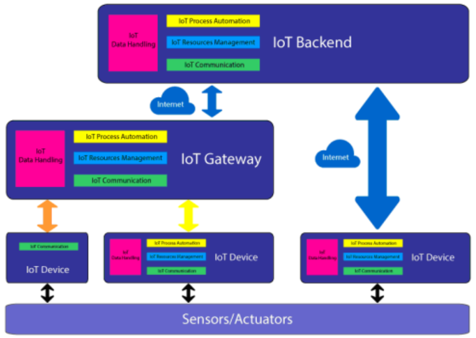
# IoT Lecture 6



https://www.in-jet.dk/en/articles.php?article_id=24

# Sensor Signal Sampling

Session 1: Data Acquisition and Reduction
- Sampling and Filtering
- Data Reduction and Analytics
- SAX

Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

Session 3: Data Storage
- SQL
- NoSQL

- Sensors measure physical phenomena
- Sensors issue noisy signals featuring a max frequency
- Continuous data is sampled to generate a (digitized) time series
- Asynchronous sampling must be greater than twice the highest signal frequency

- Exercise: How much storage is required for one second of 12-bit values, sample time 1ms?

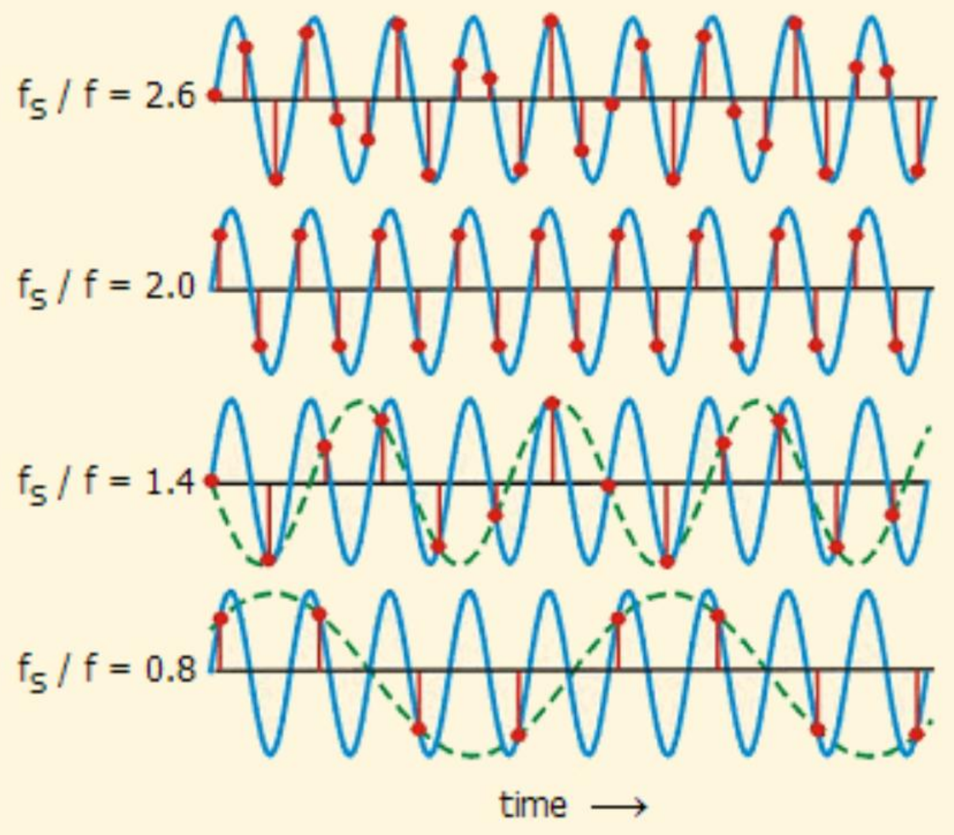



**Figure 1**. Sampling of a sinusoidal signal of frequency f at different sampling rates fS. With dashed lines are shown the alias frequencies, occurring when fS/f <2.

http://195.134.76.37/applets/AppletNyquist/Appl_Nyquist2.html

**Learning Aim 1**: The students will understand the principle of sampling and the Nyquist rule

# Sensor Signal Filtering and Reconstruction (1)


**Session 1: Data Acquisition and Reduction**
- **Sampling and Filtering**
- Data Reduction and Analytics
- SAX

Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

Session 3: Data Storage
- SQL
- NoSQL


- Process is:
- Condition the signal for sampling
  - HW filters for
    - High frequency noise reduction
    - Signal bandwidth limitations
- Signal capture per sample-and-hold
  - A/D conversion takes time and is noisy
- Make the time-series useful for analysis
  - **Smoothing** - filtering
  - **Data Reduction**

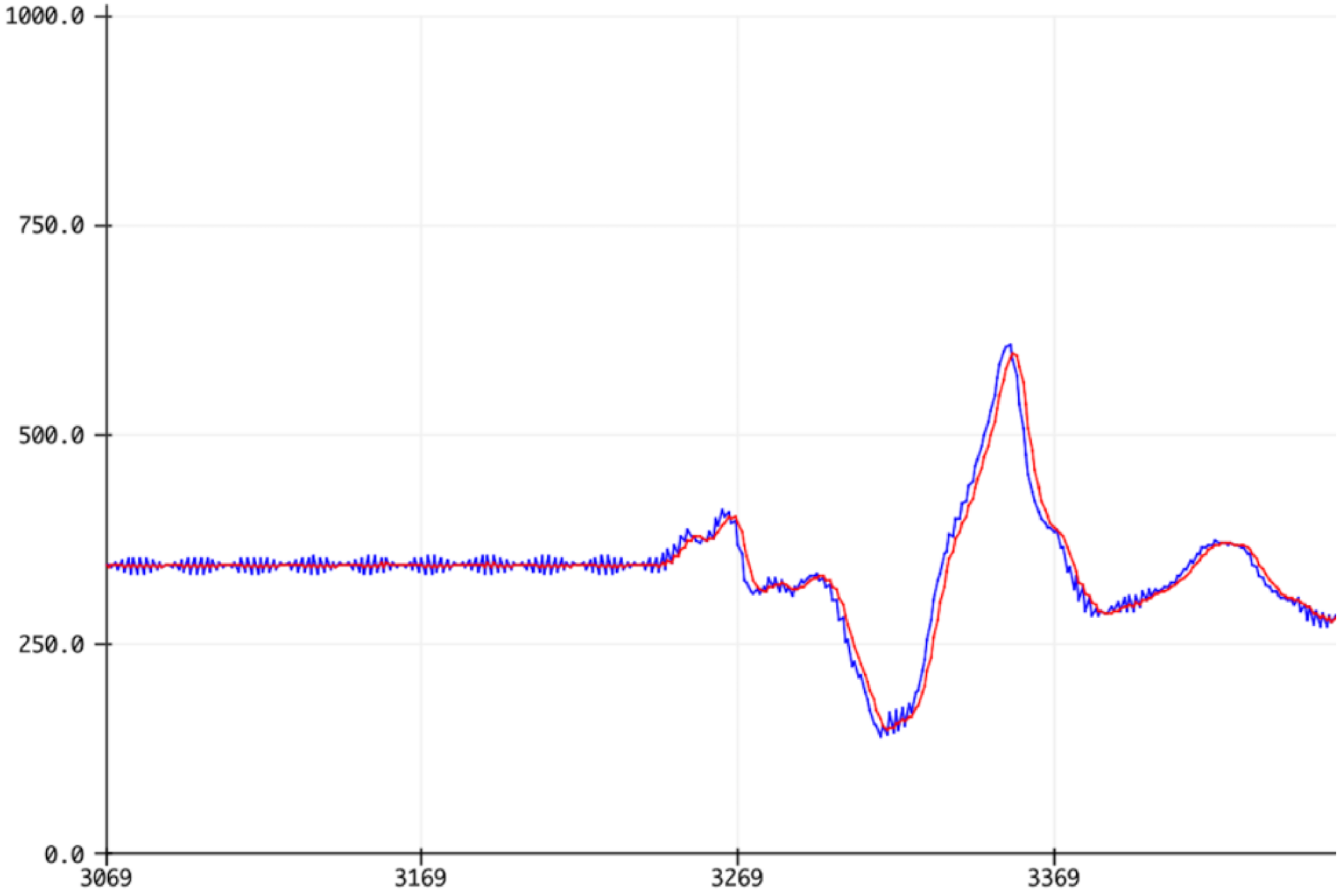
**Learning Aim 2**: The student will understand the necessity of filtering and/or smoothing

# Sensor Signal Filtering and Reconstruction (2)

**Session 1: Data Acquisition and Reduction**
- **Sampling and Filtering**
- Data Reduction and Analytics
- SAX

Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

Session 3: Data Storage
- SQL
- NoSQL




https://maker.pro/arduino/tutorial/how-to-clean-up-noisy-sensor-data-with-a-moving-average-filter

