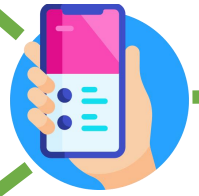
DSA5104
Principles of Data Management and Retrieval

Lecture 0: Overview

Data Is Ubiquitous

Social Media

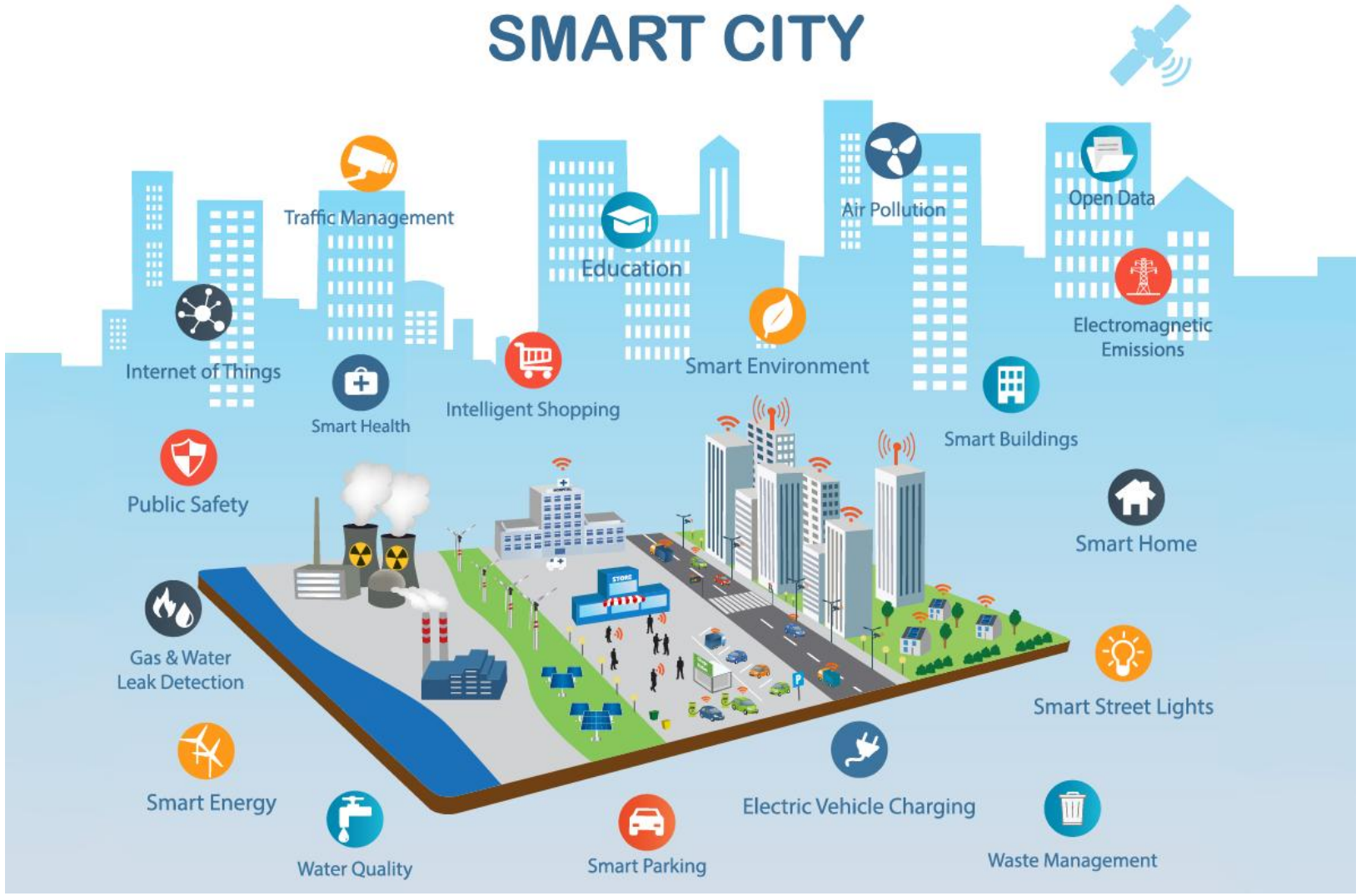


E-Commerce

Internet of Things (IoT)

https://www.nist.gov/blogs/cybersecurity-insights/more-just-milestone-botnet-roadmap-towards-more-securable-iot-devices

Big Data - Data Flooding in Smart Cities

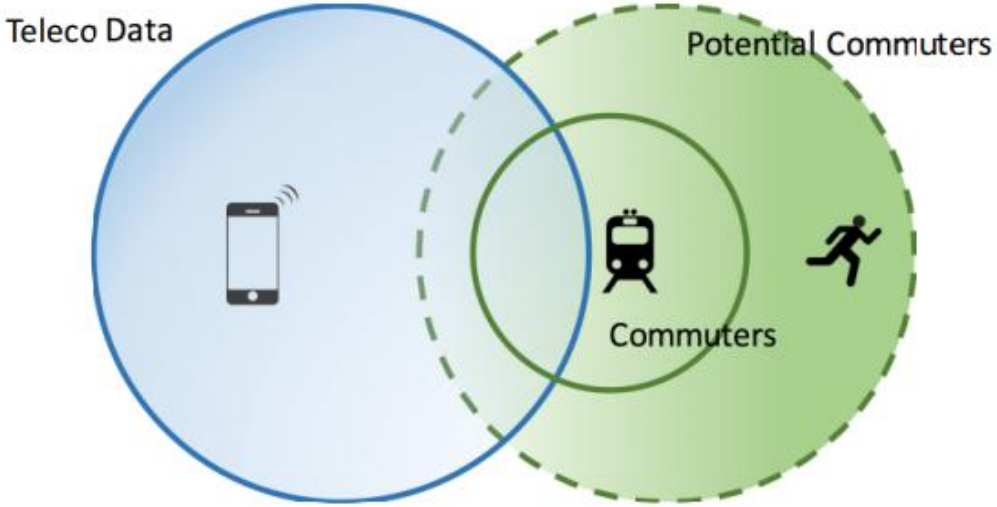


https://adtellintegration.com/smart-cities-infrastructure/

Teleco + Metro Data → Crowd Forcasting

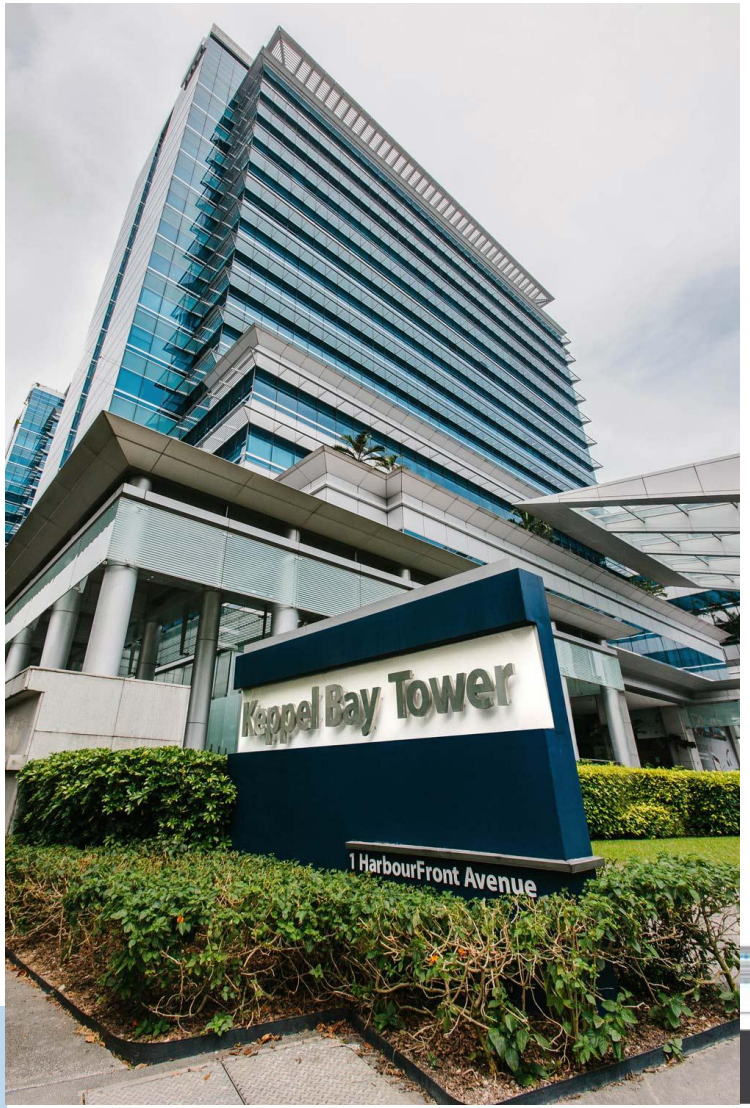
User ID	Time	Longitude	Latitude	Action
1001	8:00:00am	103.737459	1.326909	SMS
1002	8:00:01am	103.737512	1.327108	CAL
1003	8:00:03am	103.741002	1.339921	SMS
1004	8:00:03am	103.738199	1.331231	INT

