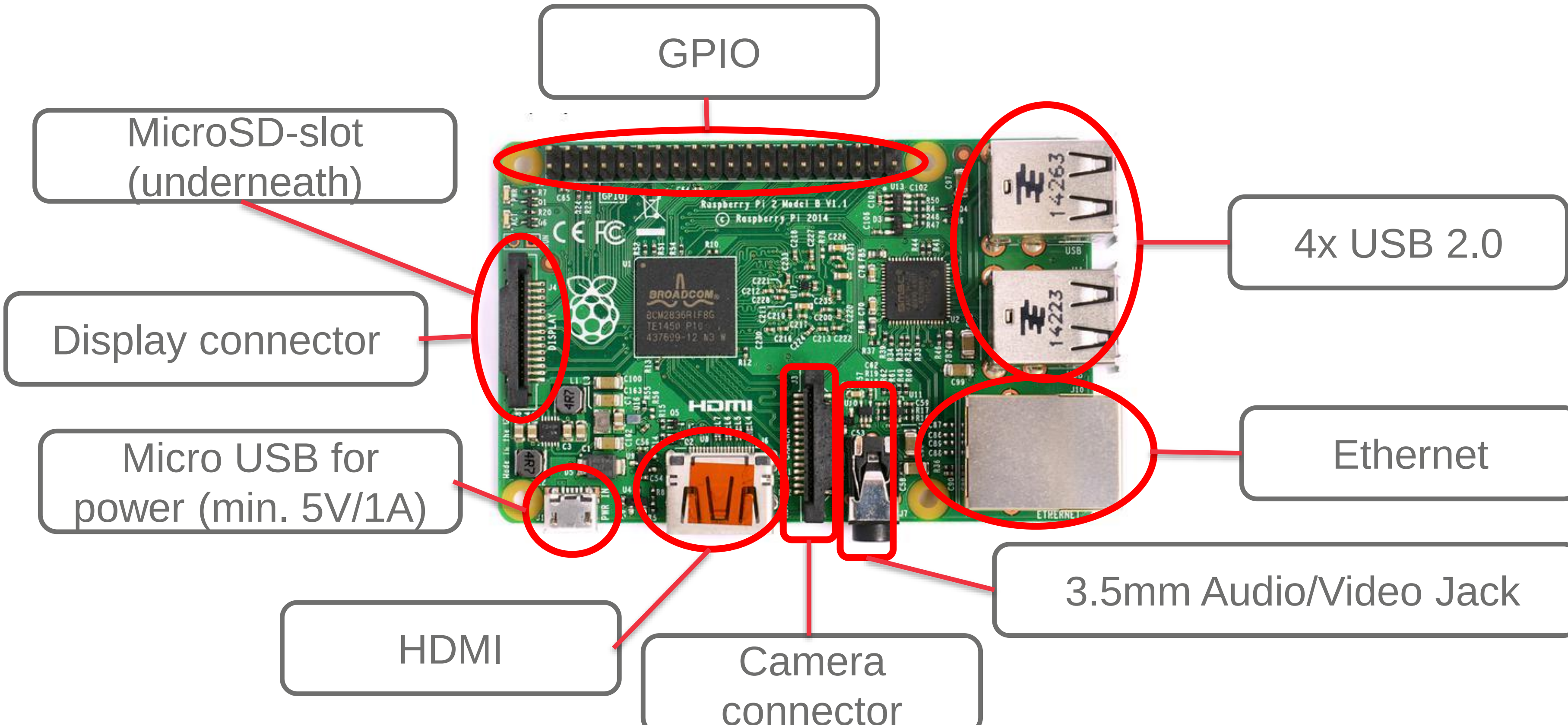
Using NOOBS:

- Download NOOBS (sd-cards with pre-installed NOOBS can purchased)
- Unzip and copy all contents to SD card (takes a while)
- Boot

Using a disk image writer:

- Download the disk image (.img)
- Write the disk image on the SD card (“hard drive”)
  - Windows: Win32 disk image writer or Etcher
  - MacOSX / Linux: Use dd (or image writer with GUI such as Etcher)
- Boot