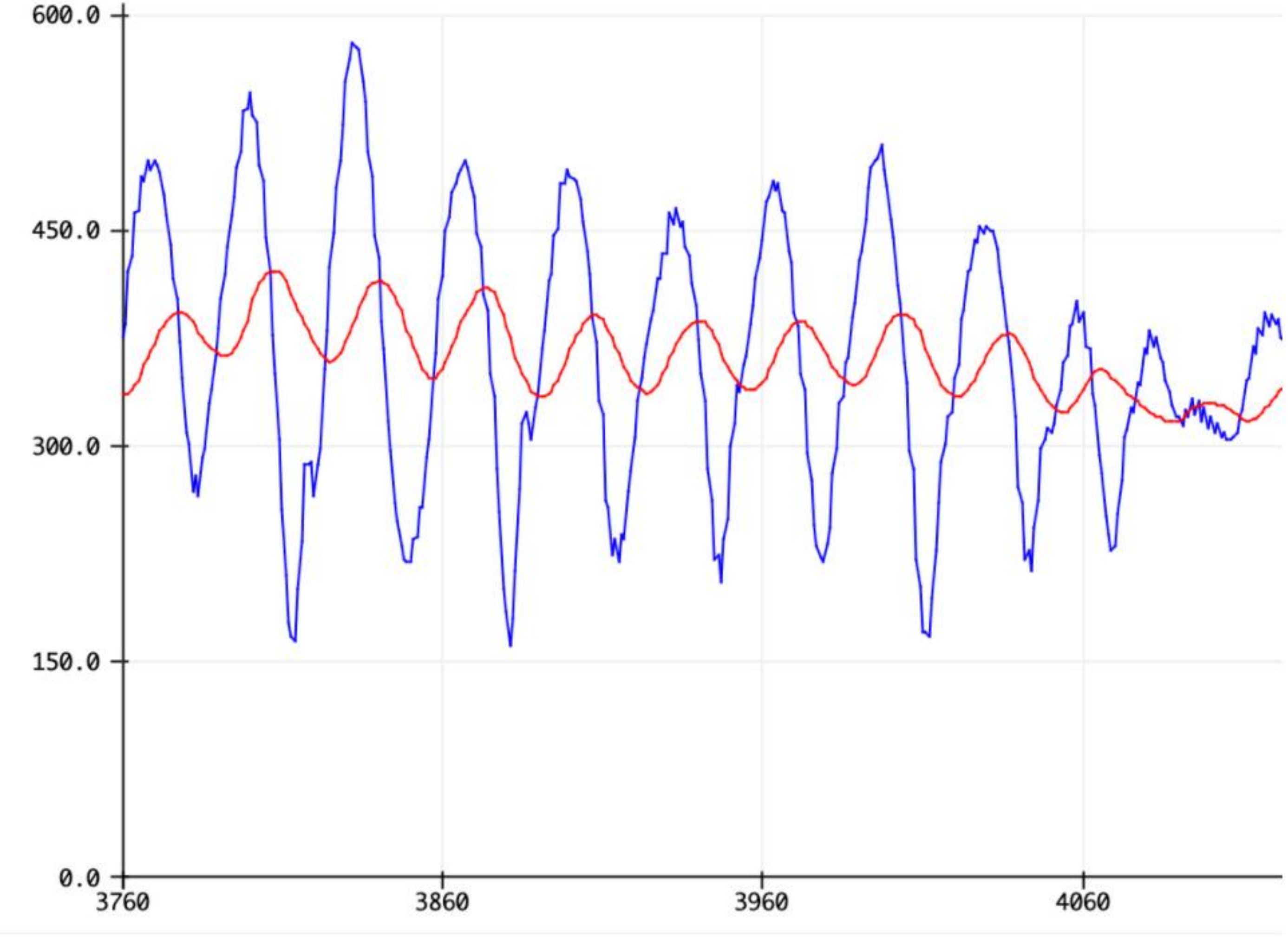
# Sensor Signal Filtering and Reconstruction (3)



**Session 1: Data Acquisition and Reduction**
- **Sampling and Filtering**
- Data Reduction and Analytics
- SAX

Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

Session 3: Data Storage
- SQL
- NoSQL




https://maker.pro/arduino/tutorial/how-to-clean-up-noisy-sensor-data-with-a-moving-average-filter

# Data Reduction

**Session 1: Data Acquisition and Reduction**
- Sampling and Filtering
- **Data Reduction and Analytics**
- SAX

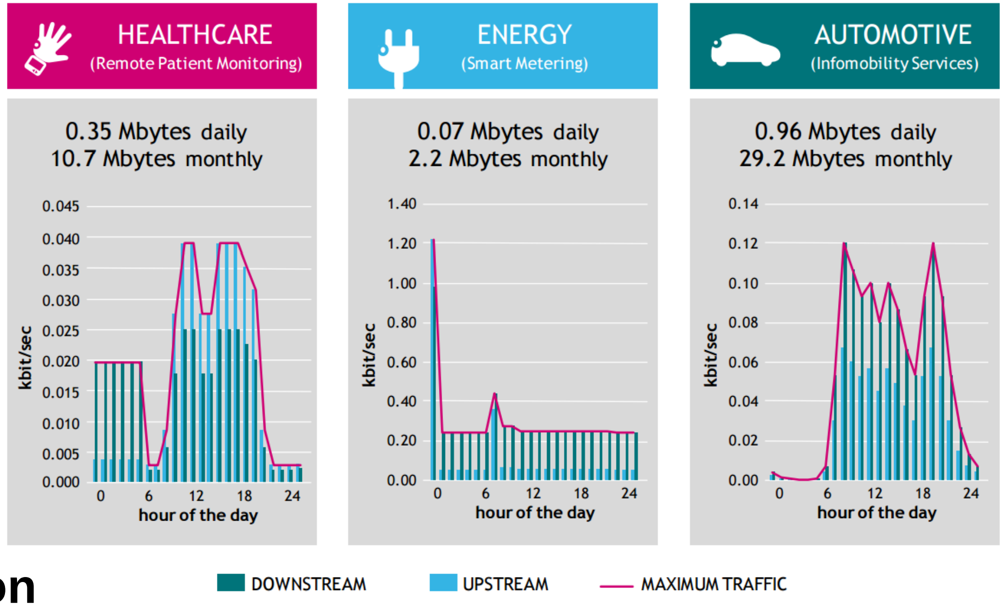
Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

Session 3: Data Storage
- SQL
- NoSQL

- Different applications have different data generation patterns
- This data needs to be stored to be useful – storage is associated with costs
  - **Data Reduction**
- When enough data has accumulated then analytics can be applied
  - Fourier/wavelet transforms
  - **Symbolic aggregate approximation**

**Learning Aim 3**: The student will understand the necessity of data reduction

Figure 4: M2M Traffic Patterns for Different Applications




http://www.onem2m.org/images/files/oneM2M-whitepaper-January-2015.pdf

# Data Reduction

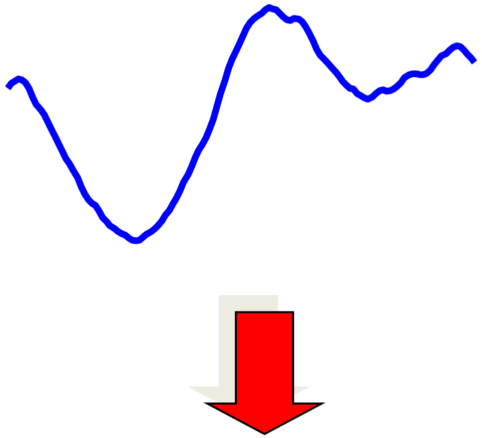
**Session 1: Data Acquisition and Reduction**
- Sampling and Filtering
- **Data Reduction and Analytics**
- SAX

Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

Session 3: Data Storage
- SQL
- NoSQL

- Symbolic Aggregate Approximation (SAX) is interesting because out of a time series it creates a symbolic string
  - Reduces the sheer quantity of data (data reduction)
  - Can use cheap string algorithms to find interesting stuff (data analytics)
  - Reasonable sensitivity and selectivity

**Learning Aim 1**: The students will be able to convert a time series into a SAX representation



baabccbc

www.cs.ucr.edu/~eamonn/**SAX**.ppt

# Data Analytics

**Session 1: Data Acquisition and Reduction**
- Sampling and Filtering
- **Data Reduction and Analytics**
- SAX

Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

Session 3: Data Storage
- SQL
- NoSQL

- Idea is to reduce a time series X of dimension n to a string of arbitrary length of dimension N where n >> N

- Steps are:
  - Normalise the time series (training set)
    - This allows us to compare apples with apples
    - Also the conversion into symbolic representation is dependent on the fact that a normalised time series tends towards a Gaussian distribution
  - Convert into a PAA (Piecewise Aggregate Approximation)
    - This performs the actual reduction in dimension
  - Convert the PAA representation into a string

# SAX (1)

**Session 1: Data Acquisition and Reduction**
- Sampling and Filtering
- Data Reduction and Analytics
- SAX

Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

Session 3: Data Storage
- SQL
- NoSQL

- Example:
- Time series (x): 2, 3, 4.8, 6, 7.1, 3, 2, 2, 1, 1
- Mean (μ): μ = (2 + 3 + 4.8 + 6 + 7.1 + 3 + 2 + 2 + 1 + 1)/10 = 3.19
- Standard Deviation (σ): = Sqrt(39.689/9) = 2.09997

$(2-3.19)^2 = 1.4161$

$(3-3.19)^2 = 0.0361$

$(4.8-3.19)^2 = 2.5921$

$(6-3.19)^2 = 7.8961$

$(7.1-3.19)^2 = 15.2881$

$(1-3.19)^2= 4.7961$

$$s = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \overline{x})^2}{N-1}}.$$

www.cs.ucr.edu/~eamonn/**SAX**.ppt

# SAX (2)

**Session 1: Data Acquisition and Reduction**
- Sampling and Filtering
- Data Reduction and Analytics
- **SAX**

Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

Session 3: Data Storage
- SQL
- NoSQL

- Time series (x): 2, 3, 4.8, 6, 7.1, 3, 2, 2, 1, 1
- Normalisation: $z_i = (x_i - \mu)/\sigma$ $\sigma = 2.09997$, $\mu = 3.19$

(2-3.19) / 2.09997 = -0.57
(3-3.19) / 2.09997 = -0.09
(4.8-3.19) / 2.09997 = 0.77
(6-3.19) / 2.09997 = 1.33
(7.1-3.19) / 2.09997 = 1.86
(1-3.19) / 2.09997 = -1.04