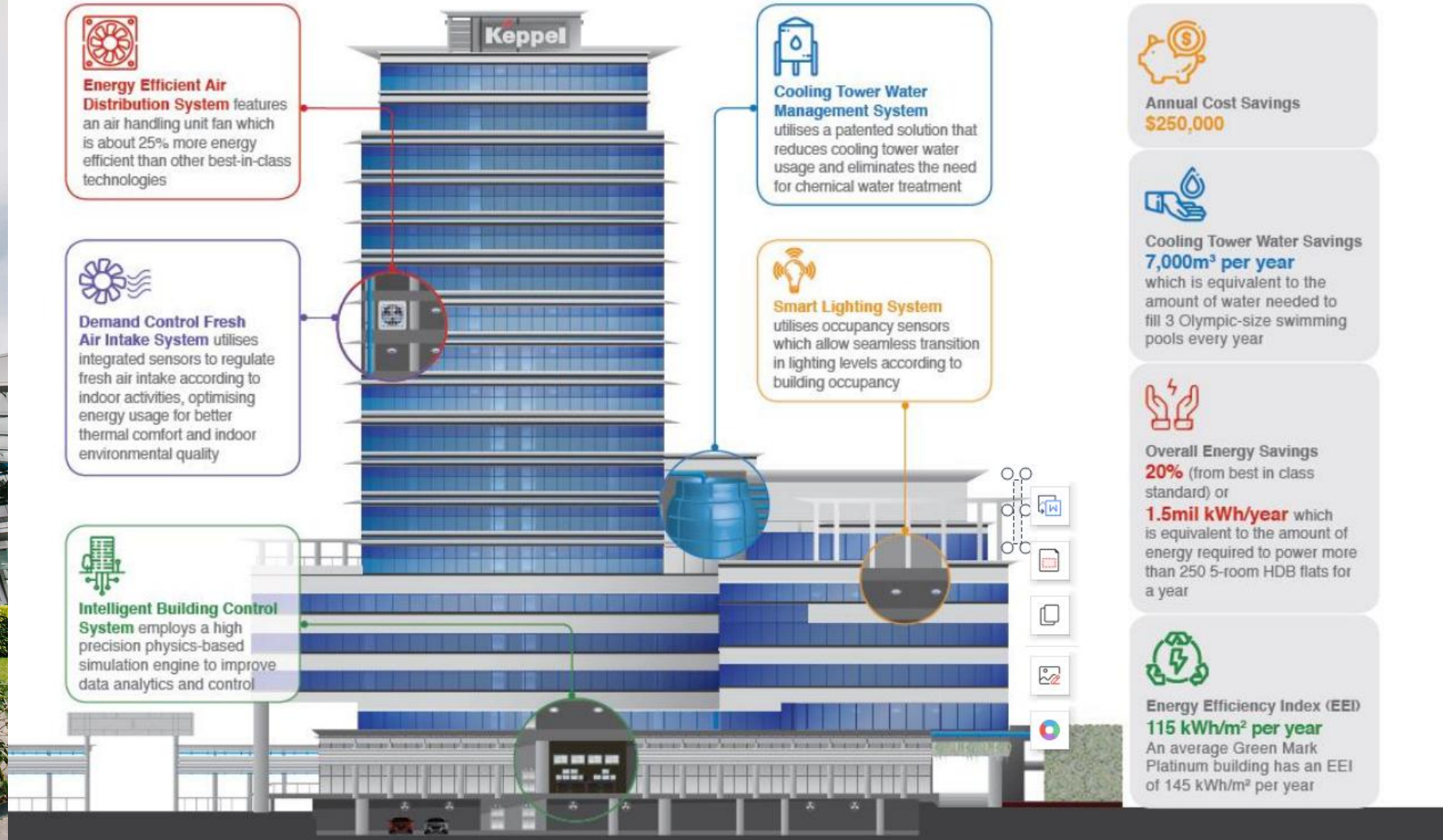
Caller Detail Records (CDR) [1]



[1] Liang, Victor C., et al. "Mercury: Metro density prediction with recurrent neural network on streaming CDR data." ICDE 2016.

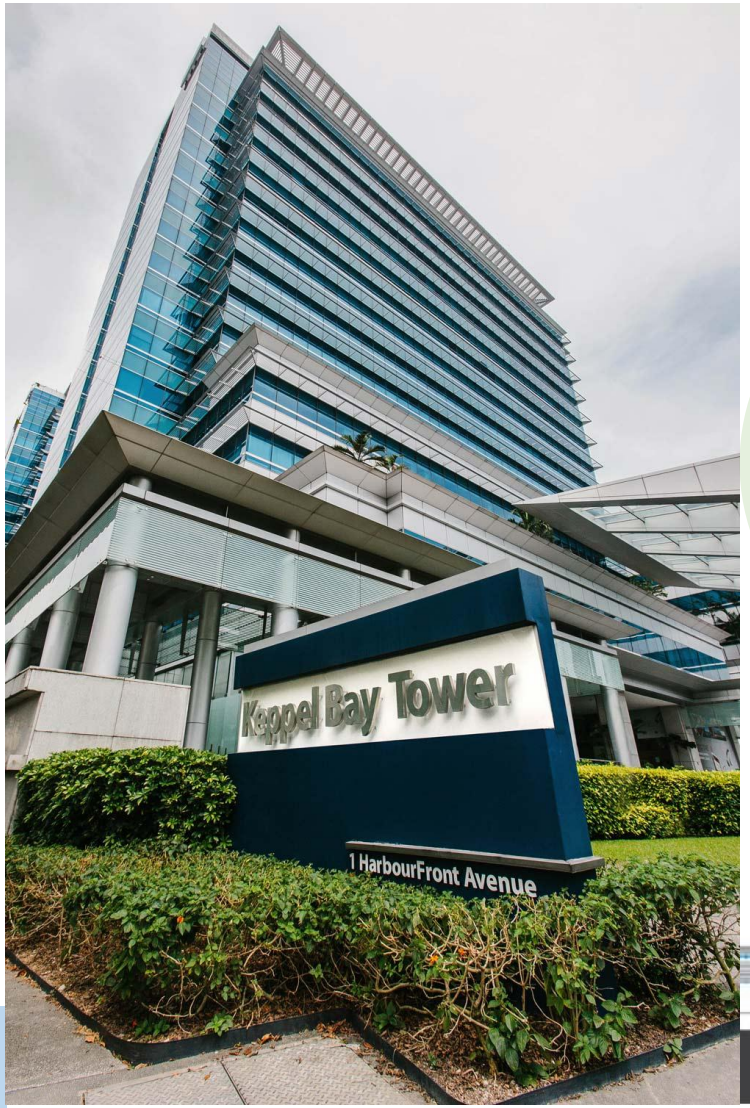
Keppel Bay Tower - Singapore's first Green Mark Platinum (Zero Energy) commercial building

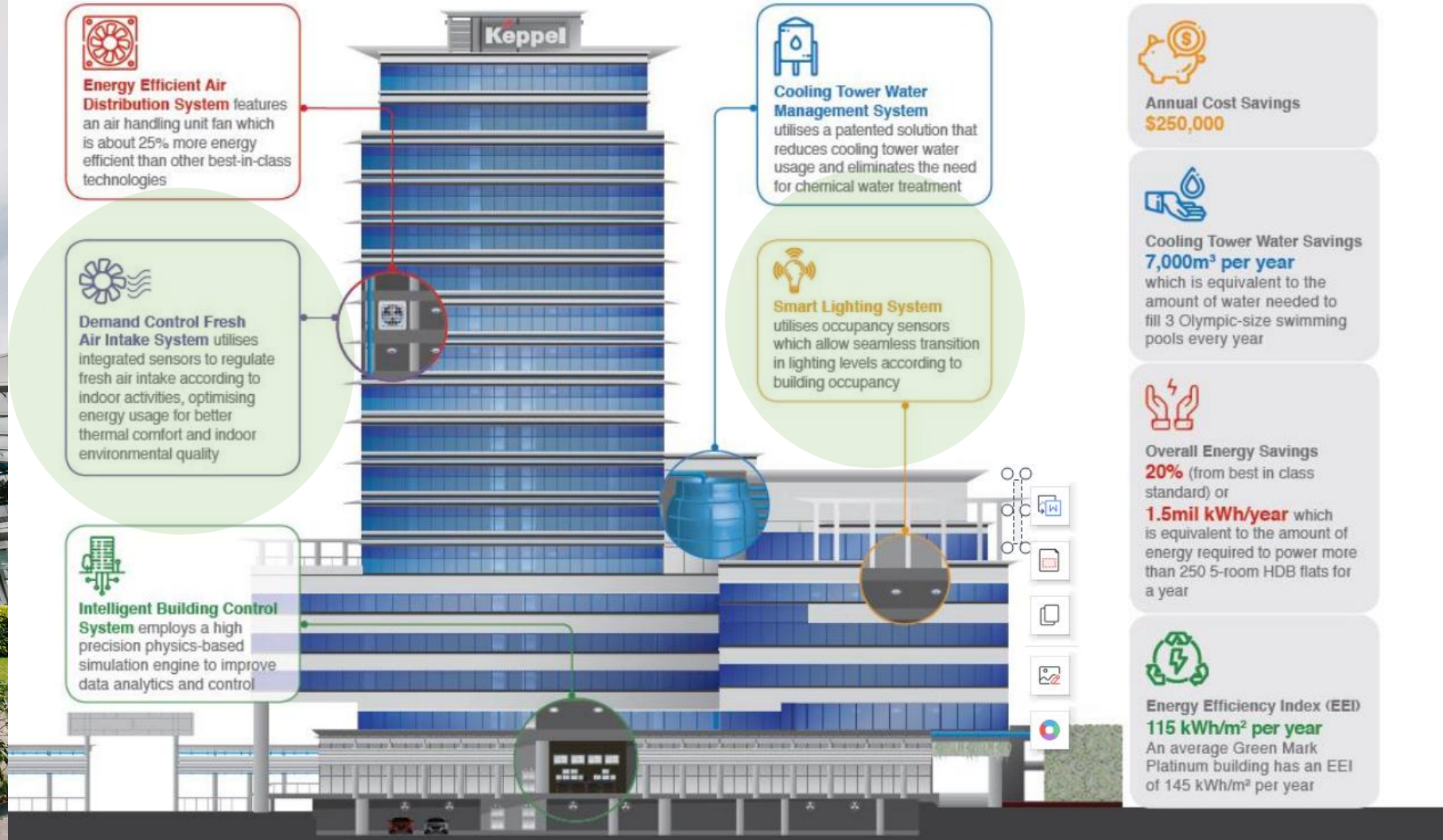




https://sleb.sg/UserFiles/Resource/GBIC/GBIC%20Demo%20Project/KBT%20GBIC%20Dashboard.pdf

Keppel Bay Tower - Singapore's first Green Mark Platinum (Zero Energy) commercial building





https://sleb.sg/UserFiles/Resource/GBIC/GBIC%20Demo%20Project/KBT%20GBIC%20Dashboard.pdf

Data Categorized by Data Types

2021 in pictures | Galle...
cnn.com

Pictures of the month: April | Reuter...
reuters.com

350+ Free Pictures [HD] | Download Free...
unsplash.com

Photographs vs pictures: Whatever you ...
recordnet.com

Download · Pexels Stock Photos
pexels.com

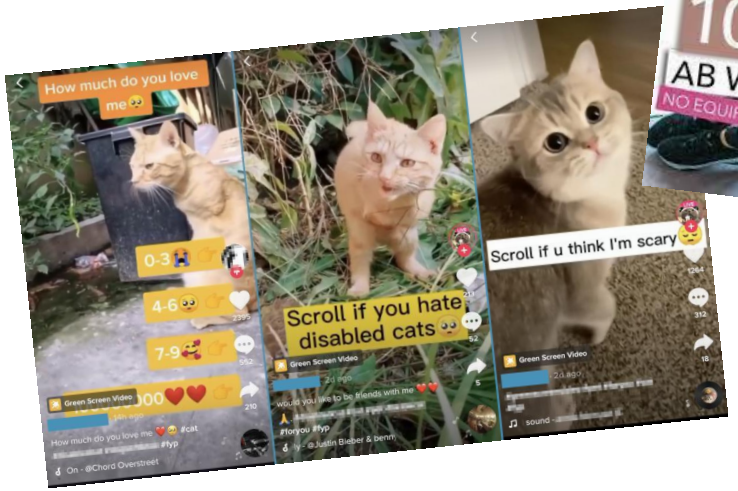
How to Take Good Pictures (10...
expertphotography.com