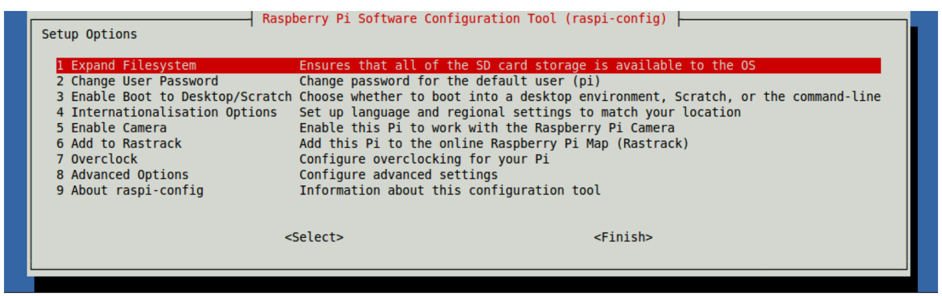
Connections

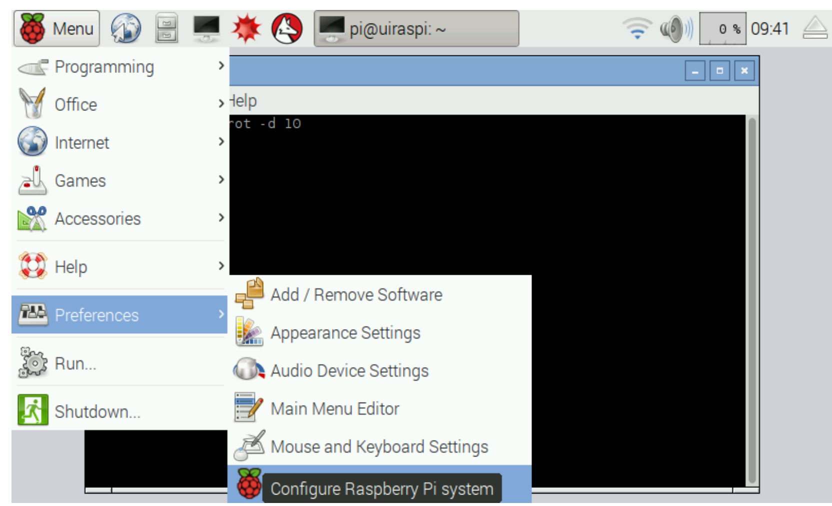


On first boot…

Technische Hochschule Rosenheim
Technical University of Applied Sciences

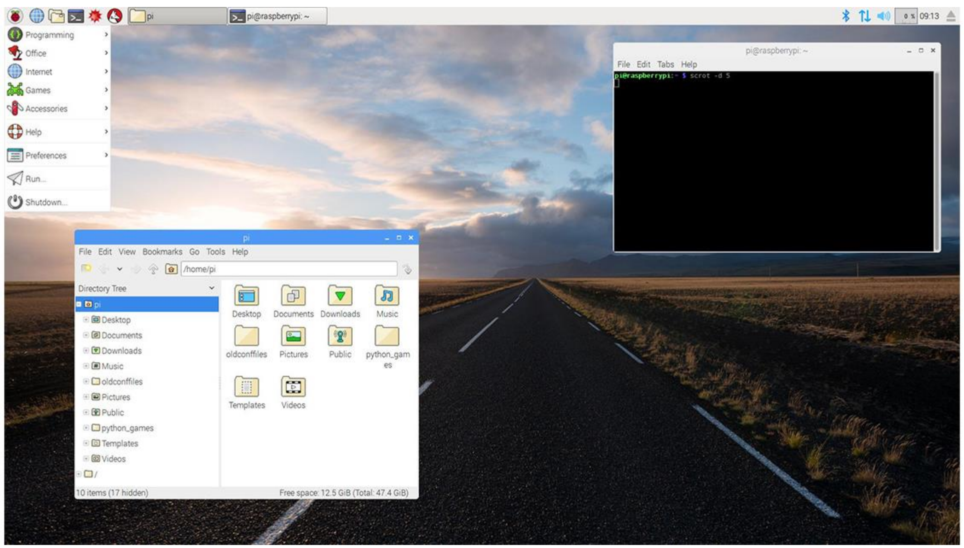
- NOOBS installer has GUI (Raspbian recommended)
- Boots into raspi-config (you can run it with “sudo raspi-config”)
- Expand file system, change password and change keyboard layout, enable ssh etc.

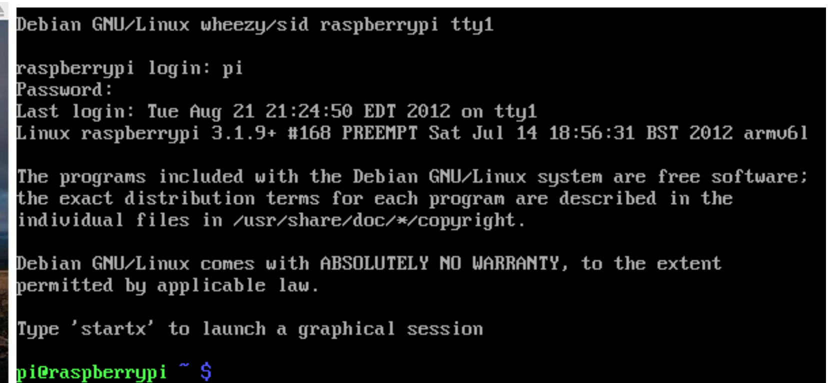




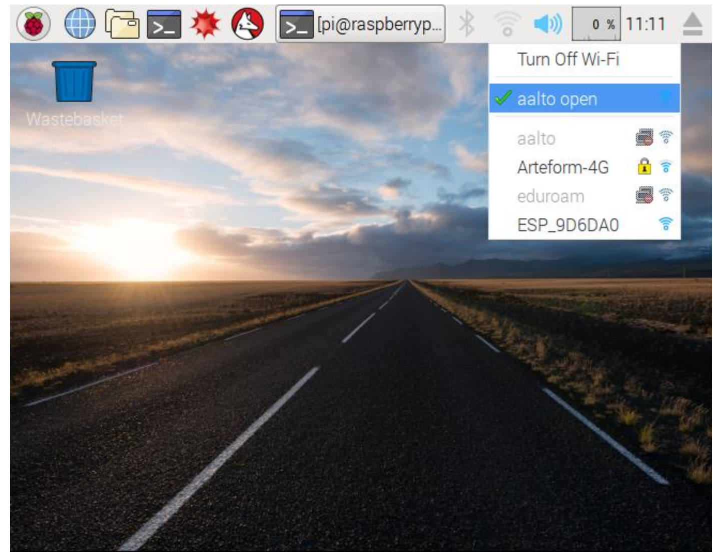
Terminal or Pixel

There are two options when booting: shell or Pixel desktop (graphical session)





Connect to WLAN



In GUI:
- Choose WiFi network and connect

Without GUI :
- sudo iwlist wlan0 scan // scan for networks
- sudo nano /etc/wpa_supplicant/wpa_supplicant.conf // open wpa_supplicant configuration
- Add this to the bottom of the file

```
network={
        ssid="aalto open"
        proto=RSN
        key_mgmt=NONE
}
```