- Normalised time series =

-0.57, -0.09, 0.77, 1.33, 1.86, -0.09, -0.57, -0.57, -1.04, -1.04

www.cs.ucr.edu/~eamonn/**SAX**.ppt

# SAX (3)

**Session 1: Data Acquisition and Reduction**
- Sampling and Filtering
- Data Reduction and Analytics
- **SAX**

Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

Session 3: Data Storage
- SQL
- NoSQL

- Time series (x): 2, 3, 4.8, 6, 7.1, 3, 2, 2, 1, 1
- Normalised time series -> -0.57, -0.09, 0.77, 1.33, 1.86, -0.09, -0.57, -0.57, -1.04, -1.04
- PAA calculation  n = 10, N = 5

$$\bar{x}_i = \frac{N}{n} \sum_{j=\frac{n}{N}(i-1)+1}^{\frac{n}{N}i} x_j$$

- PAA representation: -0.328, 1.052, 0.885, -0.567, -1.04

http://www.ismll.uni-hildesheim.de/lehre/ba-12ss/script/ba05a.pdf

# SAX (4)

**Session 1: Data Acquisition and Reduction**
- Sampling and Filtering
- Data Reduction and Analytics
- **SAX**

Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

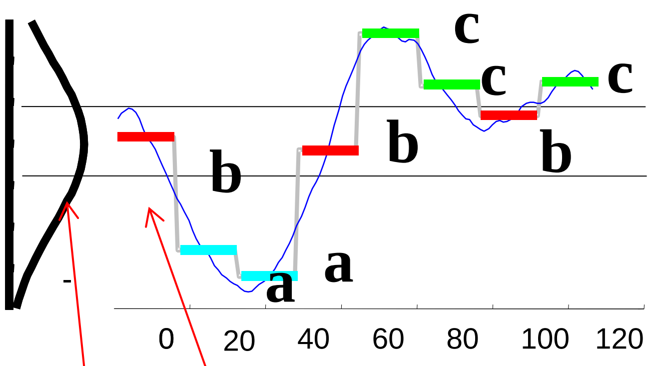
Session 3: Data Storage
- SQL
- NoSQL

- PAA representation: -0.328, 1.052, 0.885, -0.567, -1.04
- Now need to convert this into symbols

| $\beta_i$ \ $a$ | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|
| $\beta_1$ | -0.43 | -0.67 | -0.84 | -0.97 | -1.07 | -1.15 | -1.22 | -1.28 |
| $\beta_2$ | 0.43 | 0 | -0.25 | -0.43 | -0.57 | -0.67 | -0.76 | -0.84 |
| $\beta_3$ | | 0.67 | 0.25 | 0 | -0.18 | -0.32 | -0.43 | -0.52 |
| $\beta_4$ | | | 0.84 | 0.43 | 0.18 | 0 | -0.14 | -0.25 |
| $\beta_5$ | | | | 0.97 | 0.57 | 0.32 | 0.14 | 0 |
| $\beta_6$ | | | | | 1.07 | 0.67 | 0.43 | 0.25 |
| $\beta_7$ | | | | | | 1.15 | 0.76 | 0.52 |
| $\beta_8$ | | | | | | | 1.22 | 0.84 |
| $\beta_9$ | | | | | | | | 1.28 |

**Table 3:** A lookup table that contains the breakpoints that divide a Gaussian distribution in an arbitrary number (from 3 to 10) of equiprobable regions

.. three letter alphabet




Normalised time series assumed to have gaussian distribution

Where does one make the cut? ..

www.cs.ucr.edu/~eamonn/**SAX**.ppt

# SAX (5)

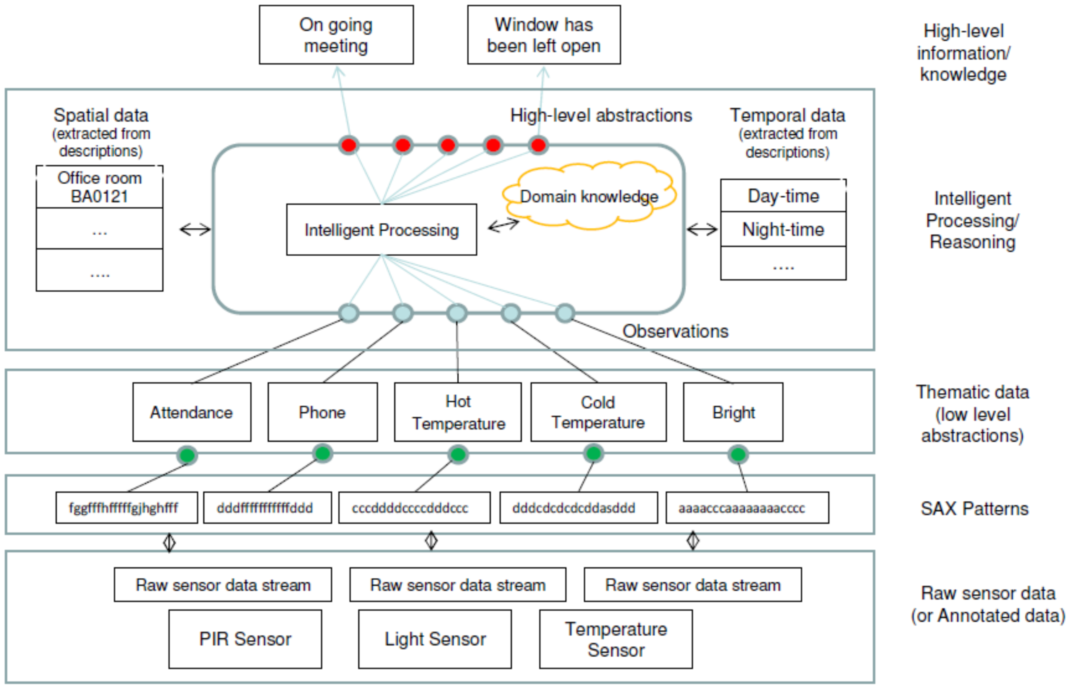


**Session 1: Data Acquisition and Reduction**
- Sampling and Filtering
- Data Reduction and Analytics
- **SAX**

Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

Session 3: Data Storage
- SQL
- NoSQL

- PAA representation: -0.328, 1.052, 0.885, -0.567, -1.04
- So for 4 letter alphabet cuts are {-0.67, 0, 0.67}
- SAX representation: bddba
- Cute …. And now?




Barnaghi, Foh, EEEM048- Internet of Things, Lecture 6- Intelligent Data Processing

# SAX (6)

**Session 1: Data Acquisition and Reduction**
- Sampling and Filtering
- Data Reduction and Analytics
- **SAX**

Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

Session 3: Data Storage
- SQL
- NoSQL

- PAA representation: -0.328, 1.052, 0.885, -0.567, -1.04
- So for 4 letter alphabet cuts are {-0.67, 0, 0.67}
- SAX representation: bddba
- Cute …. or

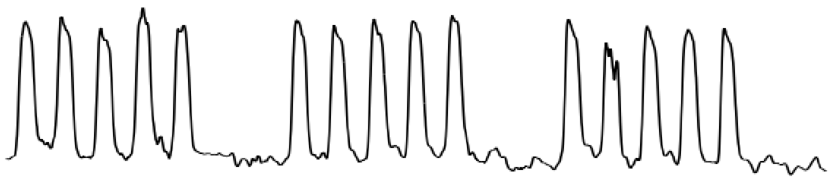


**Figure 5:** The first three weeks of the power demand dataset. Note the repeating pattern of a strong peak for each of the five weekdays, followed by relatively quite weekends

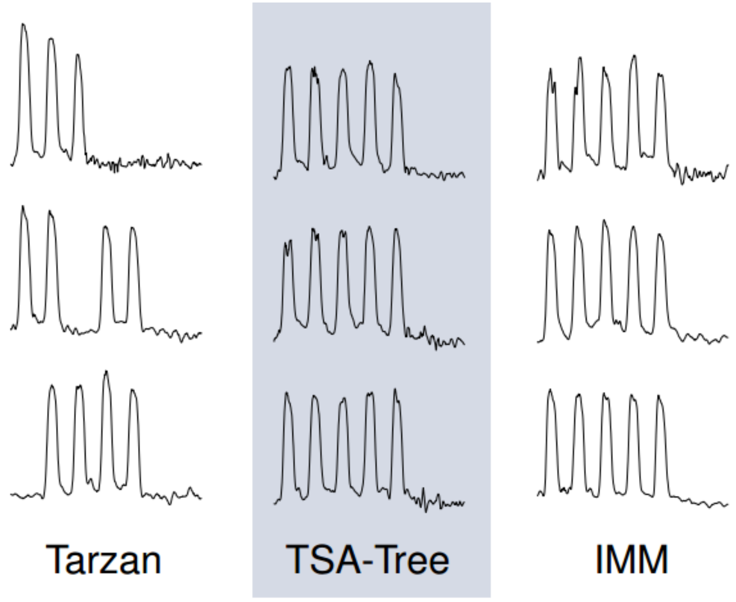



**Figure 6:** The three most surprising weeks in the power demand dataset, as determined by Tarzan, TSA-Tree and IMM

http://www.cs.ucr.edu/~eamonn/sigkdd_tarzan.pdf

# SAX Summary

**Session 1: Data Acquisition and Reduction**
- Sampling and Filtering
- Data Reduction and Analytics
- **SAX**

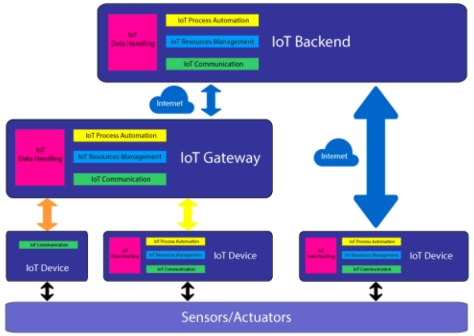
Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

Session 3: Data Storage
- SQL
- NoSQL

- SAX is a fast way of reducing the size of data sets whilst keeping them useful and allowing analytics to discover patterns using standard algorithms

- Exercise: With the following data: 2,3,5,0,1,3,2,0
  - Calculate: normalised time series
  - Calculate: PAA with N = 4
  - Calculate SAX alphabet of 3