Pictures - Home | Facebook
m.facebook.com

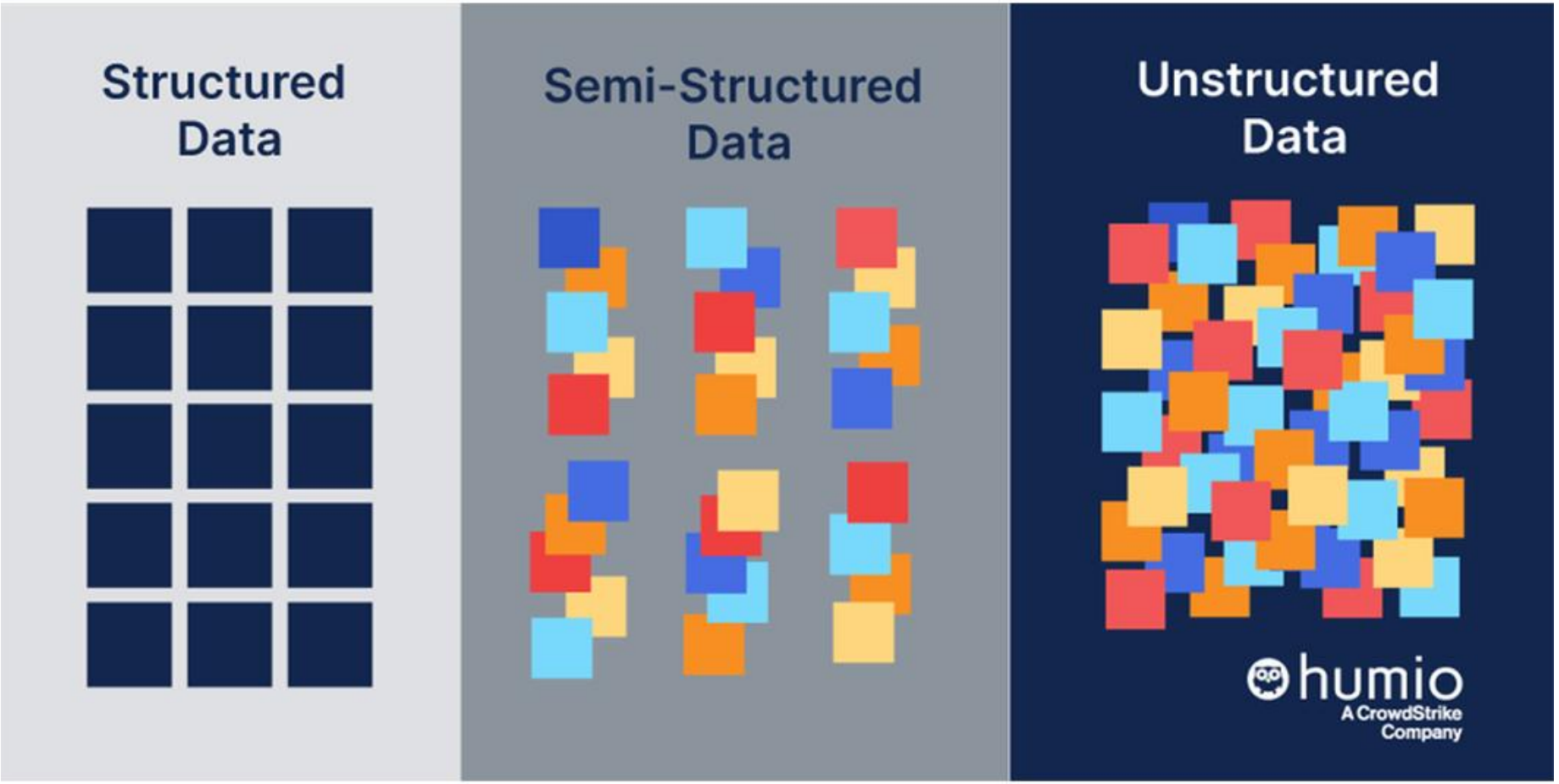
Pictures of the month: January ...
reuters.com

2021: The year in pictures | The ...
straitstimes.com

How to Take Sharp Photos
photographylife.com



Data Categorized by Data Types



https://www.crowdstrike.com/blog/structured-unstructured-and-semi-structured-logging-explained/

Structured Data

- Data conforms to a set schema
  - Numerical, categorical and text data with well-defined schema

Table name: *instructor*

Column	Data Type
ID	varchar(5)
name	varchar(20) not null
dept_name	varchar(20)
salary	numberic(8, 2)

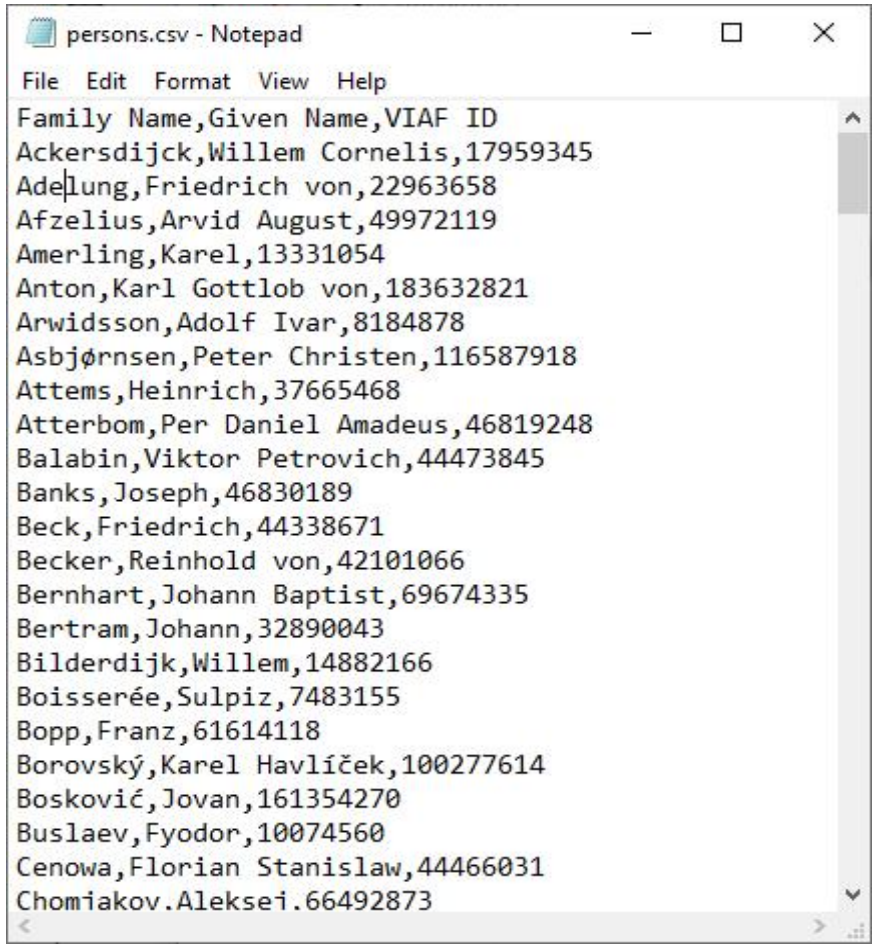
instructor

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

Semi-Structured Data

- Data with labels but no fixed schema
  - JSON, XML
  - CSV files with headers

- Usage:
  - For data exchange



https://e.nodegoat.net/CMS/upload/guide-import_person_csv_notepad.png

Semi-Structured Data

- XML - eXtensible Markup Language

```
<note>
  <to>Daddy</to>
  <from>Jimmy</from>
  <body>Don't forget to play with me this weekend!</body>
</note>
```

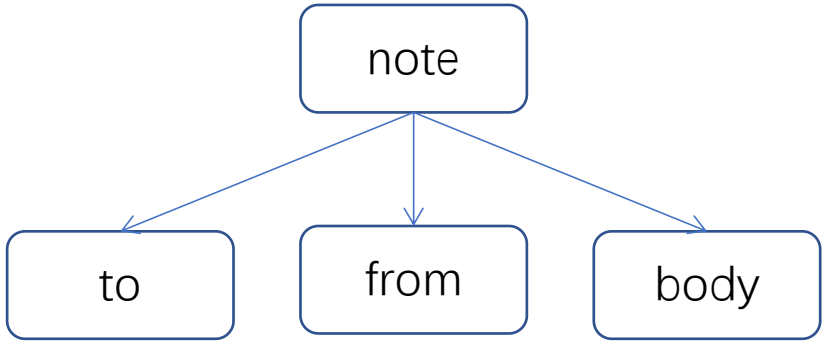
Semi-Structured Data

- XML - eXtensible Markup Language

Tag

```
<note>
  <to>Daddy</to>
  <from>Jimmy</from>
  <body>Don't forget to play with me this weekend!</body>
</note>
```

Closing tag



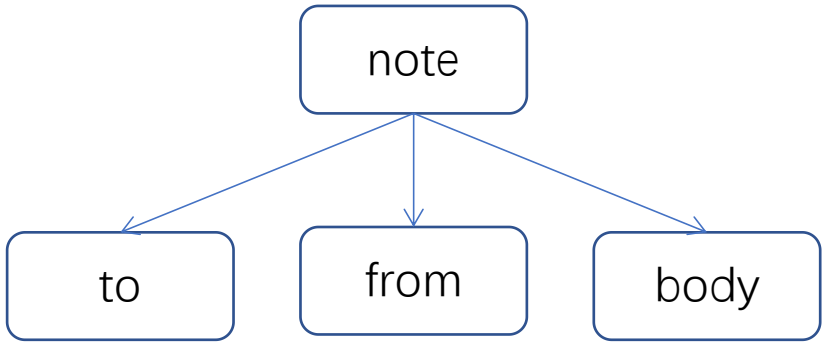
Semi-Structured Data

- XML - eXtensible Markup Language

Tag

```
<note>
  <to>Daddy</to>
  <from>Jimmy</from>
  <body>Don't forget to play with me this weekend!</body>
</note>
```

Closing tag



Note

To: Daddy

From: Jimmy

Don't forget to play with me this weekend!