Data Storage

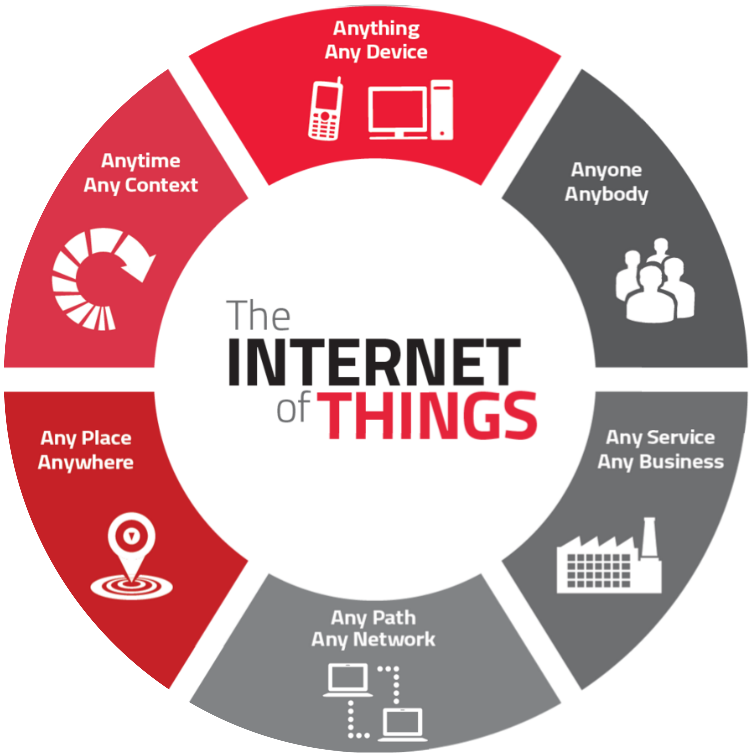
Challenges

Amount of Data (Big Data) - Volume
Speed of Data (Data Rate) - Velocity
Data format (semantics) - Variety
Data trust
Data quality
DSVGO (privacy)
Encryption (Transport security)
Compression
Data Selection

Technology

SQL
Map-and-Reduce (Hadoop)
Stream Processing (Spark)

IoT Meets Big Data



When does data become Big Data

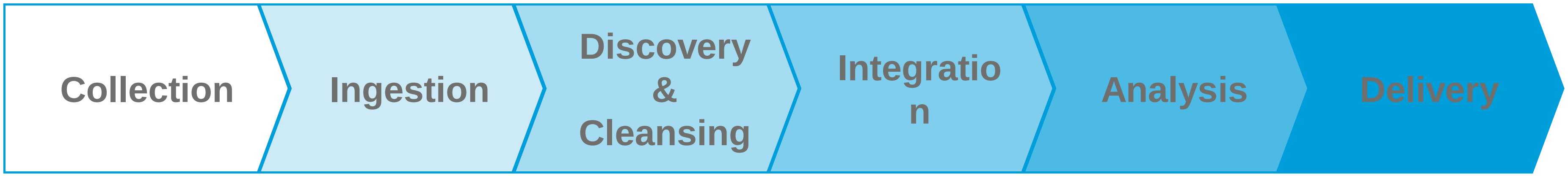
Technische Hochschule Rosenheim
Technical University of Applied Sciences

Big data is generally defined by its “three Vs”: volume, velocity, and variety. Thinkers in the field have occasionally argued for adding additional “Vs” (such as veracity and value), but the classic three Vs provide a nice overview of what defines big data.

- **Volume** is the “big” in big data. Current volumes of data (4 zettabytes worldwide this year and growing) dwarf anything we have had to tackle in human history.
- **Velocity** refers to the rate at which big data needs to be processed. This is both a factor of how quickly new data gets created, but also that we often want to look at live data streams for fresh data, like Twitter feeds for sentiment analysis or real-time Internet of Things (IoT) sensor data streams.
- **Variety** refers to unstructured (images, audio, and the like) or semi-structured (JSON documents).

Big Data Value Chain

Technische Hochschule Rosenheim
Technical University of Applied Sciences



Collection – Structured, unstructured and semi-structured data from multiple sources

Ingestion – loading vast amounts of data onto a single data store

Discovery & Cleansing – understanding format and content; clean up and formatting

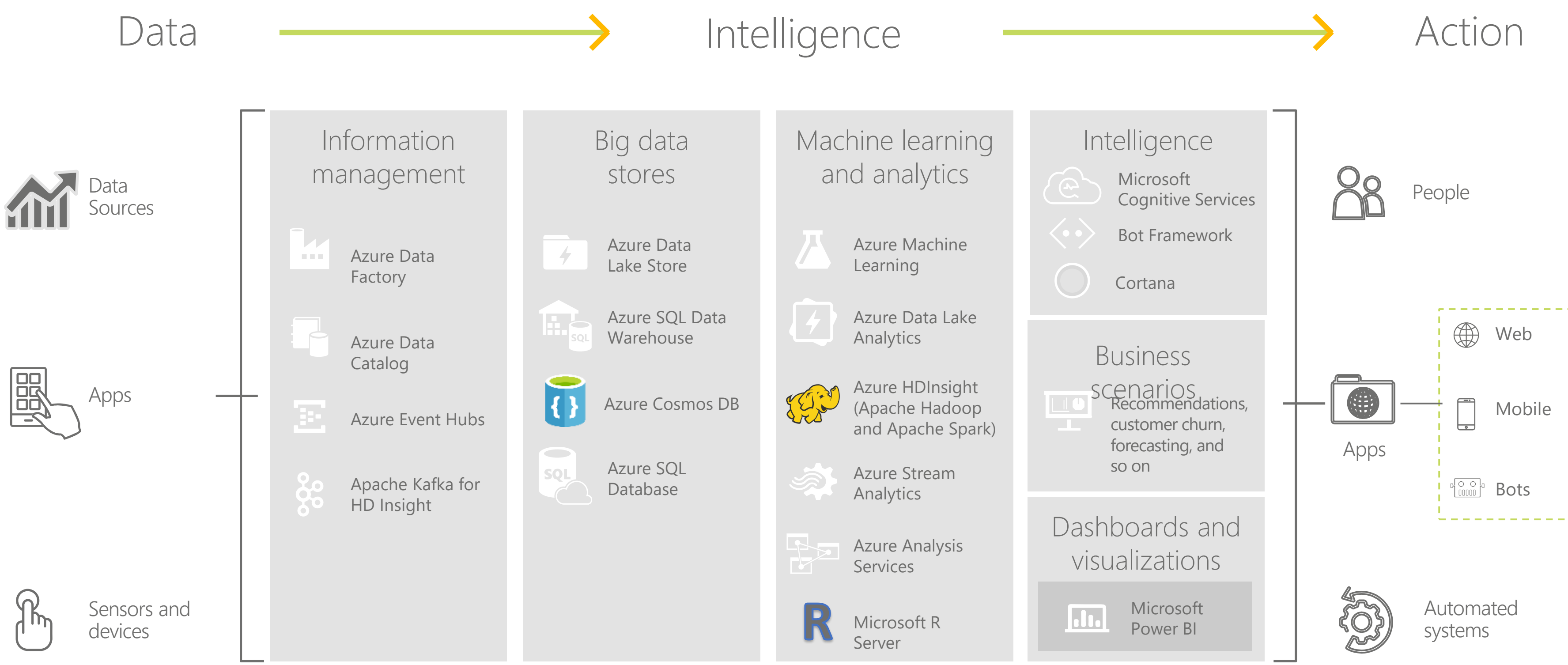
Integration – linking, entity extraction, entity resolution, indexing and data fusion