# Session 2: IoT Data Handling




**Learning Aim:** The students will be able to explain the principle of application partitioning and its application to common IoT scenarios

https://www.in-jet.dk/en/articles.php?article_id=24

# IoT Scope

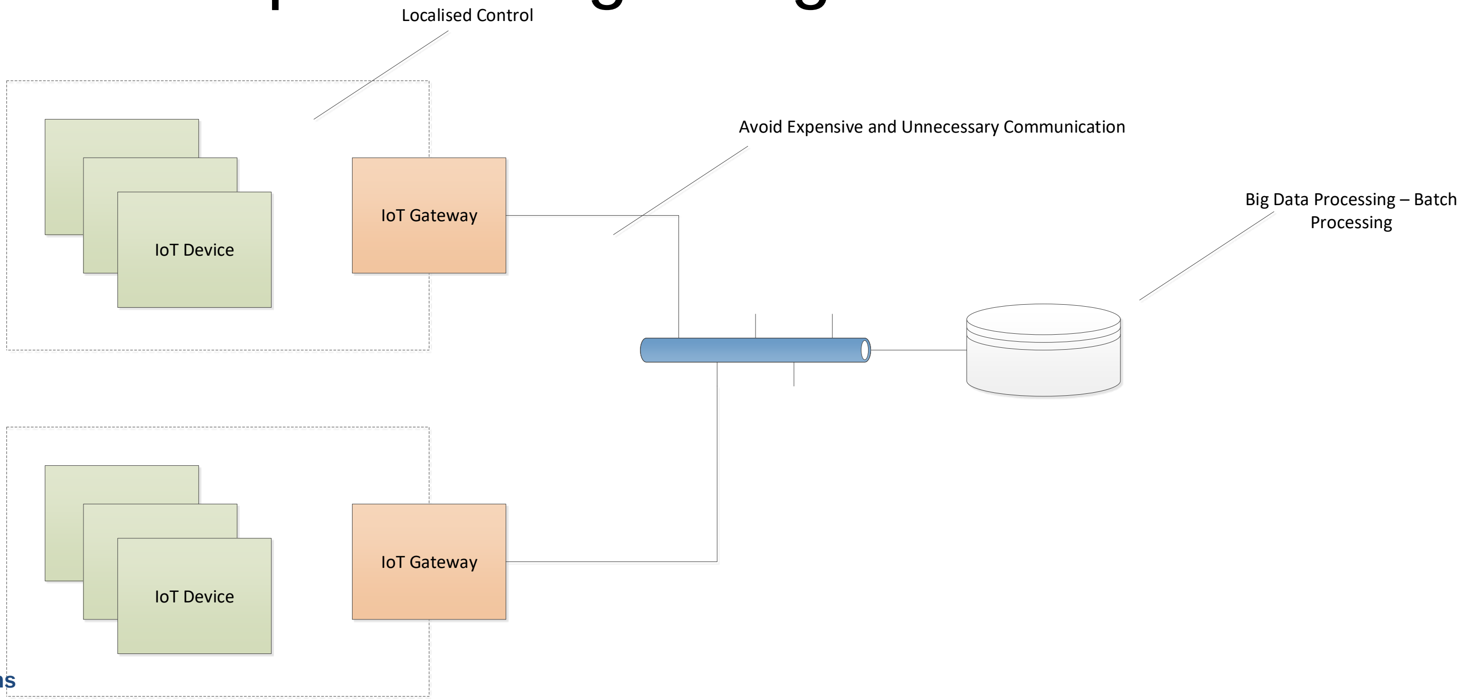


Session 1: Data Acquisition and Reduction
- Sampling and Filtering
- Data Reduction and Analytics
- SAX

**Session 2: Data Handling**
- **Communication Costs**
- Scenarios
- Stream Analytics

Session 3: Data Storage
- SQL
- NoSQL

- IoT Systems have a large scope
  - Localised control
  - Communication to some mass storage media
  - Batch processing of big data



# Data Formats: SenML (Sensor markup Language)

Session 1: Data Acquisition and Reduction
- Sampling and Filtering
- Data Reduction and Analytics
- SAX

**Session 2: Data Handling**
- **Communication Costs**
- Scenarios
- Stream Analytics

Session 3: Data Storage
- SQL
- NoSQL

- Draft specification SenML
  - https://datatracker.ietf.org/doc/draft-jennings-core-senml/
  - Specifies XML, EXI and JSON formats

```
{"bn": "http://[2001:db8::1]",
 "bt": 1320067464,
 "bu": "%RH",
 "e": [
  { "v": 20.0, "t": 0 },
  { "sv": "E 24' 30.621", "u": "lon", "t": 0 },
  { "sv": "N 60' 7.965", "u": "lat", "t": 0 },
  { "v": 20.3, "t": 60 },
  { "sv": "E 24' 30.622", "u": "lon", "t": 60 },
  { "sv": "N 60' 7.965", "u": "lat", "t": 60 },
  { "v": 20.7, "t": 120 },
  { "sv": "E 24' 30.623", "u": "lon", "t": 120 },
  { "sv": "N 60' 7.966", "u": "lat", "t": 120 },
  { "v": 98.0, "u": "%EL", "t": 150 },
  { "v": 21.2, "t": 180 },
  { "sv": "E 24' 30.628", "u": "lon", "t": 180 },
  { "sv": "N 60' 7.967", "u": "lat", "t": 180 }]
}
```

Base Name

Base Time (Unix)

Base Unit

Measurement or parameters

String Value

Value

Offset from Base Time

Unit

**Learning Aim:** The students will be able to explain, code and decode the format of a SenML document

https://tools.ietf.org/pdf/draft-jennings-core-senml-01.pdf

# Data Formats: SenML Exercise



Session 1: Data Acquisition and Reduction
- Sampling and Filtering
- Data Reduction and Analytics
- SAX

**Session 2: Data Handling**
- **Communication Costs**
- Scenarios
- Stream Analytics

Session 3: Data Storage
- SQL
- NoSQL

- Decode this:


```
{"bn": "urn:dev:mac:0024befffe804ff1/",
 "bt": 1276020076,
 "bu": "A",
 "ver": 1,
 "e":[
     { "n": "voltage", "u": "V", "v": 120.1 },
     { "n": "current", "t": -5, "v": 1.2 },
     { "n": "current", "t": -4, "v": 1.30 },
     { "n": "current", "t": -3, "v": 0.14e1 },
     { "n": "current", "t": -2, "v": 1.5 },
     { "n": "current", "t": -1, "v": 1.6 },
     { "n": "current", "t": 0,   "v": 1.7 }]
}
```


- Using this:

| SenML | JSON | Type |
|---|---|---|
| Base Name | bn | String |
| Base Time | bt | Number |
| Base Units | bu | Number |
| Version | ver | Number |
| Measurement or Parameters | e | Array |

| SenML | JSON | Notes |
|---|---|---|
| Name | n | String |
| Units | u | String |
| Value | v | Floating point |
| String Value | sv | String |
| Boolean Value | bv | Boolean |
| Value Sum | s | Floating point |
| Time | t | Number |
| Update Time | ut | Number |

https://tools.ietf.org/pdf/draft-jennings-core-senml-01.pdf

# Data Formats: CBOR (Concise Binary Object Representation)

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering
InES Institute of Embedded Systems

Session 1: Data Acquisition and Reduction
- Sampling and Filtering
- Data Reduction and Analytics
- SAX

**Session 2: Data Handling**
- **Communication Costs**
- Scenarios
- Stream Analytics

Session 3: Data Storage
- SQL
- NoSQL

**Learning Aim:** The students will be able to explain the principle of a CBOR document

- Draft specification CBOR
  - https://www.rfc-editor.org/rfc/rfc8949.html
  - JSON-like (conceptually) binary encoding of data (24 bytes)

```
name: "Strawberry Pie"
data: <00 01 02 03 04 05 06 07 08 09>
```

Base64 Encoded JSON (51 bytes)

{"name":"Strawberry Pie","data":"AAECAwQFBgcICQ=="}

CBOR Encoded (35 bytes)

a2646e616d656e5374726177626572727920506965646174614a00010203040506070809

```
a2                    -- Map, 2 pairs
  64                  -- String, length: 4
    6e616d65          -- {Key:0}, "name"
  6e                  -- String, length: 14
    5374726177626572727920506965 -- {Val:0}, "Strawberry Pie"
  64                  -- String, length: 4
    64617461 -- {Key:1}, "data"
  4a                  -- Bytes, length: 10
    00010203040506070809 -- {Val:1}, 00010203040506070809
```



https://www.endpointdev.com/blog/2019/03/extensible-binary-encoding-with-cbor/

# Local Control

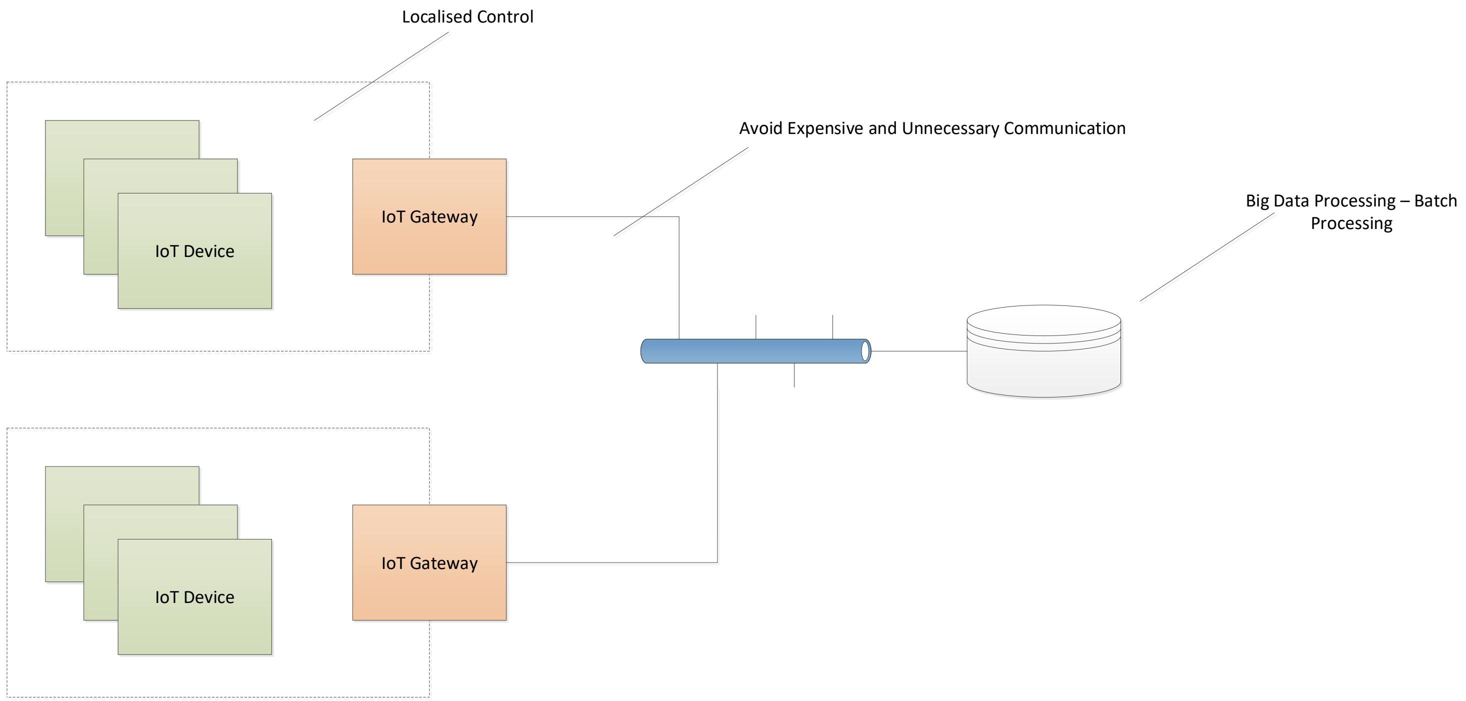


Session 1: Data Acquisition and Reduction
- Sampling and Filtering
- Data Reduction and Analytics
- SAX

**Session 2: Data Handling**
- Communication Costs
- **Scenarios**
- Stream Analytics

Session 3: Data Storage
- SQL
- NoSQL

- Story: George gets an insurance reduction if he subscribes to a water leak detection service
- Exercise: Where is this service best located?