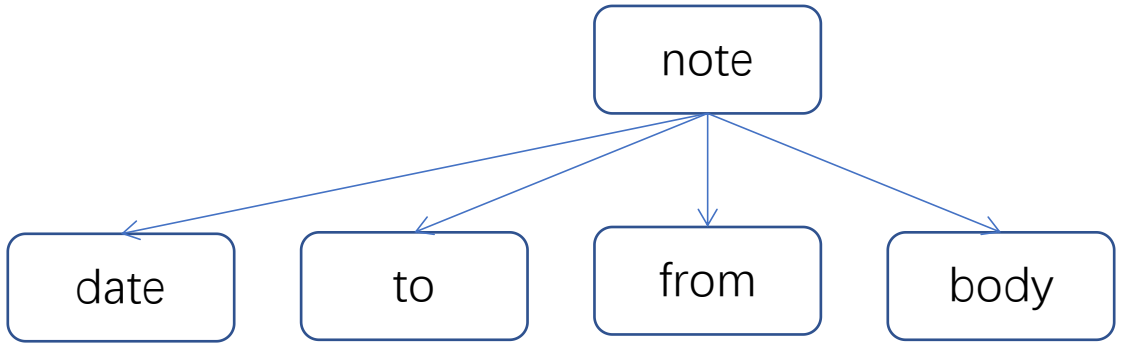
Semi-Structured Data

- XML - eXtensible Markup Language

Tag

```
1  <note>
2    <to>Daddy</to>
3    <from>Jimmy</from>
4    <body>Don't forget to play with me this weekend!</body>
5  </note>
```

Closing tag



Note

To: Daddy

From: Jimmy

Don't forget to play with me this weekend!

Semi-Structured Data

- XML

```
<?xml version="1.0" encoding="ISO-8859-15"?>
<package destination="SU" origin="ASR" version="1.0">
  <recognized_sentence>
    <information>
      I would like the train fares from Valencia to Madrid
    </information>
    <confidences>
        <word confidence="0.47" value="I" />
        <word confidence="0.68" value="would" />
        <word confidence="0.53" value="like" />
        <word confidence="0.75" value="the" />
        <word confidence="0.64" value="train" />
        <word confidence="0.56" value="fares" />
        <word confidence="0.84" value="from" />
        <word confidence="0.93" value="Valencia" />
        <word confidence="0.78" value="to" />
        <word confidence="0.93" value="Madrid" />
    </confidences>
  </recognized_sentence>
  <grammar name="dihana.jsgf">
</package>
```

http://www.ling.helsinki.fi/~kjokinen/ICSLP06-DoD/Programme/GriolDavid06.pdf

Semi-Structured Data

- XML

A prolog defines the XML version and the character encoding

```
<?xml version="1.0" encoding="ISO-8859-15"?>
<package destination="SU" origin="ASR" version="1.0">
  <recognized_sentence>
    <information>
      I would like the train fares from Valencia to Madrid
    </information>
    <confidences>
        <word confidence="0.47" value="I" />
        <word confidence="0.68" value="would" />
        <word confidence="0.53" value="like" />
        <word confidence="0.75" value="the" />
        <word confidence="0.64" value="train" />
        <word confidence="0.56" value="fares" />
        <word confidence="0.84" value="from" />
        <word confidence="0.93" value="Valencia" />
        <word confidence="0.78" value="to" />
        <word confidence="0.93" value="Madrid" />
    </confidences>
  </recognized_sentence>
  <grammar name="dihana.jsgf">
</package>
```

http://www.ling.helsinki.fi/~kjokinen/ICSLP06-DoD/Programme/GriolDavid06.pdf

Semi-Structured Data

- XML

A prolog defines the XML version and the character encoding

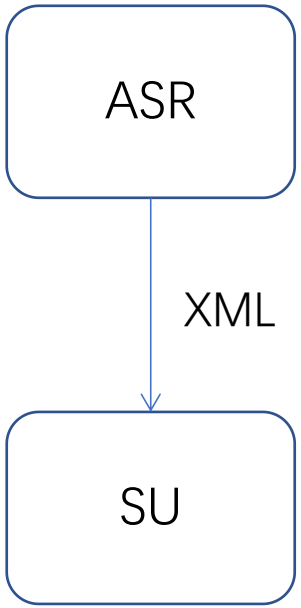
```
<?xml version="1.0" encoding="ISO-8859-15"?>
<package destination="SU" origin="ASR" version="1.0">
  <recognized_sentence>
    <information>
      I would like the train fares from Valencia to Madrid
    </information>
    <confidences>
        <word confidence="0.47" value="I" />
        <word confidence="0.68" value="would" />
        <word confidence="0.53" value="like" />
        <word confidence="0.75" value="the" />
        <word confidence="0.64" value="train" />
        <word confidence="0.56" value="fares" />
        <word confidence="0.84" value="from" />
        <word confidence="0.93" value="Valencia" />
        <word confidence="0.78" value="to" />
        <word confidence="0.93" value="Madrid" />
    </confidences>
  </recognized_sentence>
  <grammar name="dihana.jsgf">
</package>
```

Element *package* has three attributes: destination, origin and version.

http://www.ling.helsinki.fi/~kjokinen/ICSLP06-DoD/Programme/GriolDavid06.pdf

Semi-Structured Data

- XML



A prolog defines the XML version and the character encoding

```
<?xml version="1.0" encoding="ISO-8859-15"?>
<package destination="SU" origin="ASR" version="1.0">
  <recognized_sentence>
    <information>
      I would like the train fares from Valencia to Madrid
    </information>
    <confidences>
        <word confidence="0.47" value="I" />
        <word confidence="0.68" value="would" />
        <word confidence="0.53" value="like" />
        <word confidence="0.75" value="the" />
        <word confidence="0.64" value="train" />
        <word confidence="0.56" value="fares" />
        <word confidence="0.84" value="from" />
        <word confidence="0.93" value="Valencia" />
        <word confidence="0.78" value="to" />
        <word confidence="0.93" value="Madrid" />
    </confidences>
  </recognized_sentence>
  <grammar name="dihana.jsgf">
</package>
```

Element *package* has three attributes: destination, origin and version.

http://www.ling.helsinki.fi/~kjokinen/ICSLP06-DoD/Programme/GriolDavid06.pdf

Semi-Structured Data

- JSON

```
{"menu": {
  "id": "file",
  "value": "File",
  "popup": {
    "menuitem": [
      {"value": "New", "onclick": "CreateNewDoc()"},
      {"value": "Open", "onclick": "OpenDoc()"},
      {"value": "Close", "onclick": "CloseDoc()"}
    ]
  }
}}
```

https://json.org/example.html

Semi-Structured Data

- JSON

key/value pair

```
{"menu": {
  "id": "file",
  "value": "File",
  "popup": {
    "menuitem": [
      {"value": "New", "onclick": "CreateNewDoc()"},
      {"value": "Open", "onclick": "OpenDoc()"},
      {"value": Close", "onclick": "CloseDoc()"}
    ]
  }
}}
```

https://json.org/example.html

Semi-Structured Data

- JSON

key/value pair

seprated by commas

```
{"menu": {
  "id": "file",
  "value": "File",
  "popup": {
    "menuitem": [
      {"value": "New", "onclick": "CreateNewDoc()"},
      {"value": "Open", "onclick": "OpenDoc()"},
      {"value": "Close", "onclick": "CloseDoc()"}
    ]
  }
}}
```

https://json.org/example.html

Semi-Structured Data

- JSON

key/value pair

```
{"menu": {
  "id": "file",
  "value": "File",
  "popup": {
    "menuitem": [
      {"value": "New", "onclick": "CreateNewDoc()"},
      {"value": "Open", "onclick": "OpenDoc()"},
      {"value": "Close", "onclick": "CloseDoc()"}
    ]
  }
}}
```

seprated by commas

Curly brackets hold objects

https://json.org/example.html

Semi-Structured Data

- JSON

key/value pair

```
{"menu": {
  "id": "file",
  "value": "File",
  "popup": {
    "menuitem": [
      {"value": "New", "onclick": "CreateNewDoc()"},
      {"value": "Open", "onclick": "OpenDoc()"},
      {"value": "Close", "onclick": "CloseDoc()"}
    ]
  }
}}
```

seprated by commas

Square brackets hold arrays

Curly brackets hold objects

https://json.org/example.html

Semi-Structured Data

- JSON

```
{"menu": {
  "id": "file",
  "value": "File",
  "popup": {
    "menuitem": [
      {"value": "New", "onclick": "CreateNewDoc()"},
      {"value": "Open", "onclick": "OpenDoc()"},
      {"value": "Close", "onclick": "CloseDoc()"}
    ]
  }
}}
```

- XML

```
<menu id="file" value="File">
  <popup>
    <menuitem value="New" onclick="CreateNewDoc()" />
    <menuitem value="Open" onclick="OpenDoc()" />
    <menuitem value="Close" onclick="CloseDoc()" />
  </popup>
</menu>
```

https://json.org/example.html

Load JSON File Using Pandas

data.json

```
1  {
2    "Duration":{
3      "0":60,
4      "1":60,
5      "2":60,
6      "3":45,
7      "4":45,
8      "5":60
9    },
10   "Pulse":{
11     "0":110,
12     "1":117,
13     "2":103,
14     "3":109,
15     "4":117,
16     "5":102
17   },
18   "Maxpulse":{
19     "0":130,
20     "1":145,
21     "2":135,
22     "3":175,
23     "4":148,
24     "5":127
25   },
26   "Calories":{
27     "0":409,
28     "1":479,
29     "2":340,
30     "3":282,
31     "4":406,
32     "5":300
33   }
34 }
```

https://www.w3schools.com/python/pandas/pandas_csv.asp

Load JSON File Using Pandas

data.json

```
1  {
2    "Duration":{
3      "0":60,
4      "1":60,
5      "2":60,
6      "3":45,
7      "4":45,
8      "5":60
9    },
10   "Pulse":{
11     "0":110,
12     "1":117,
13     "2":103,
14     "3":109,
15     "4":117,
16     "5":102
17   },
18   "Maxpulse":{
19     "0":130,
20     "1":145,
21     "2":135,
22     "3":175,
23     "4":148,
24     "5":127
25   },
26   "Calories":{
27     "0":409,
28     "1":479,
29     "2":340,
30     "3":282,
31     "4":406,
32     "5":300
33   }
34 }
```

https://www.w3schools.com/python/pandas/pandas_csv.asp

Load JSON File Using Pandas

data.json

```
1  {
2    "Duration":{
3      "0":60,
4      "1":60,
5      "2":60,
6      "3":45,
7      "4":45,
8      "5":60
9    },
10   "Pulse":{
11     "0":110,
12     "1":117,
13     "2":103,
14     "3":109,
15     "4":117,
16     "5":102
17   },
18   "Maxpulse":{
19     "0":130,
20     "1":145,
21     "2":135,
22     "3":175,
23     "4":148,
24     "5":127
25   },
26   "Calories":{
27     "0":409,
28     "1":479,
29     "2":340,
30     "3":282,
31     "4":406,
32     "5":300
33   }
34 }
```

Jupyter Notebook

In [1]:
```
import pandas as pd

df = pd.read_json('data.json')
```

In [2]:
```
df
```

Out[2]:

	Duration	Pulse	Maxpulse	Calories
0	60	110	130	409
1	60	117	145	479
2	60	103	135	340
3	45	109	175	282
4	45	117	148	406
5	60	102	127	300

Unstructured Data

"I'm Hilda. I was born in 1990."