Analysis – Intelligence, statistics, predictive and text analytics, machine learning

Delivery – querying, visualization, real time delivery on enterprise-class availability

Source O'Reilly Strata 2012

Azure Big Data platform
Transform data into intelligent actions and predictions



From(data).To(insights)

Technische Hochschule Rosenheim
Technical University of Applied Sciences

3Vs: Volume, Variety, Velocity

Structured

SQL
SQL Azure
SQL Table

Semi structured

Hadoop
Map&Reduce
Data Lake

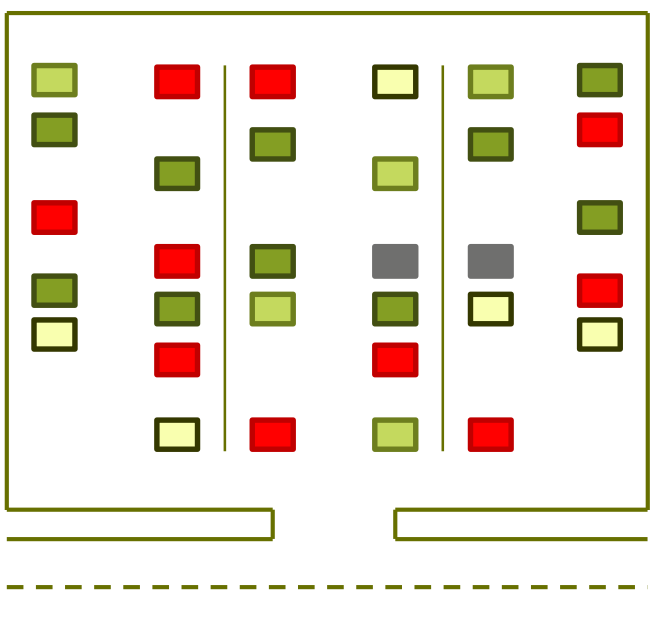
Event Data Streams

Low-latency
High-throughput
In-flight

Advanced Analytics (Machine Learning)

- Predictive Maintenance, Early-Fault-Detection, Idle Prediction

Understanding Streaming Data (1)

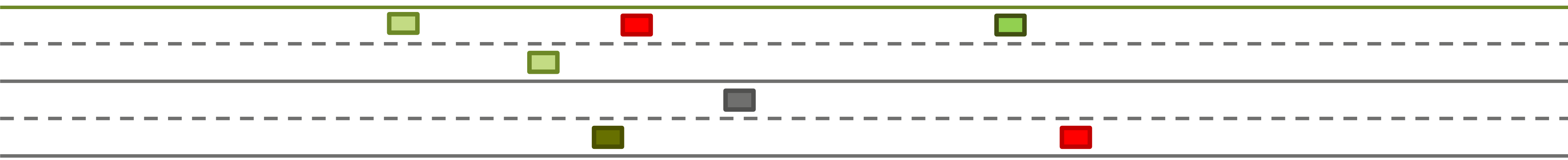


- **Question**: “How many red cars are in the parking lot?”

- Prerequisite: Data is static! (no cars leaving or entering!)
- Answering with a **relational database**:
  - Walk out to the parking lot
  - Count vehicles that are: ‘Red’ and ‘Cars’

```
(FROM vhcl in ParkingLot
 WHERE vhcl.type  = “CAR”
 &&    vhld.color = “RED”
 SELECT vhcl).Count()
```

Understanding Streaming Data (2)



“How many trucks **(red cars)** went from right to left in the last **10 seconds**”?

“How many **green cars (= electrical cars)** are passing every **60 minutes**”?

“What is the **running average speed** of cars on the left lane **compared** to the right lane?

This is the streaming data paradigm in a nutshell – ask questions about data in flight.