# Local Control - Platforms



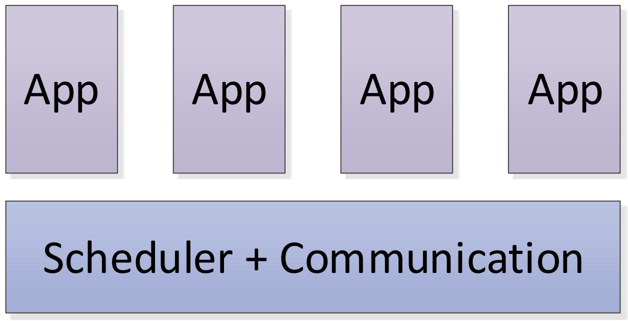
Session 1: Data Acquisition and Reduction
- Sampling and Filtering
- Data Reduction and Analytics
- SAX

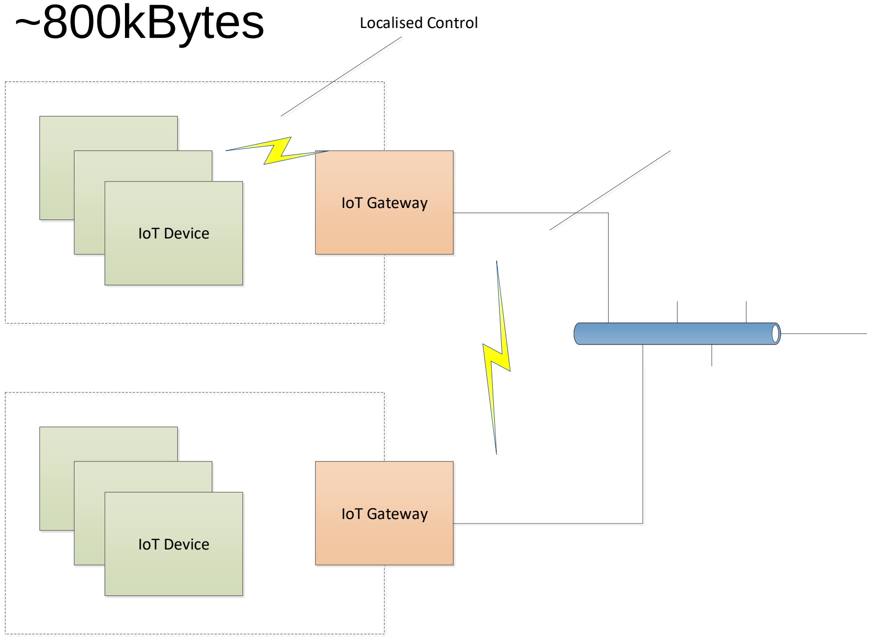
Session 2: Data Handling
- Communication Costs
- **Scenarios**
- Stream Analytics

Session 3: Data Storage
- SQL
- NoSQL

## Apache Edgent

- Microkernel architecture
- Local data analysis can also be used to reduce communication costs and backend processing costs
- Edge devices can communicate with other edge devices
- Java based – core JARS and extensions ~800kBytes




**Learning Aim:** The students will be able to explain the principle of a microkernel architecture

**Learning Aim:** The students will be able to explain why a microkernel architecture is suited to IoT application distribution and orchestration

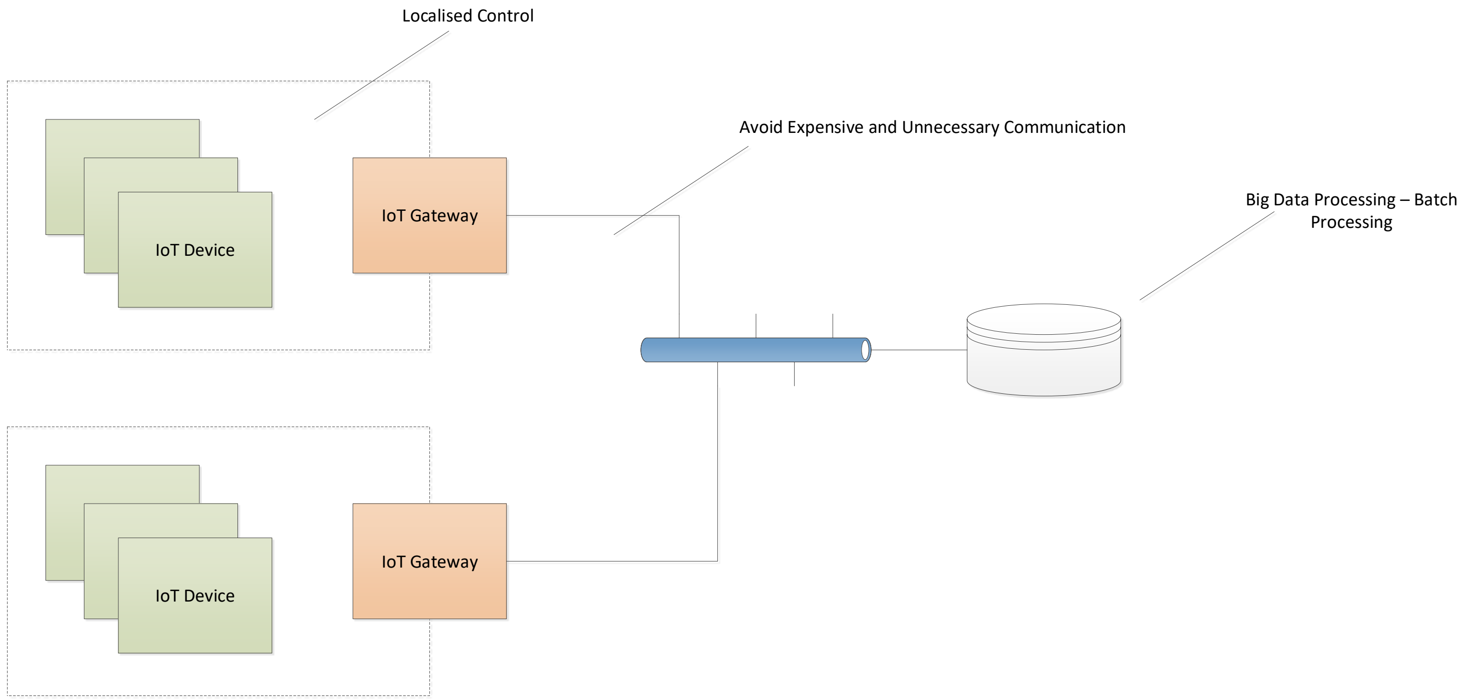
# Backend Control

- Story: George wants monthly billing with 10% overpayment
- Exercise: Where is this service best located?


Session 1: Data Acquisition and Reduction
- Sampling and Filtering
- Data Reduction and Analytics
- SAX

**Session 2: Data Handling**
- Communication Costs
- **Scenarios**
- Stream Analytics

Session 3: Data Storage
- SQL
- NoSQL

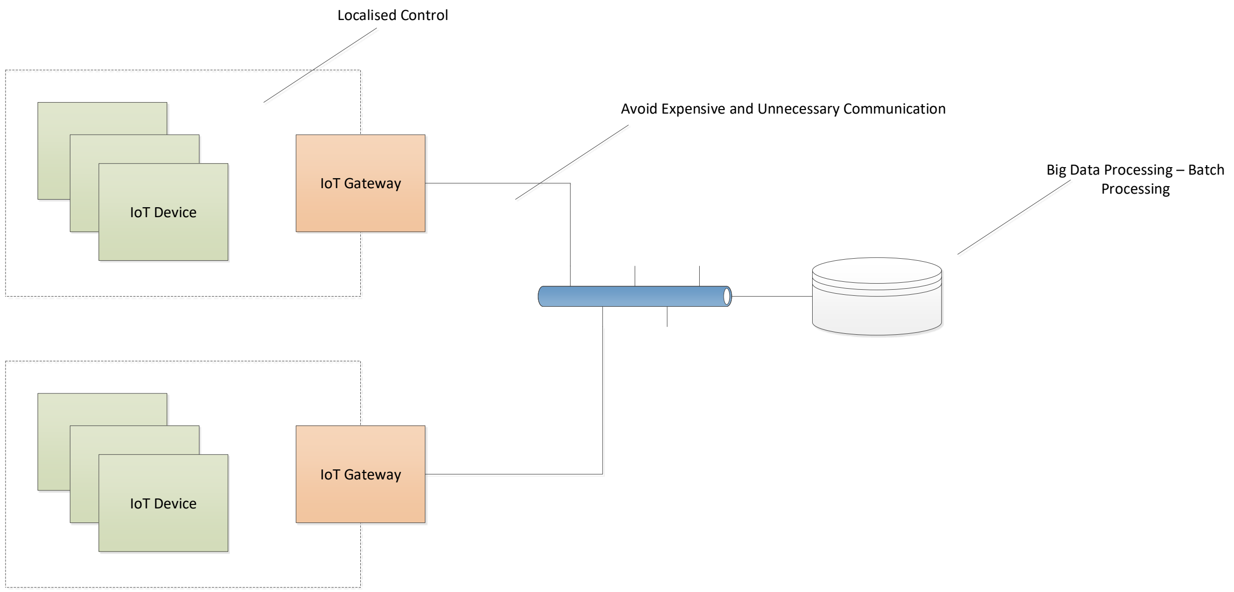
# ? Control


Session 1: Data Acquisition and Reduction
- Sampling and Filtering
- Data Reduction and Analytics
- SAX

Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

Session 3: Data Storage
- SQL
- NoSQL


- Story: Water corporation want to match supply with demand
  - Near real-time processing using analytics and prediction algorithms
- Exercise: Where is this service best located?