"My name is Max. I'm turning 20 this year."

Unstructured Data - Natural Language Text

Discharge Summary

Patient Name: *Russell Johnson*
Medical Record Number: *123456789*
Admission Date: *08/01/14*
Discharge Date: *08/05/14*
Attending Physician: *Dr Gary Marshall*
Dictated by: *Dr Gary Marhsall*

Primary Care Physician: *Dr Dianna Miller*
Referring Physician:
Consulting Physician(s): *Dr Gary Marshall - hospitalist*
Condition on Discharge: *stable*

Final Diagnosis: *RLL pneumonia, COPD exacerbation, mild CHF, osteoarthritis*

Procedures: *none*

History of Present Illness *72 year old thin white male presented to emergency on 8/1/14 with shortness of breath, weakness and dehydration. Chest X-ray showed right lower lobe infiltrate, ABGs unremarkable. Pulse ox on RA was 79%.*

1) *Pneumonia: treated with ceftriaxone and azithromycin iv. Switched to PO after 72 hours.*
2) *Exacerbation of COPD: patient treated with inhaled and oral steroids, O2 at 2l/nc. On RA at time of discharge*
3) *Weakness and dehydration: secondary to pneumonia and COPD. Responded well to strengthening with PT and regular meals.*

Discharge Medications *Zithromycin daily until gone, inhalers #of puffs,*

Discharge Instructions: *no activity restriction, regular diet, follow up in two to three weeks with regular physician.*

Unstructured Data - Natural Language Text

Discharge Summary

Patient Name: *Russell Johnson*
Medical Record Number: *123456789*
Admission Date: *08/01/14*
Discharge Date: *08/05/14*
Attending Physician: *Dr Gary Marshall*
Dictated by: *Dr Gary Marhsall*

Primary Care Physician: *Dr Dianna Miller*
Referring Physician:
Consulting Physician(s): *Dr Gary Marshall - hospitalist*
Condition on Discharge: *stable*

Final Diagnosis: *RLL pneumonia, COPD exacerbation, mild CHF, osteoarthritis*

Procedures: *none*

History of Present Illness *72 year old thin white male presented to emergency on 8/1/14 with shortness of breath, weakness and dehydration. Chest X-ray showed right lower lobe infiltrate, ABGs unremarkable. Pulse ox on RA was 79%.*

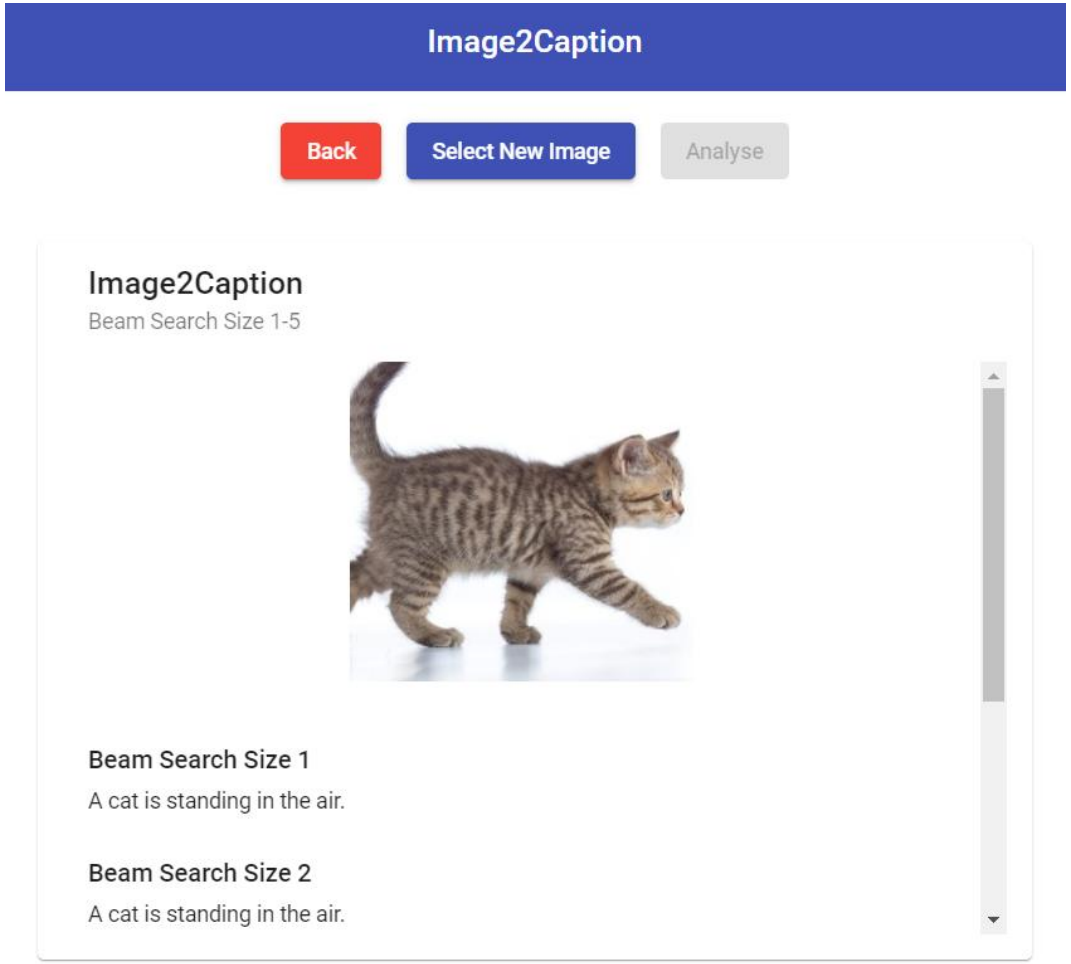
1) *Pneumonia: treated with ceftriaxone and azithromycin iv. Switched to PO after 72 hours.*
2) *Exacerbation of COPD: patient treated with inhaled and oral steroids, O2 at 2l/nc. On RA at time of discharge*
3) *Weakness and dehydration: secondary to pneumonia and COPD. Responded well to strengthening with PT and regular meals.*

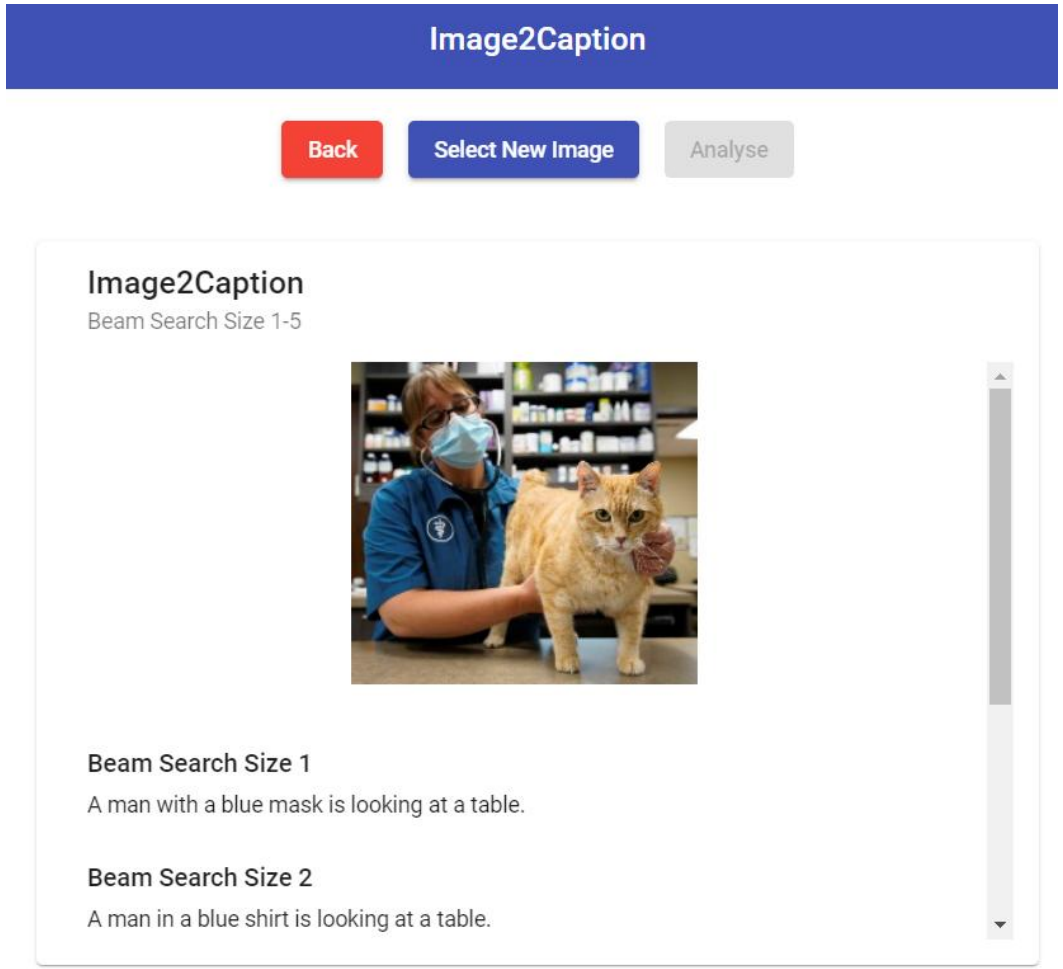
Discharge Medications *Zithromycin daily until gone, inhalers #of puffs,*