Data Stream in a Nutshell

Technische Hochschule Rosenheim
Technical University of Applied Sciences

- A data stream is a continuous sequence of data tuples
  - Think of standard tuples of relational databases
  - + time information (**timestamp = TS**)

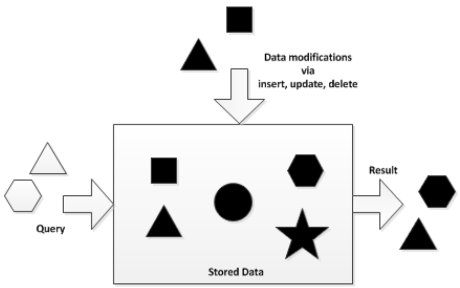
- **That means:**
  - **Data is moving**!
  - Continuously generated (assumed **infinite**!)
- Potentially **high pace.**
- System has to process data without first storing everything

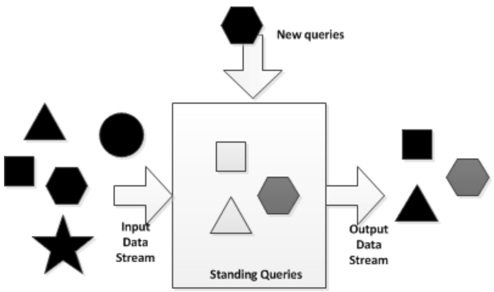
Data Stream Scenarios

- Sensor networks for environmental monitoring
  - Avalanche risk level computation
  - Insights for agriculture
  - Air pollution (urban) monitoring
- Real-time analysis of stock market changes
  - Computing statistics over streams, e.g., for decision support
  - Opportunities for reacting in real-time
  - Even with fully automated means: algorithmic trading
- Social Media Analysis
  - Sentiment analysis of products
  - News and trend analysis
- Industrial monitoring
  - Energy consumption and other key performance indicators (KPIs)
  - Predictive maintenance

Streaming Data Paradigm

	Database Applications	Data Streaming Applications
Concept	Persist data (&relations) (Data at Rest)	Volatile data streams (Data in Motion)
Strategy	Random access	Sequential access
Query Paradigm	Ad-hoc queries or requests	Continuous standing queries
Memory/Storage	(in theory) unlimited storage	Main memory limitations
Execution Time	Little or no time requirements (Seconds, hours, days)	Consideration of the order of the input (< 1 sec.)
Data Rate	Low rate (Hundreds of events/sec)	High rate (tens of thousands of events/sec or more)
Query Semantics	Declarative relational analytics	Declarative relational *and temporal* analytics
Accuracy	Assumes exact data	Assumes inaccurate data





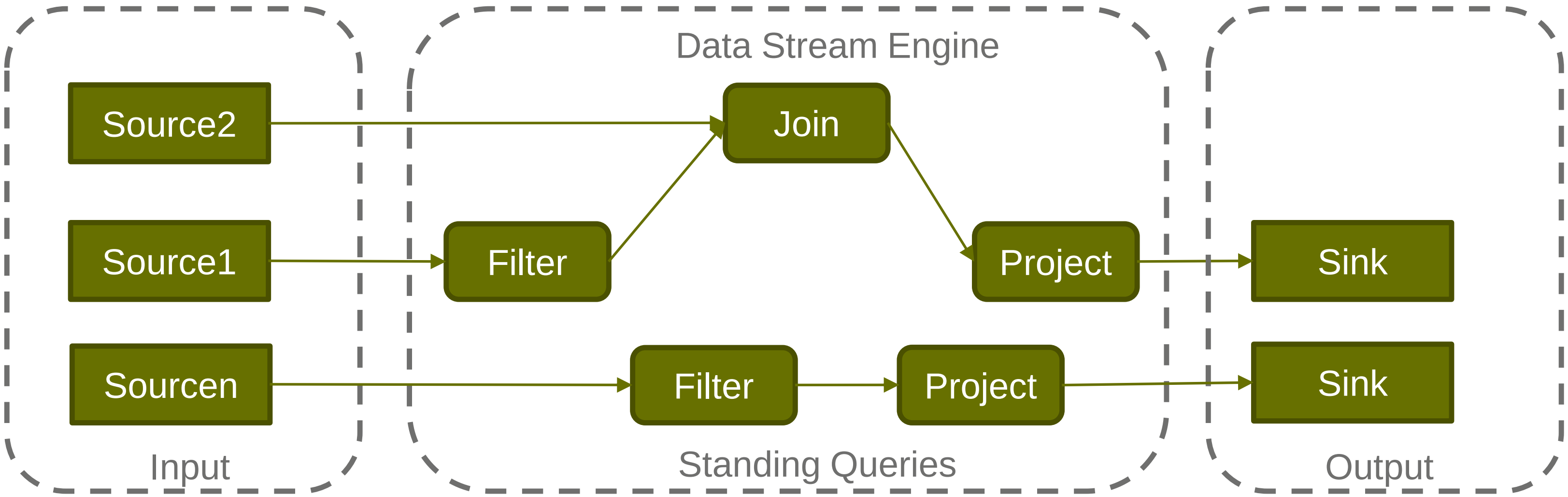
Data Stream Terminology

Term	Notation	Description	Technology Sample
Tuple/ payload	p = <k1:v1, k2:v2, …>	A un-ordered list of named elements (e.g. JSON)	
Event	e = <ts, p>	A timestamped tuple	
Stream	s	A stream is a possibly infinite sequence of events	
Source	src |> s	A source is emitting continuously events	Storm: Spout StreamInsight: IObservable
Operator	$S_{1,in}$ .. $S_{n,in}$|> o |> $s_{1,out}$…$S_{m,out}$	Processor of *n* input streams producing *m* output stream of events	Storm: Bolt StreamInsight: Function
Topology	G = (src[1..n], o, sk)	A graph of calculation represented as a network (= query) with *n* sources and 1 sink	
Sink	s |> sk	Consumes results of a Stream	StreamInsight: IObserver