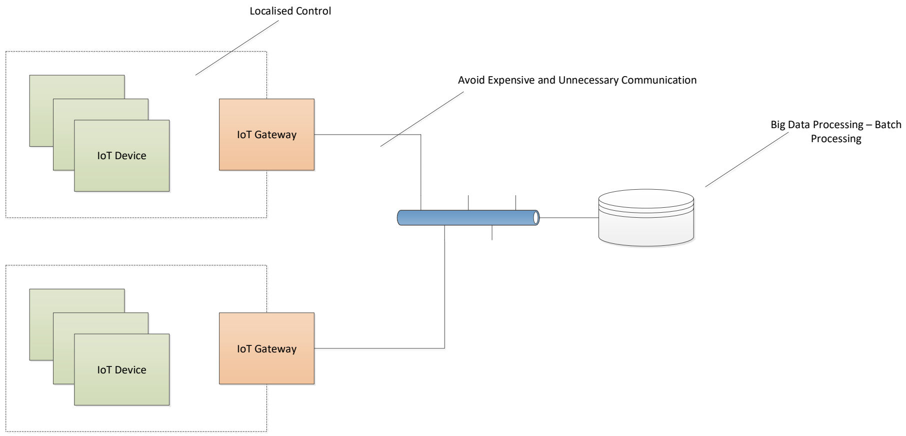
# Answer - stream analytics

Session 1: Data Acquisition and Reduction
- Sampling and Filtering
- Data Reduction and Analytics
- SAX

Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

Session 3: Data Storage
- SQL
- NoSQL

- Single pass over time series
  - Very low memory usage
- Processed continuously or in small batches
- Landmark Model
  - From current time to start of series
- Sliding Window Model
  - Window over the data
- Dampening Model
  - Assigns weights to values over time series




- Looking for: number and frequency of items, median, frequency, moments …

# Stream analytics - Locations

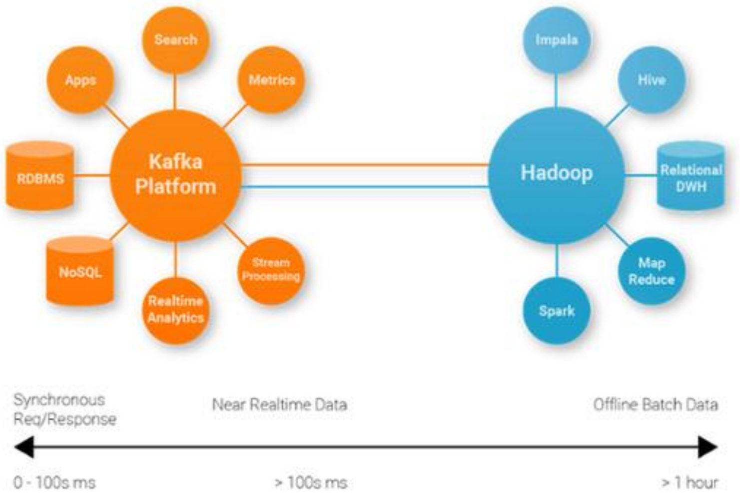


Session 1: Data Acquisition and Reduction
- Sampling and Filtering
- Data Reduction and Analytics
- SAX

Session 2: Data Handling
- Communication Costs
- Scenarios
- **Stream Analytics**

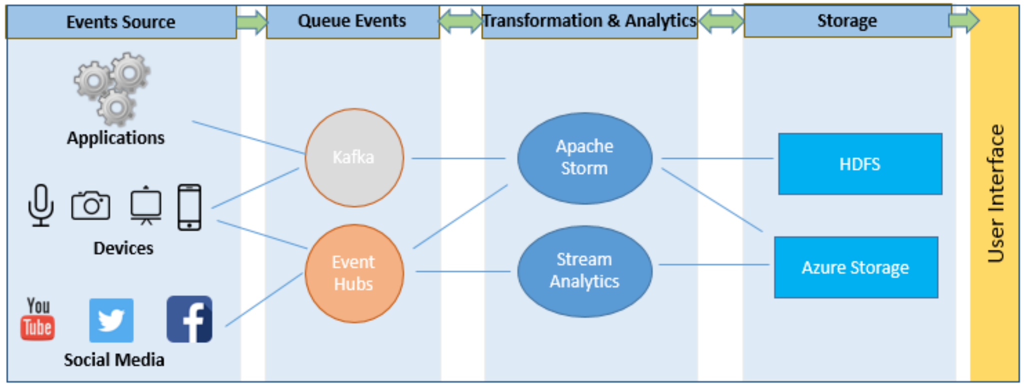
Session 3: Data Storage
- SQL
- NoSQL

- Stream analytics creates new streams
  - reduce time series
  - creates time series




- New streams need a destination

- Idea – combine stream analytics with messaging systems




https://www.computerworld.com/article/2999864/big-data/how-apache-kafka-is-greasing-the-wheels-for-big-data.html
https://www.hcltech.com/blogs/apache-storm-hdinsight

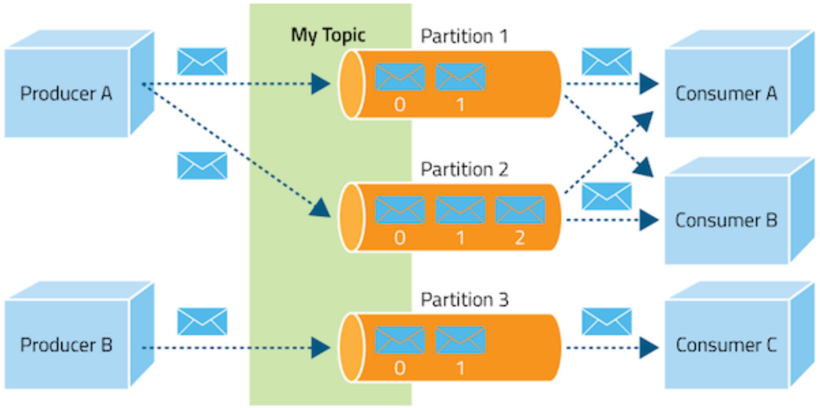
# Stream analytics - Locations


Session 1: Data Acquisition and Reduction
- Sampling and Filtering
- Data Reduction and Analytics
- SAX

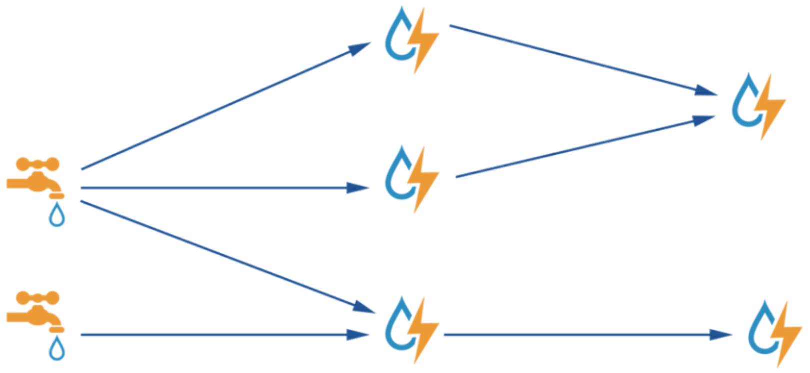
Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

Session 3: Data Storage
- SQL
- NoSQL


- f.i Apache Kafka (LinkedIn)
  - Business-logic level messaging service
  - Stores messages in queues
  - Messages can be consumed
  - Message lifetime a function of calendar time, not of consumption




- f.i. Apache Storm
  - Framework for streaming analytics
  - Attachments for databases and Kafka



https://www.cloudera.com/documentation/kafka/1-2-x/topics/kafka.html

# Session 3: Data Storage


**Learning Aim 3:** The students will be able to explain the difference between a relational and noSQL database

# Data Bases



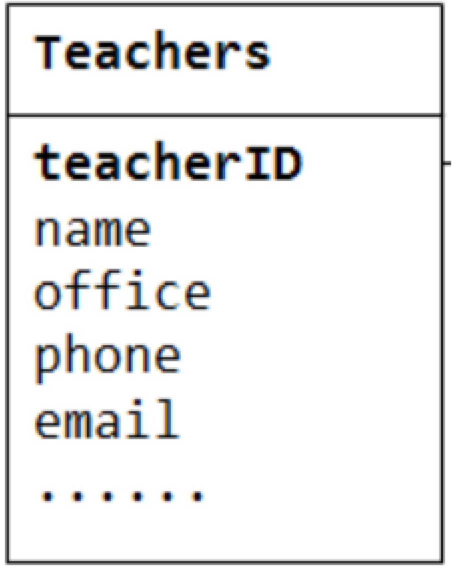
Session 1: Data Acquisition and Reduction
- Sampling and Filtering
- Data Reduction and Analytics
- SAX

Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

**Session 3: Data Storage**
- **SQL**
- NoSQL

## Relational **Data**bases

- Based around a table made up of columns of keys and rows of values
- Primary key is the searchable key
- Rows cannot be duplicated – primary key needs to be unique
- Simple key = one primary key. Composite key if made up of multiple columns
- Values can have various basic types




– The table teachers has a primary and unique key teacherID

https://www.ntu.edu.sg/home/ehchua/programming/sql/Relational_Database_Design.html

# Data Bases

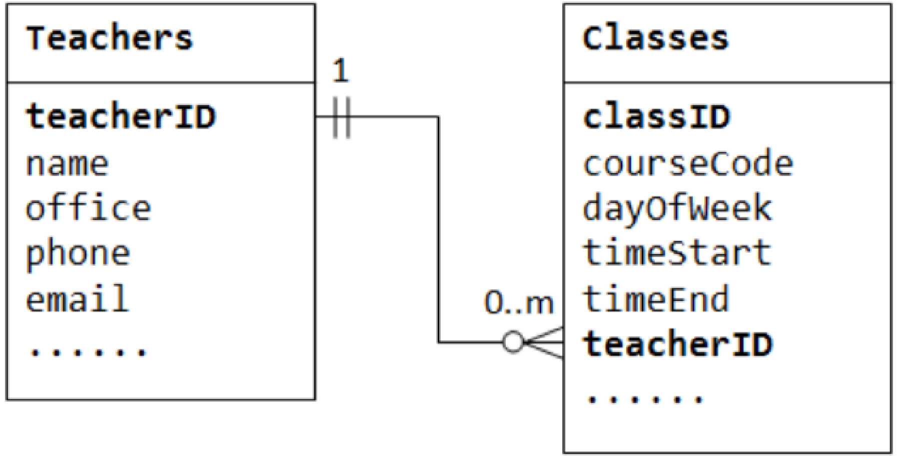
Session 1: Data Acquisition and Reduction
- Sampling and Filtering
- Data Reduction and Analytics
- SAX

Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

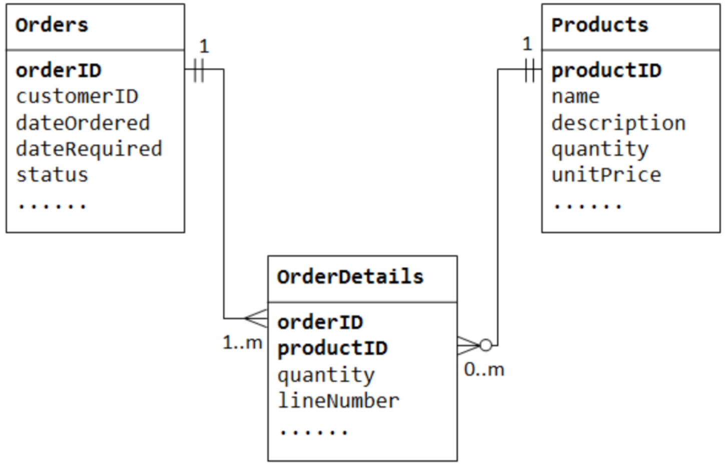
**Session 3: Data Storage**
- **SQL**
- NoSQL

## **Relational** Database

- Relationships between tables
- one to many
- teacherID in table Classes is a *foreign key*




- Many to many managed over a junction table
  - To retrieve the details of an order need to search for orderID and find the particular products over the productID




https://www.ntu.edu.sg/home/ehchua/programming/sql/Relational_Database_Design.html

# Data Bases - SQL

Session 1: Data Acquisition and Reduction
- Sampling and Filtering
- Data Reduction and Analytics
- SAX

Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

**Session 3: Data Storage**
- **SQL**
- NoSQL

## Structured Query Language

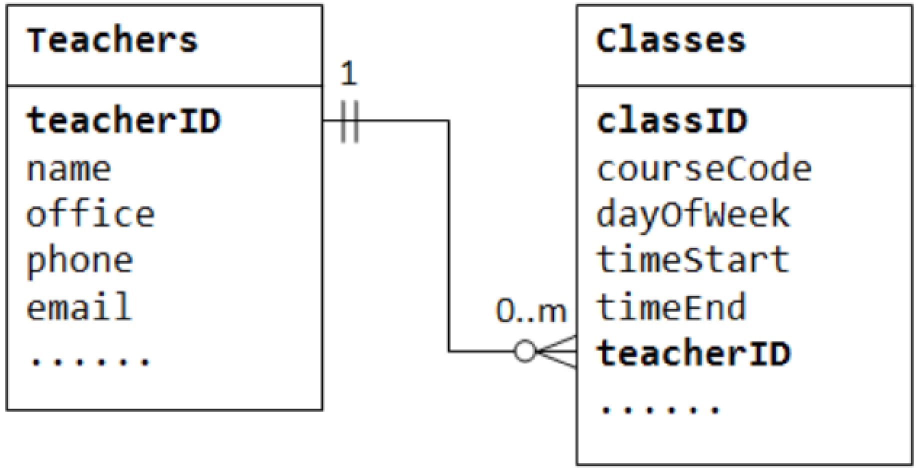
- Interface between user and database
- Not as standard as it could be – details vary from DB to DB

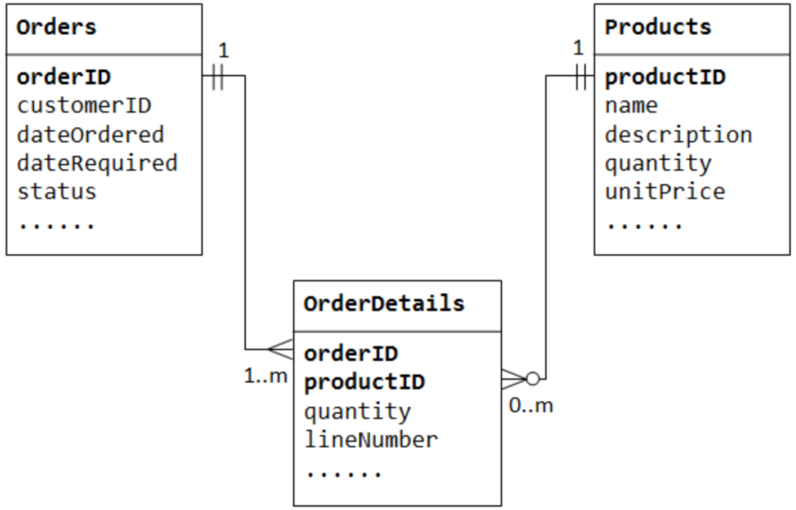
```
// Insert new data
INSERT Teachers (teacherID, name, office, email)
VALUES ('7676', 'Doran', 'TW114', 'donn@zhaw.ch')

// update data
UPDATE Teachers
SET office = 'TW108'

// read entry
SELECT name
FROM Teachers
WHERE office LIKE 'TW%'

// delete entry
DELETE Teachers
WHERE name = Doran
```




https://www.ntu.edu.sg/home/ehchua/programming/sql/Relational_Database_Design.html

# Data Bases - SQL

Session 1: Data Acquisition and Reduction
- Sampling and Filtering
- Data Reduction and Analytics
- SAX

Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

**Session 3: Data Storage**
- **SQL**
- NoSQL

## Join

- Selecting data sets

### SQL JOIN Examples

**Problem:** List all orders with customer information

| ORDER |
|---|
| Id (key) |
| OrderDate |
| OrderNumber |
| CustomerId |
| TotalAmount |

| CUSTOMER |
|---|
| Id (key) |
| FirstName |
| LastName |
| City |
| Country |
| Phone |

```
SELECT OrderNumber, TotalAmount, FirstName, LastName, City, Country
  FROM [Order] JOIN Customer
    ON [Order].CustomerId = Customer.Id
```

In this example using table aliases for [Order] and Customer might have been useful.

**Results:** 830 records.

| OrderNumber | TotalAmount | FirstName | LastName | City | Country |
|---|---|---|---|---|---|
| 542378 | 440.00 | Paul | Henriot | Reims | France |

https://www.dofactory.com/sql/join

# Databases ACID



Session 1: Data Acquisition and Reduction
- Sampling and Filtering
- Data Reduction and Analytics
- SAX

Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

**Session 3: Data Storage**
- **SQL**
- NoSQL

- Fundamental Property of Relational Databases

**INSERT** Teachers (teacherID, name, gender)
**VALUES** ('7676', 'Doran', 'neutral')

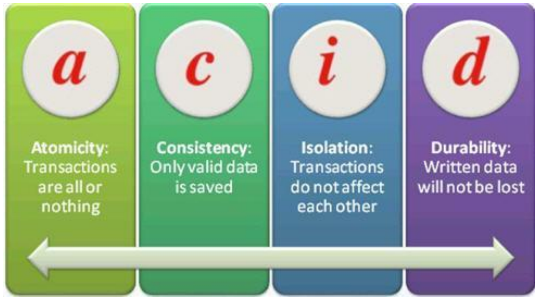
- Consistency -> if 'neutral' not allowed then **INSERT** will be rejected
- Atomicity -> the other values will not be stored either

**INSERT** Teachers (teacherID, name, gender)
**VALUES** ('7676', 'Doran', 'female')

**UPDATE** Teachers (teacherID, name, gender)
**VALUES** ('7676', 'Doran', 'male')

- Isolation -> first SQL block acquires lock and will finish before second statement set

- Durability -> data will never be lost




https://www.linkedin.com/pulse/what-acid-properties-database-aseem-jain

# Data Bases

- First and second generation databases:
  - Generally use their own operating system /file systems
  - Don’t scale well - not good for huge data quantities
  - not good for unstructured or semi-structured data
  - Often represent single point of failure
  - Tends towards slowness

http://www.allthingsdistributed.com/2009/10/amazon_relational_database_service.html


Session 1: Data Acquisition and Reduction
- Sampling and Filtering
- Data Reduction and Analytics
- SAX

Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

**Session 3: Data Storage**
- **SQL**
- NoSQL

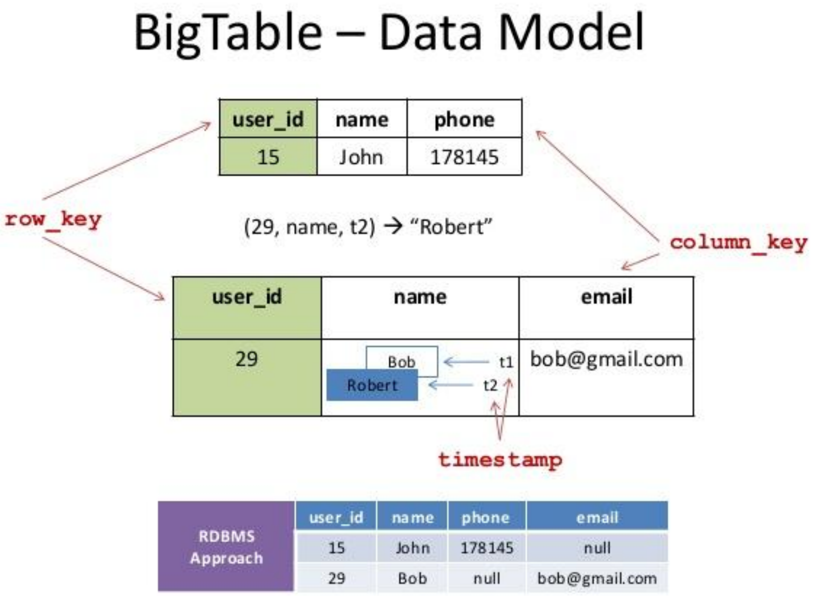
# NoSQL (1)