Discharge Instructions: *no activity restriction, regular diet, follow up in two to three weeks with regular physician.*

Unstructured Data - Multimedia Content

Image2Caption



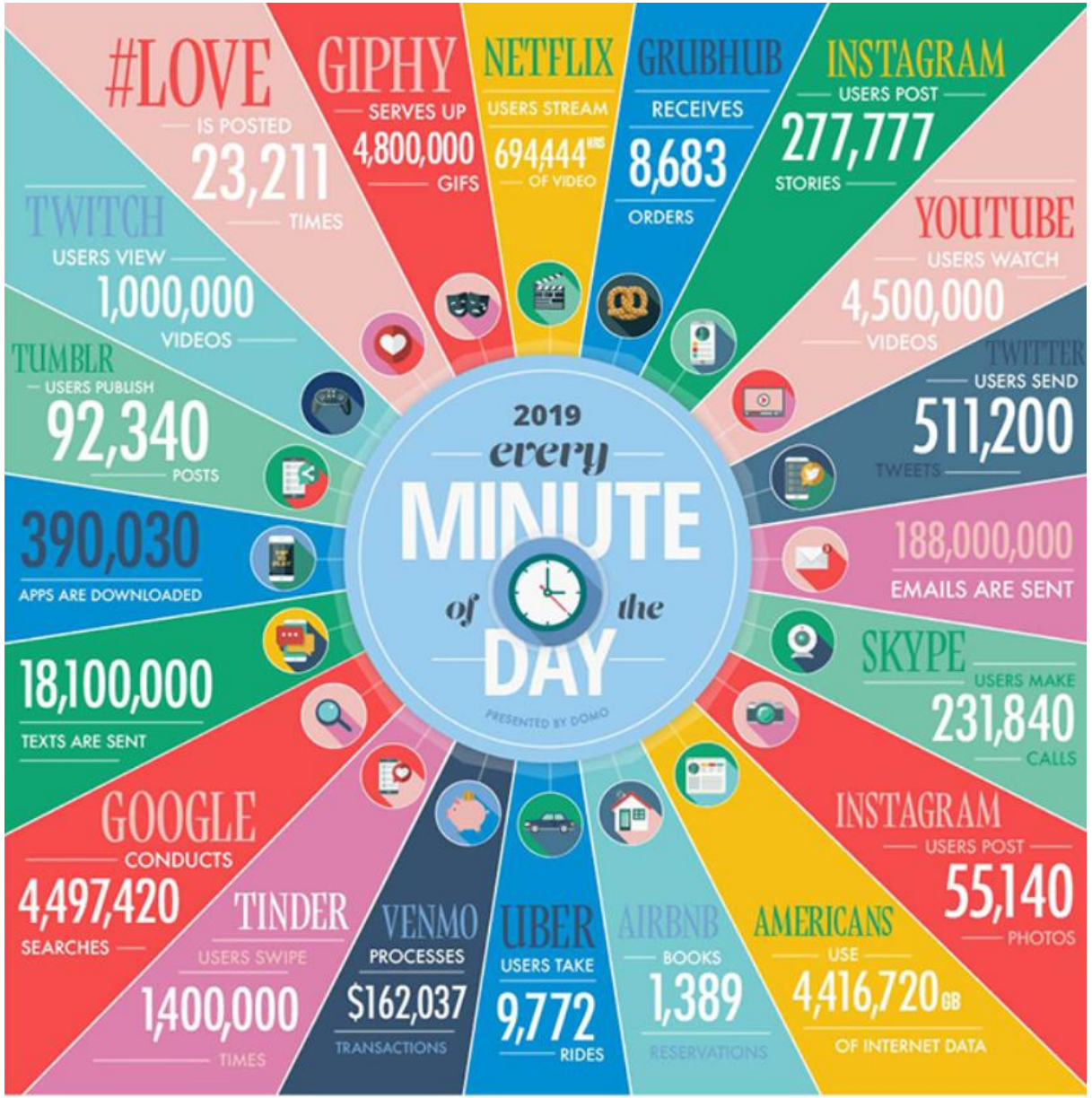


https://image2caption.pascalperle.de/

What is Big Data?

- Big data sets are too large or complex to be processed by traditional methods.

Consider that **in a single minute** (2019) there are:

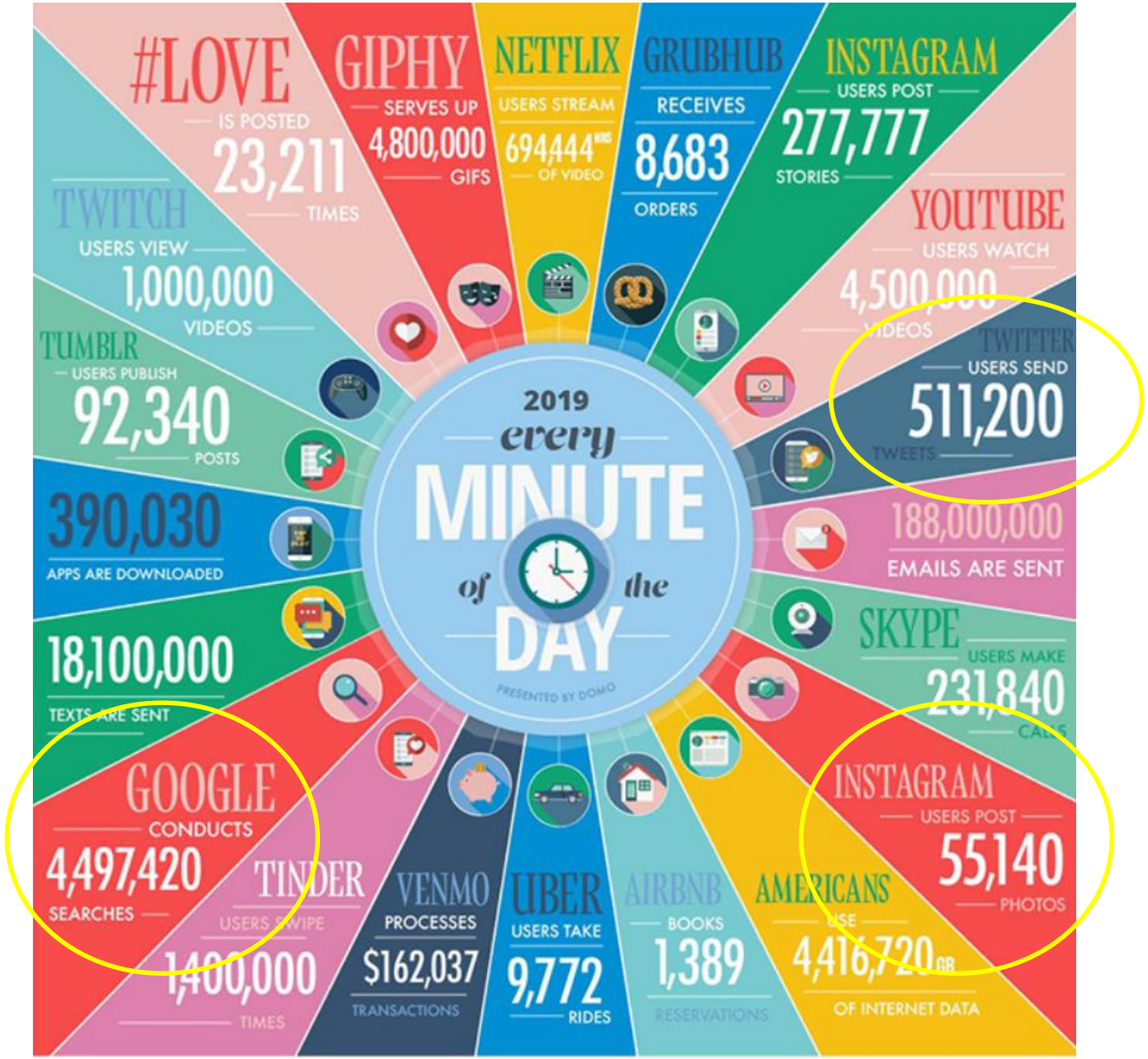


https://web-assets.domo.com/blog/wp-content/uploads/2019/07/data-never-sleeps-7-896kb.jpg

What is Big Data?

- Big data sets are too large or complex to be processed by traditional methods.

Consider that **in a single minute** (2019) there are:

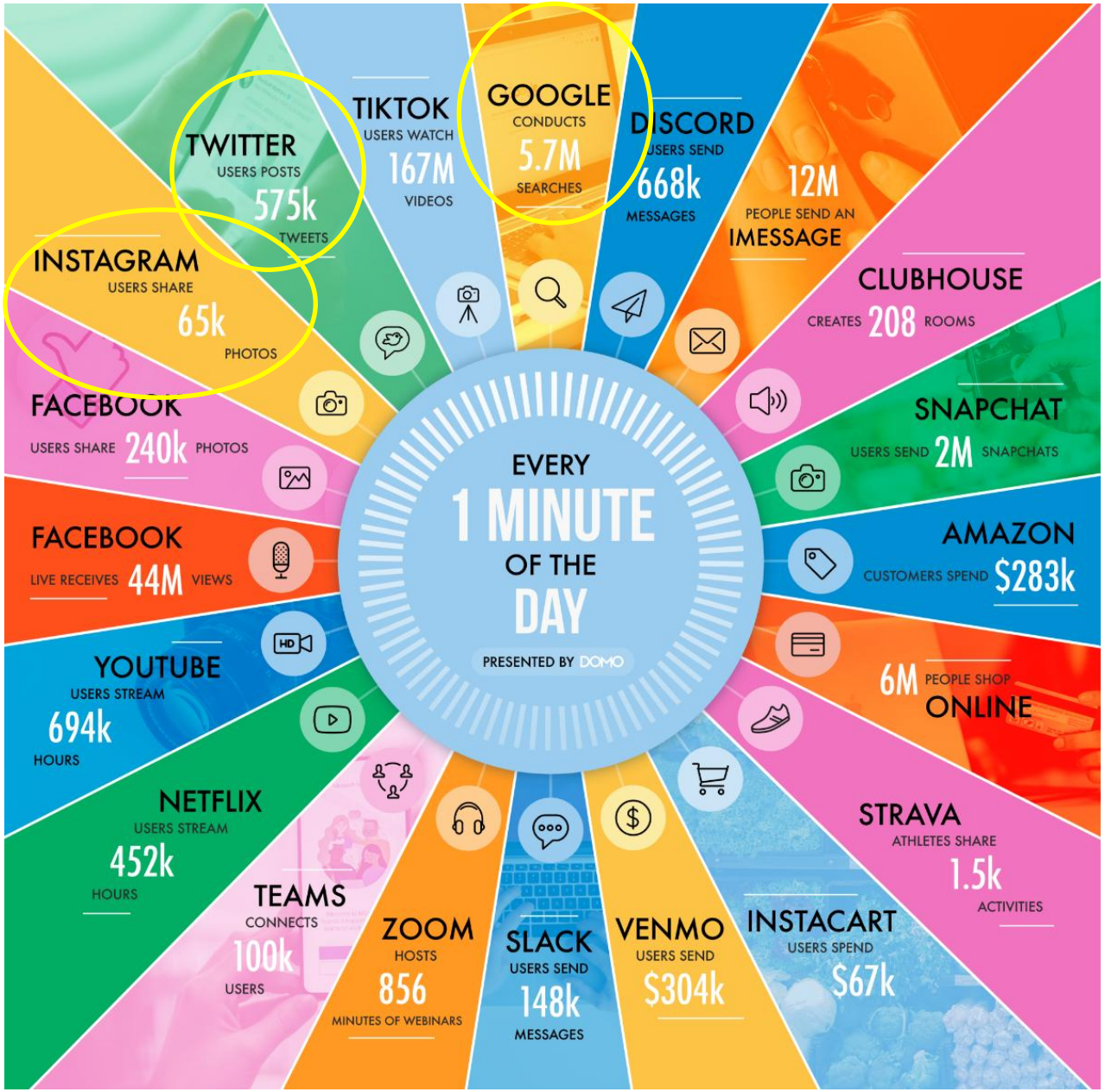


https://web-assets.domo.com/blog/wp-content/uploads/2019/07/data-never-sleeps-7-896kb.jpg

What is Big Data?

- Big data sets are too large or complex to be processed by traditional methods.

Consider that **in a single minute** (2021) there are:



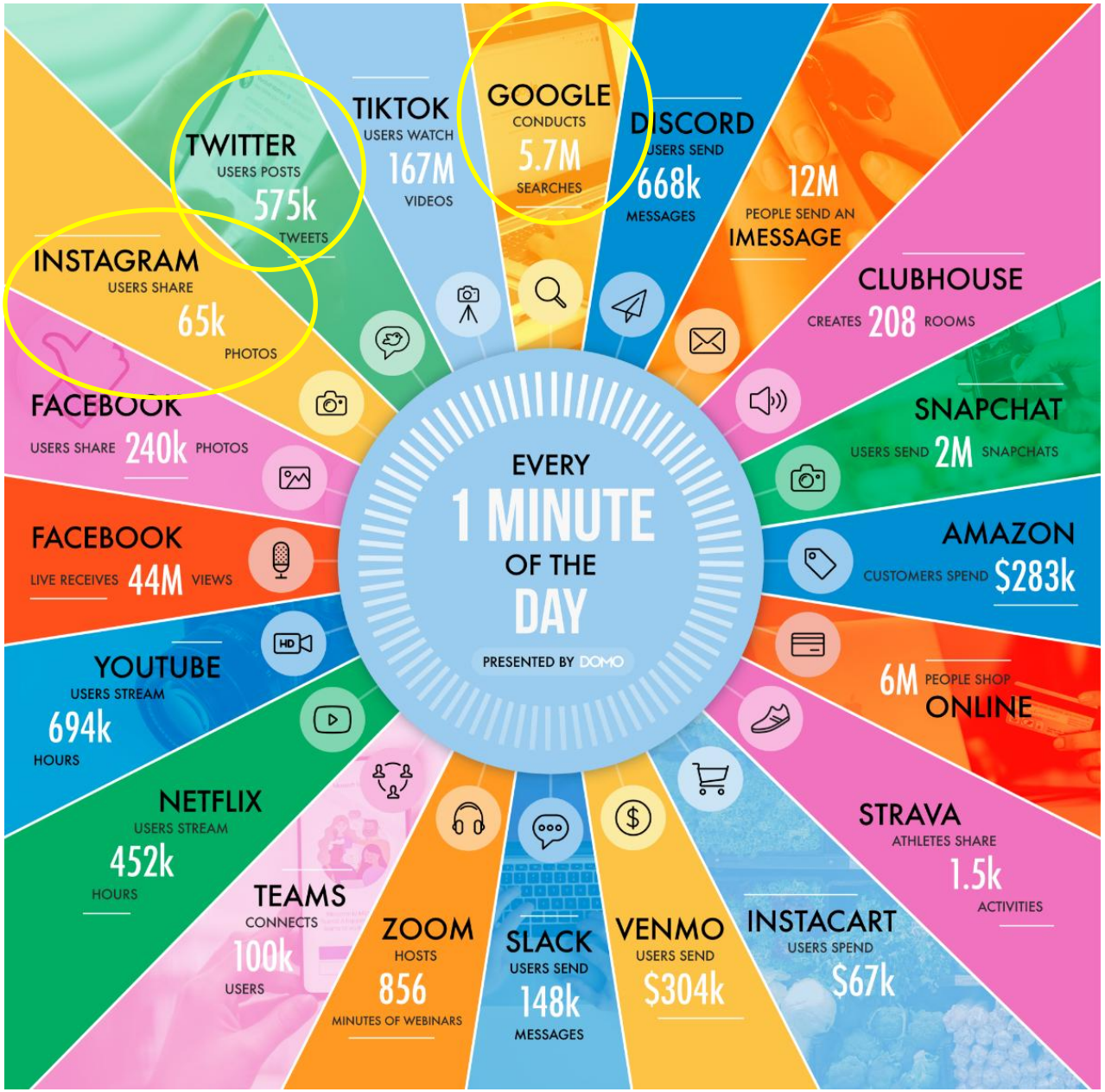
https://web-assets.domo.com/blog/wp-content/uploads/2021/09/data-never-sleeps-9.0-1200px-1.png

What is Big Data?

- Big data sets are too large or complex to be processed by traditional methods.

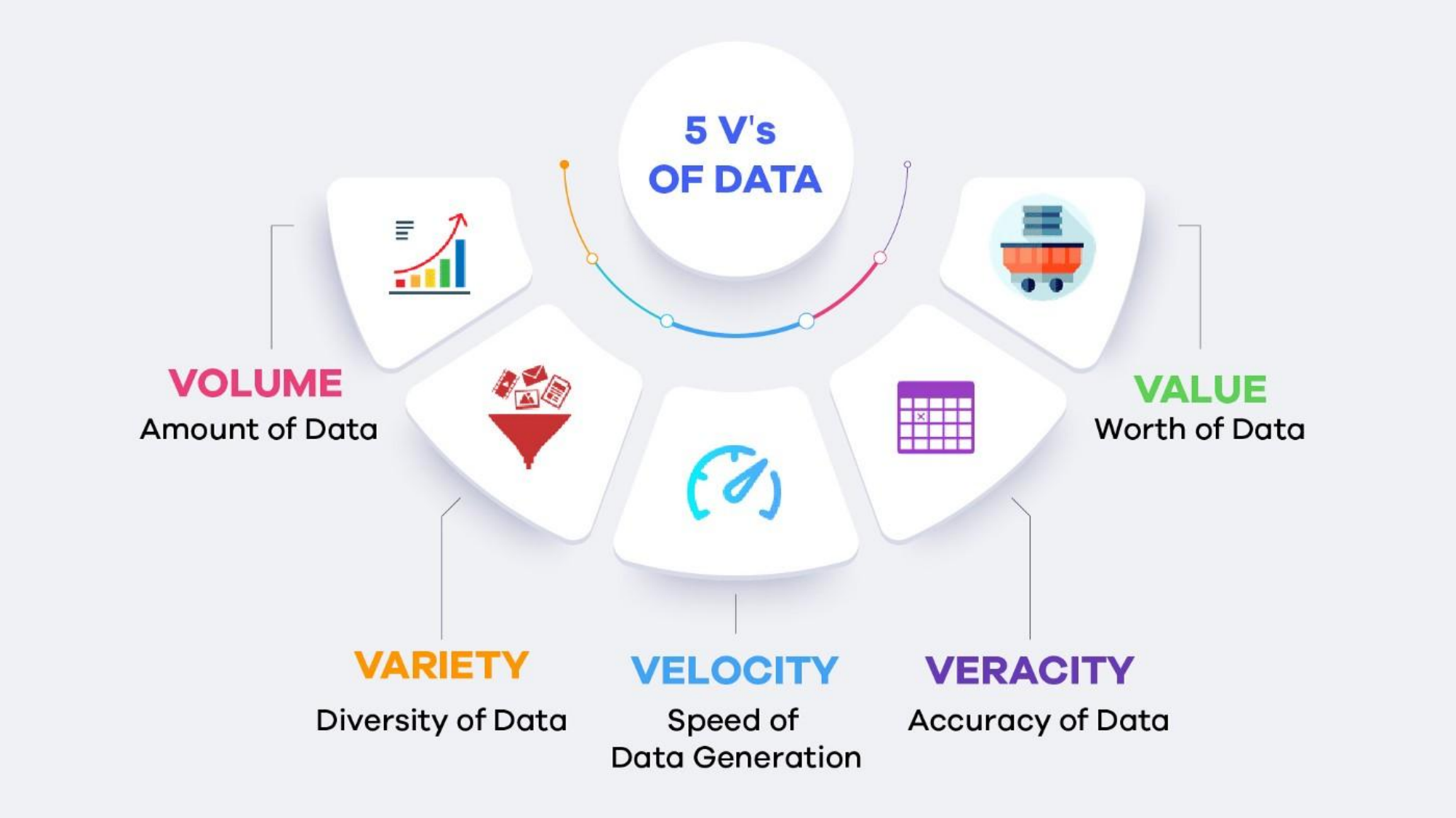
Consider that **in a single minute** (2021) there are:

Activity	2019	2021	Increase
Google Search	4.5M	5.7M	26.7%
Instagram Photos	55K	65K	18.2%
Twitter Tweets	511K	575K	12.5%