Data Stream Topology

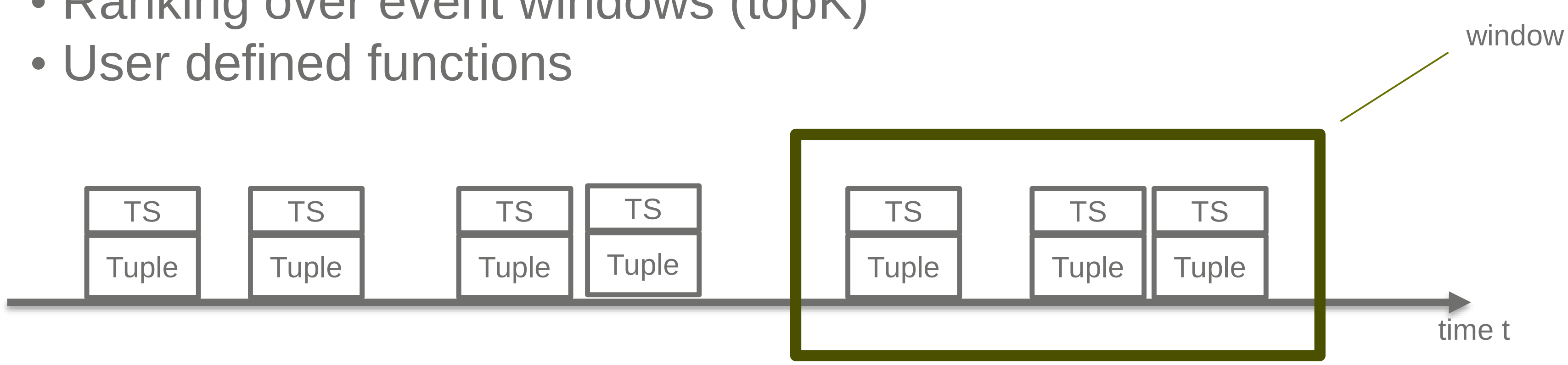
A standing query is an *instance* of a topology (=query graph)

A *data stream engine* handles heavy lifting for data streams



Query Expressiveness

- Stream Operations
  - Selection of events = filter
  - Calculations on the tuple payload = projection
  - Correlation of streams = join
  - Stream partitioning = grouping
- Window Operations
  - Aggregation (sum, count, min, max…) over event windows
  - Ranking over event windows (topK)
  - User defined functions



Query Expressiveness

Filter
Projection

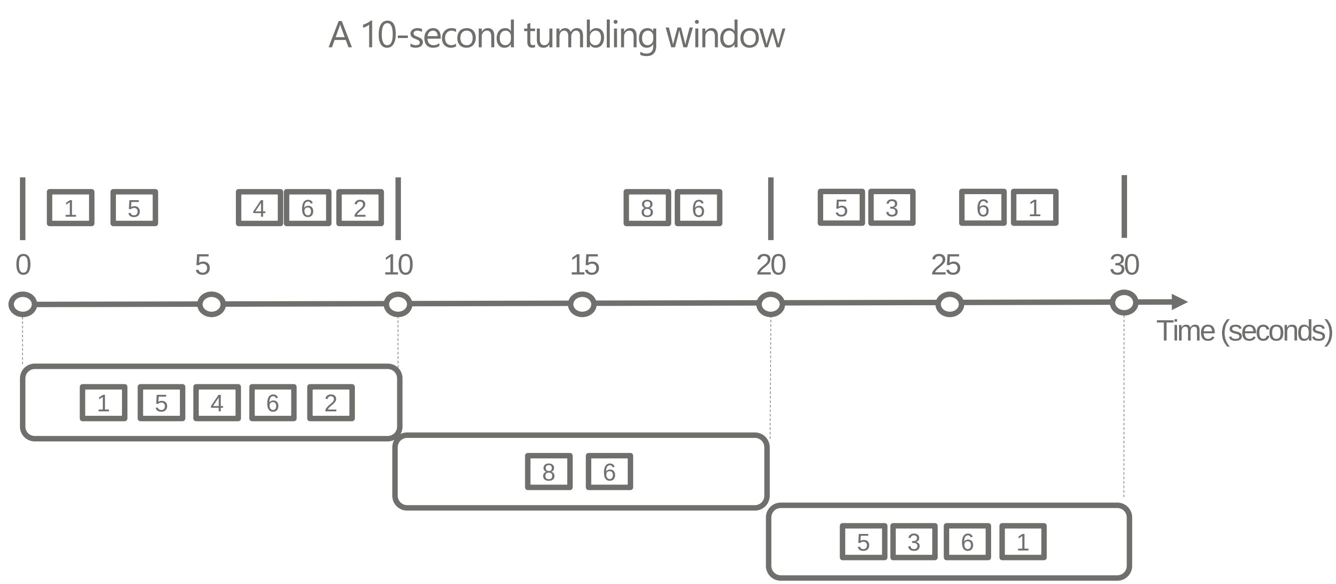
```
var result = from e in inputStream
        where e.id > 3
        select new {
                id = e.id,
                  W = (double)e.intW / 10 };
```

Correlation (Join)

Projection

```
var result = from eLeft in inputStream1
                join eRight in inputStream2
                on eLeft.id equals eRight.id
                select new {
                      id = eLeft.id,
                      diff = eLeft.W - eRight.w
};
```