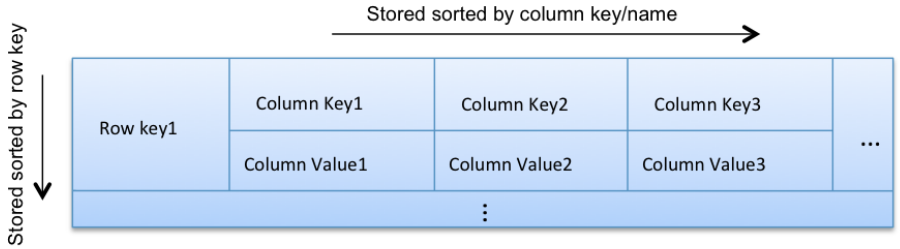
Session 1: Data Acquisition and Reduction
- Sampling and Filtering
- Data Reduction and Analytics
- SAX

Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

**Session 3: Data Storage**
- SQL
- **NoSQL**

- Consistency hard to achieve together with user response performance => NoSQL
  - Name spawned by database w/o SQL and re-used to denote non relational DBs
- Tend towards key-value pairs
  - BigTable from Google
    - Row + column+ timestamp
  - Cassandra – facebook
    - Column orientated
    - No joins, must be handled by application




https://www.slideshare.net/GrishaWeintraub/cap-28353551
http://www.ebaytechblog.com/2012/07/16/cassandra-data-modeling-best-practices-part-1/

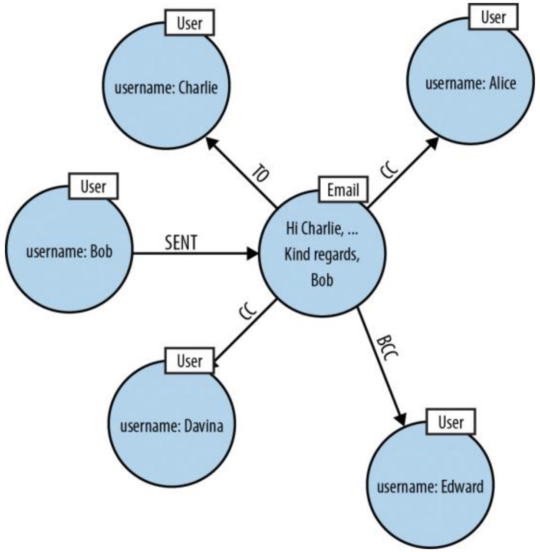
# NoSQL (2)

Session 1: Data Acquisition and Reduction
- Sampling and Filtering
- Data Reduction and Analytics
- SAX

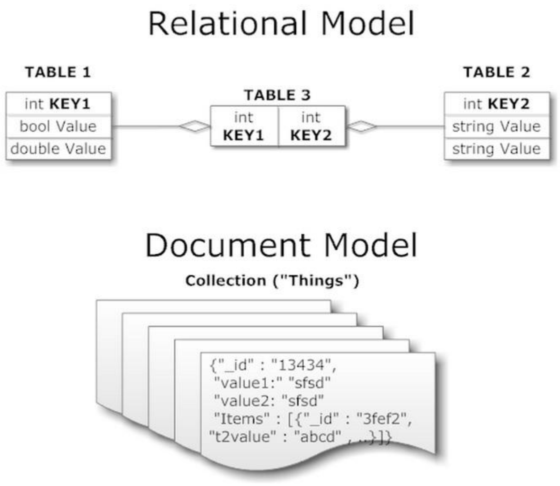
Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

**Session 3: Data Storage**
- SQL
- **NoSQL**

- GraphDB
  - Store data as a graph




- Document Database
  - Key value pairs in f.i. JSON
  - Joins must be supported in application
  - Easy to map data from and to OO programs
  - MungoDB




https://neo4j.com/blog/data-modeling-pitfalls/
https://www.morpheusdata.com/blog/2015-03-06-when-one-data-model-just-won-t-do-database-design-that-supports-polyglot-persistence

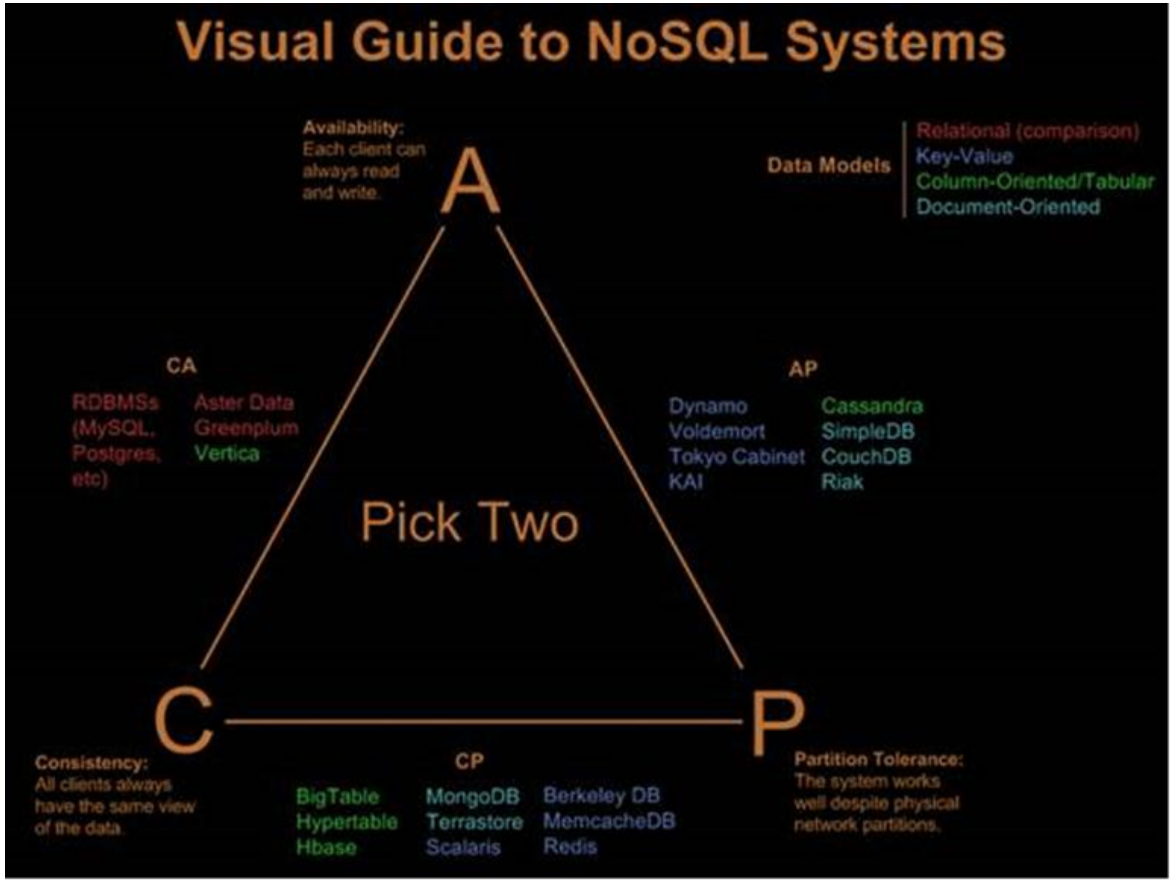
# Huge Data

Session 1: Data Acquisition and Reduction
- Sampling and Filtering
- Data Reduction and Analytics
- SAX

Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

**Session 3: Data Storage**
- SQL
- **NoSQL**

- NoSQL tend towards distributed databases
  - Problem -> distributed, user experience and consistency are uneasy bedfellows
  - CAP theory






https://www.linkedin.com/pulse/what-acid-properties-database-aseem-jain
http://blog.nahurst.com/visual-guide-to-nosql-systems

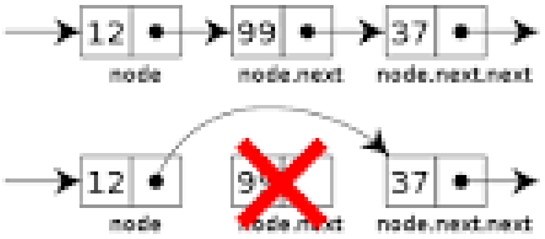
# Practical Issues (1)

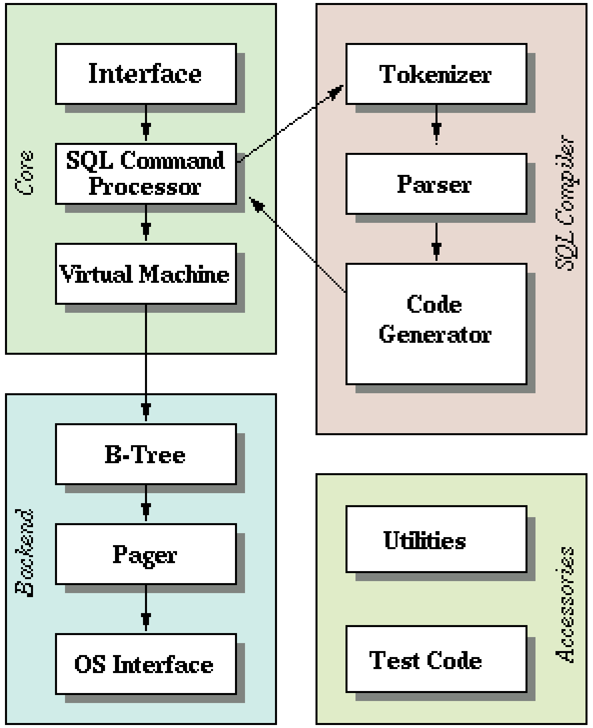
Session 1: Data Acquisition and Reduction
- Sampling and Filtering
- Data Reduction and Analytics
- SAX

Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

**Session 3: Data Storage**
- SQL
- **NoSQL**

- Databases required on embedded devices
  - Generally constrained implementations implemented using linked lists or some other such dynamic storage form
  - SQlite uses files (therefore file system -> OS)
    - Min size ~300k, linux version ca 800k
  - In large DB-s in-memory is often used to increase responsivity  (file system/disk accesses expensive




https://en.wikipedia.org/wiki/Linked_list
https://www.sqlite.org/arch.html

# Practical Issues (2)

- Databases required on embedded devices
  - Cassandra functions on raspberryPi
    - In our experience not well

- Seems to be a need for a solution that combines small resource use, inter device data persistence and cloud-sized DB connectivity for constrained applications.

Session 1: Data Acquisition and Reduction
- Sampling and Filtering
- Data Reduction and Analytics
- SAX

Session 2: Data Handling
- Communication Costs
- Scenarios
- Stream Analytics

**Session 3: Data Storage**
- SQL
- **NoSQL**