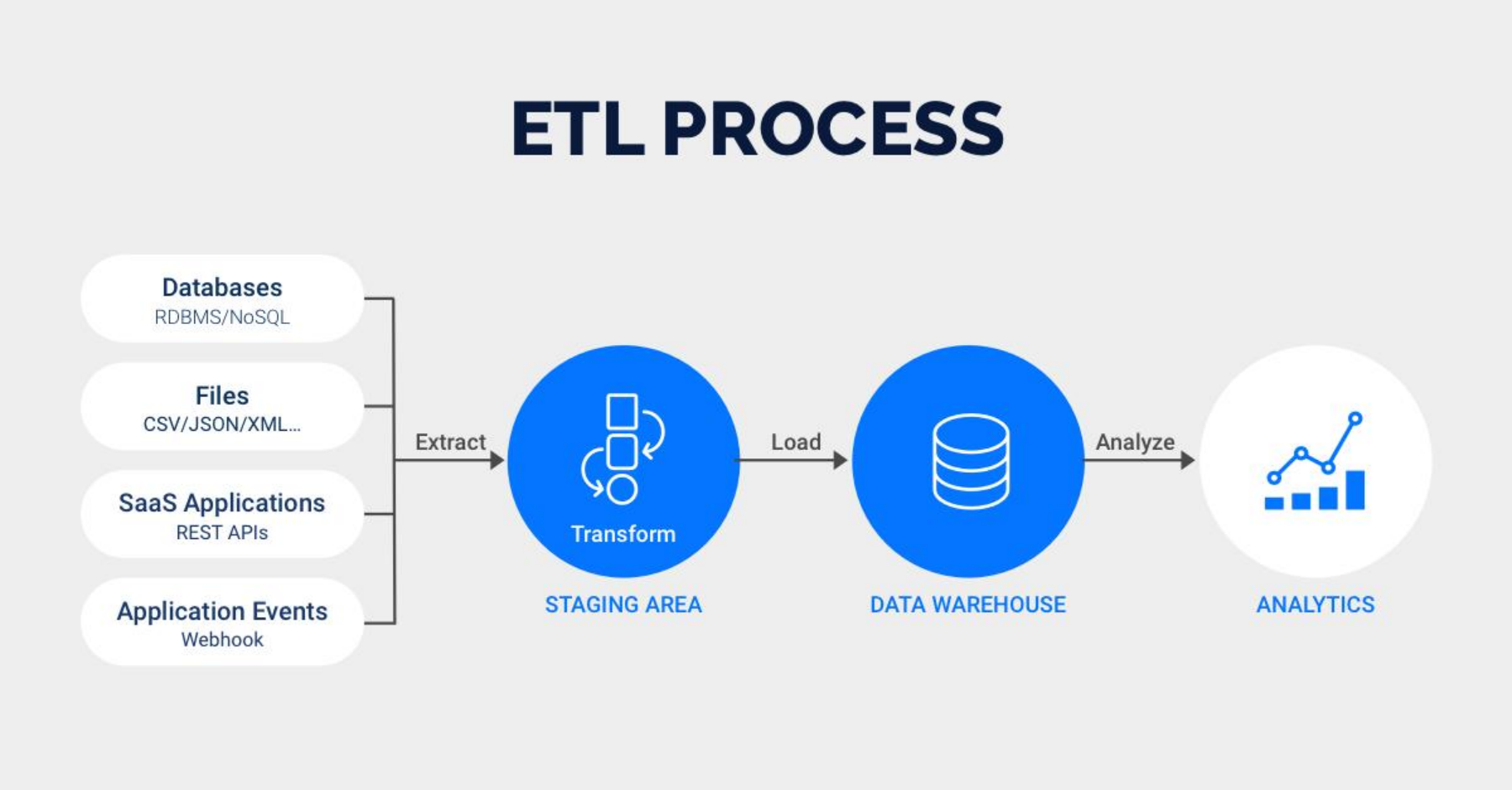
https://web-assets.domo.com/blog/wp-content/uploads/2021/09/data-never-sleeps-9.0-1200px-1.png

The 5 Vs of Big Data



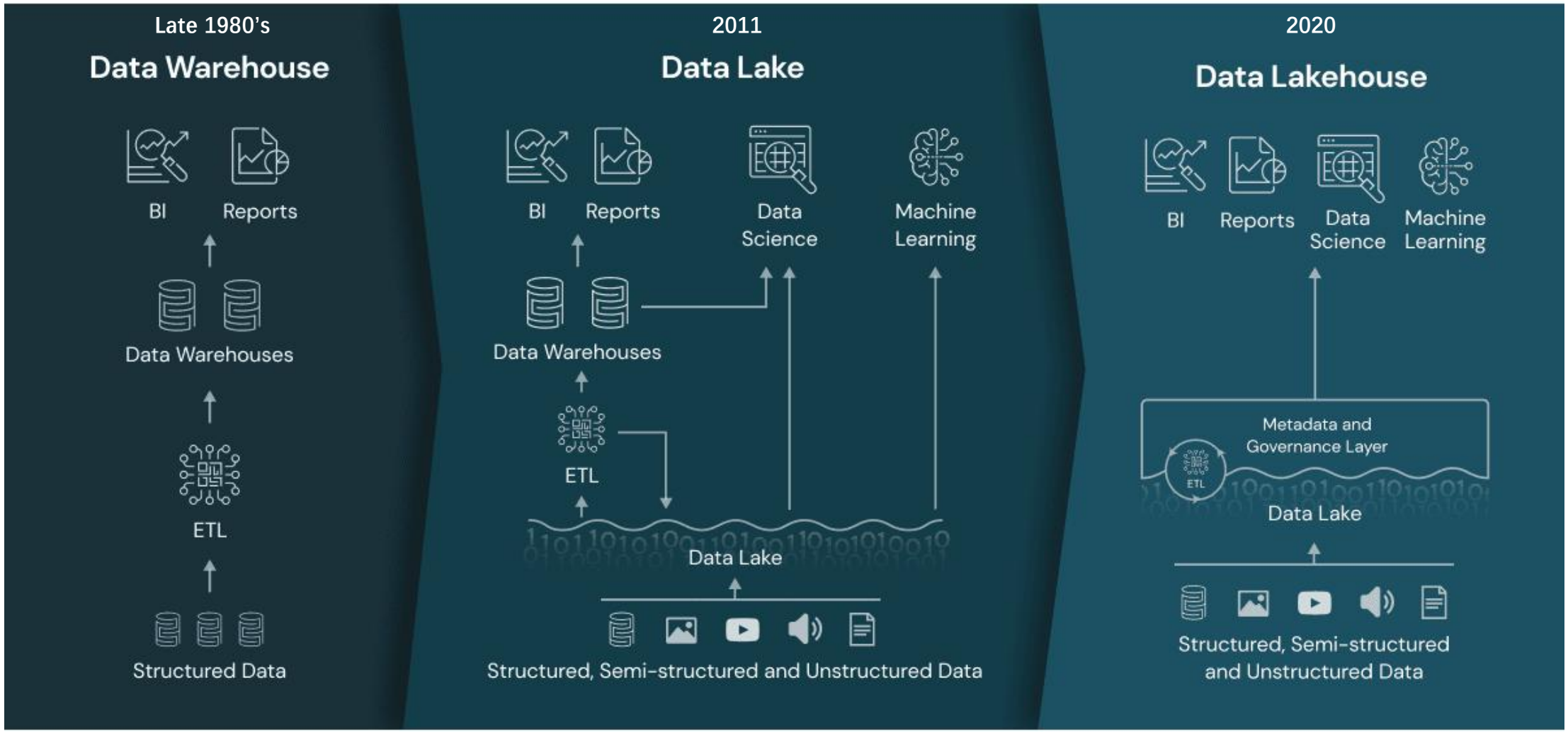
https://medium.com/@get_excelsior/big-data-explained-the-5v-s-of-data-ae80cbe8ded1

Data Warehouse & ETL (Extract, Transform, Load)



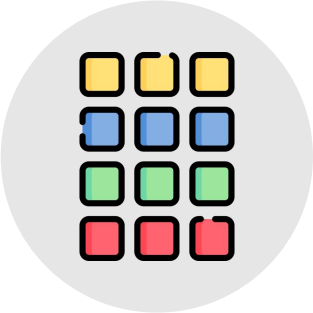
https://rivery.io/wp-content/uploads/2020/05/ETL-Process-for-linkedin3.png

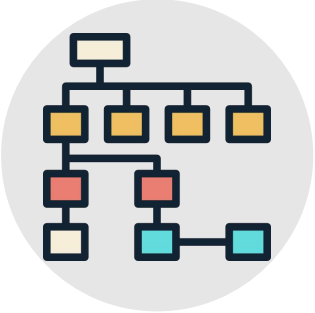
Data Lakehouse

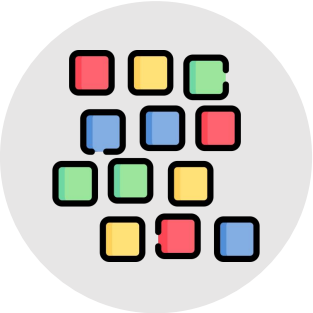


https://www.databricks.com/wp-content/uploads/2020/01/data-lakehouse-new-1024x538.png

Summary







	Structured	Semi-Structured	Unstructured
Definition	• Data with predefined schema	• Data with flexible schema (e.g., XML)	• Data without predefined schema

Database Systems	• Relational Database
	  • MySQL, PostgreSQL
		• MongoDB/HBase	• No-SQL Databases
			  • Object store (S3)

Query Languae	• SQL	• XPath • XQuery	• ElasticSearch for text

Course Overview

- This course is about managing and retrieving different types of data, i.e., structured, semi-structured and unstructred data using different *database sysmtems*

- "*Data management* is the practice of collecting, keeping, and using data securely, efficiently, and cost-effectively." - *Oracle*

- "*Data retrieval* means obtaining data from a Database Management System (DBMS)" - Wikipedia
  - Find or extract information

Topics - Relational Database

- Introduction to Database Systems and Relational Model
- Intermediate & Advanced SQL
- Entity-Relationship Model & Relational Database Design
- Complex Data Types
- Big Data & Data Analytics
- Indexing and Hashing
- Query Processing
- Transactions

Topics - Semi-Structured Data Management

- XML
- JSON
- XPath & XQuery

Topics - Modern Data Stack

- ETL / ELT
- ELK - Elastic search for text
- No-SQL Database Systems

Late Policy

- Each deadline will be extended once with penalty.
- You will lose 30% of the points for a project or homework if you submit your assignment by the extended deadline.
- No submission is allowed beyond the extended deadline.
- Please contact the instructor ASAP if something comes up.

Thank You