Time Windows

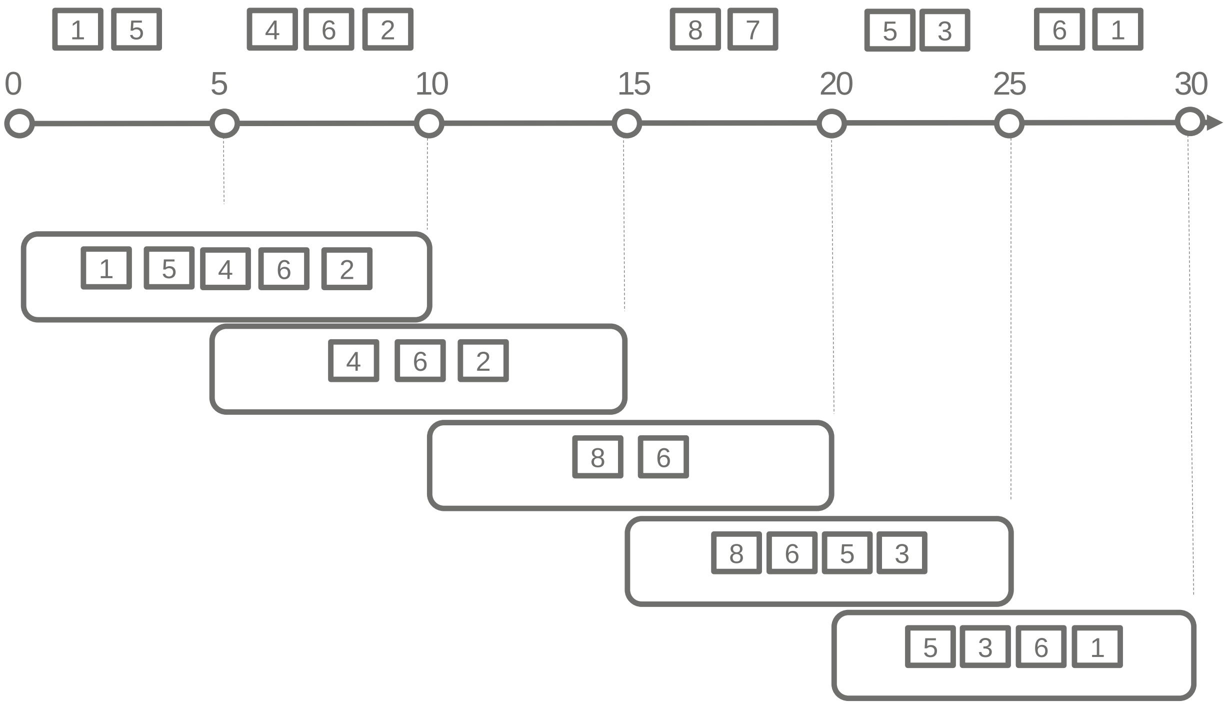


How many vehicles entered each toll both every 10 seconds?

```
SELECT TollId, COUNT(*) FROM EntryStream
TIMESTAMP BY EntryTime
GROUP BY TollId, TumblingWindow(second,10)
```

Time Windows

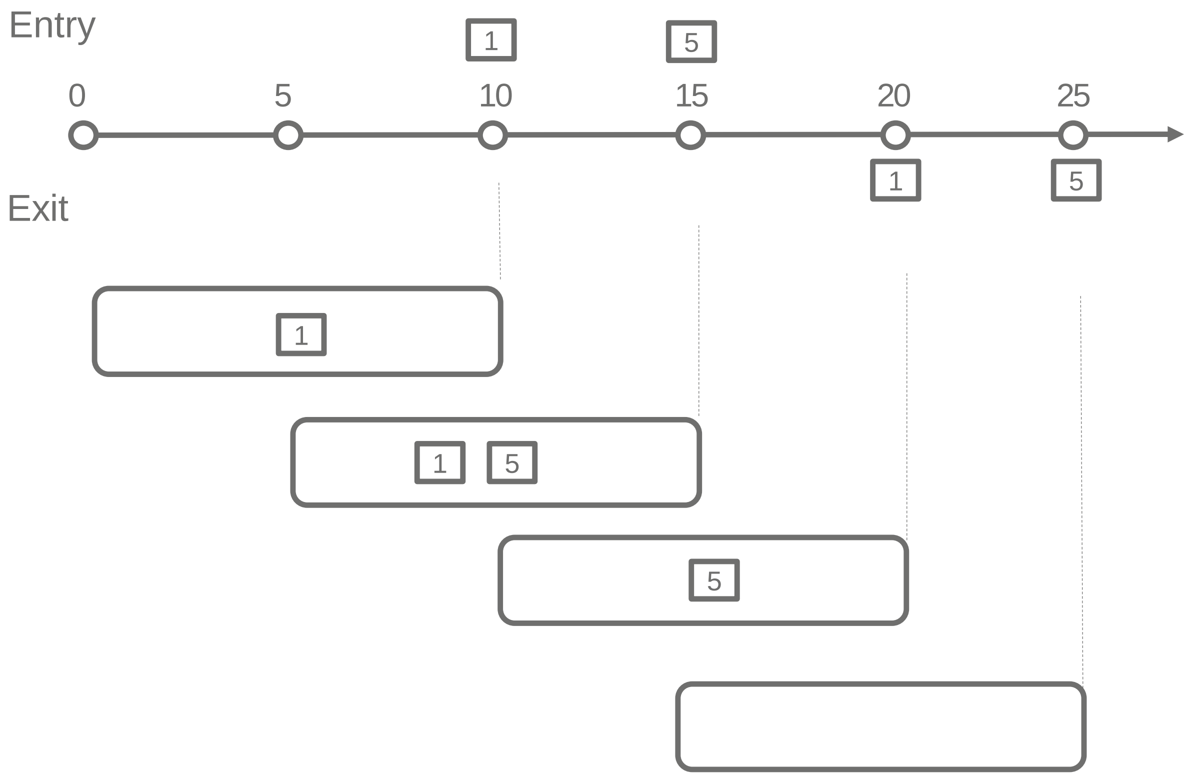
A 10-second Hopping Window with a 5-second "Hop"



Report every 5 seconds the total weight of cars that entered each toll both in the past 10 seconds

```
SELECT TollId, SUM(VehicleWeight) AS
TotalWeight
FROM EntryStream TIMESTAMP BY EntryTime
GROUP BY TollId, HoppingWindow(second, 10 , 5)
```

Windows

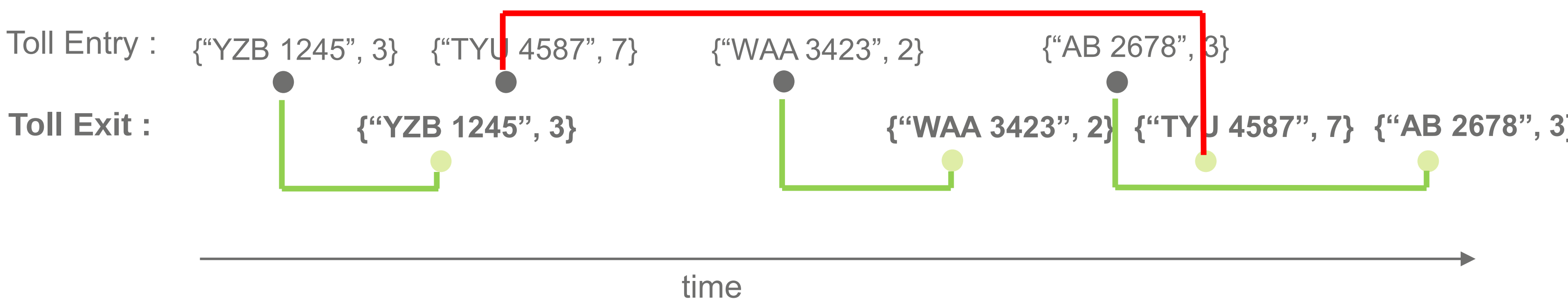


Report all toll booths which have served more than 3 vehicle in the last 10 seconds

```
SELECT TollId, Count(*)
FROM EntryStream TIMESTAMP BY EntryTime
GROUP BY TollId, SlidingWindow(second, 10)
HAVING COUNT(*) > 3
```

Joining multiple streams

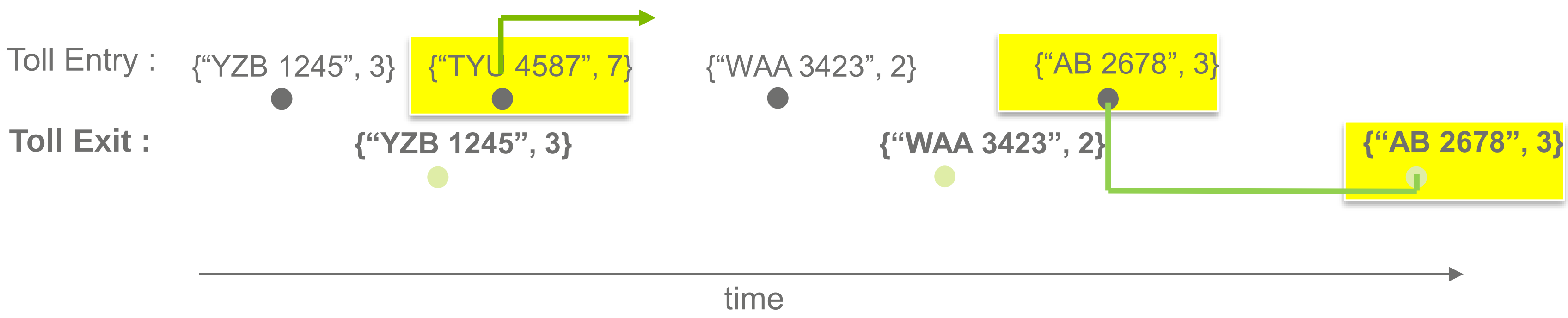
Calculate the time required for each car to pass the toll (in maximum 15 minutes)




```
SELECT EN.LicensePlate, DATEDIFF(minute, EN.EntryTime, EX.ExitTime) AS TotalTime
FROM EntryStream EN TIMESTAMP BY EntryTime
JOIN ExitStream EX TIMESTAMP BY ExitTime
     ON EN.TollId = EX.TollId AND EN.LicensePlate = EX.LicensePlate
          AND DATEDIFF(minute, EN, EX) BETWEEN 0 AND 15
```

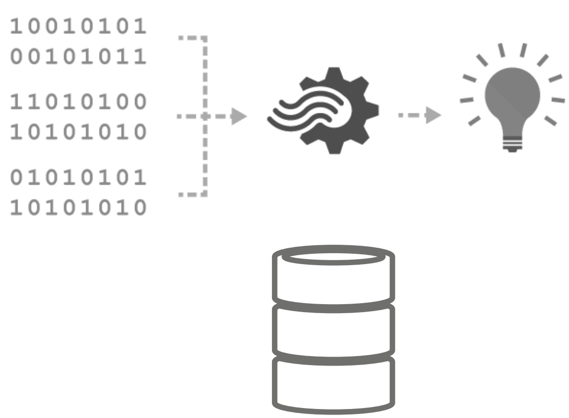
Detecting absence of events

Report all cars that did not pass the toll booth within 5 minutes




```
SELECT EN.LicensePlate, EN.TollId FROM EntryStream EN TIMESTAMP BY EntryTime
LEFT OUTER JOIN ExitStream EX TIMESTAMP BY ExitTime
   ON EN.TollId = EX.TollId AND EN.LicensePlate = EX.LicensePlate
      AND DATEDIFF(minute, EN, EX) BETWEEN 0 AND 5
WHERE EX.ExitTime IS NULL
```

Reference Data



Seamless correlation of event streams with reference data

Static or slowly-changing data stored in blobs

Scanned for changes on a settable cadence

JOIN (**INNER** or **LEFT OUTER**) between streams and reference data sources

Reference data appears like another input in the query

```
SELECT myRefData.Name, myStream.Value
FROM myStream
JOIN myRefData
      ON myStream.myKey = myRefData.myKey
```

Technologies

- **Apache Spark**: Fast and general engine for large-scale data processing
- **Apache Storm**: Distributed real-time computation system to reliably process unbounded streams of data
- **Apache Flink**: Scalable stream and batch processing
- **Apace Ignite**: In-memory computing platform

Commercial:

- Tibco Event Stream Processing
- IBM Streams
- Microsoft StreamInsight/ Azure Stream Analytics
- Amazon AWS Kinesis Analytics