



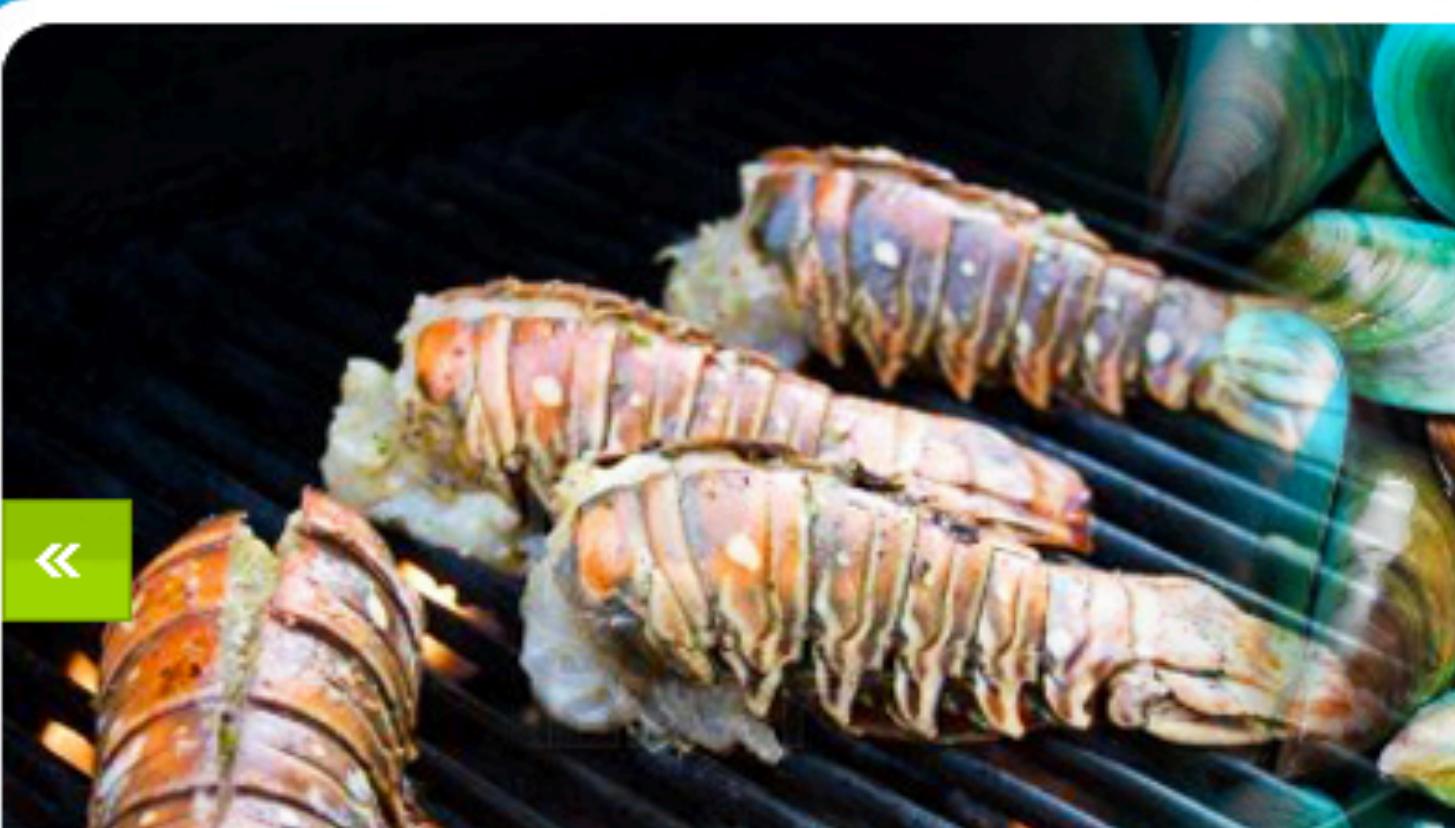
Problem Statement



Home

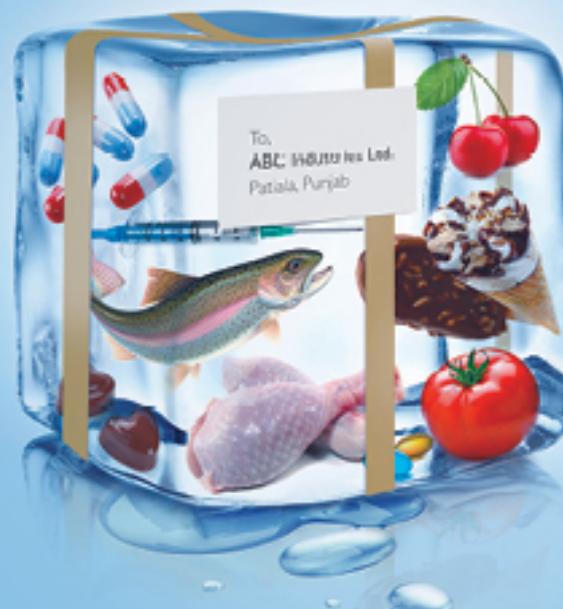
Services

- ❖ Black Tiger Shrimps
- ❖ Vannamei Shrimps
- ❖ Scampi
- ❖ Pangasius (Basa Fish)
- ❖ Crustaceans
- ❖ Fishes
- ❖ Cephalopods





When technology leads the way,
freshness follows.

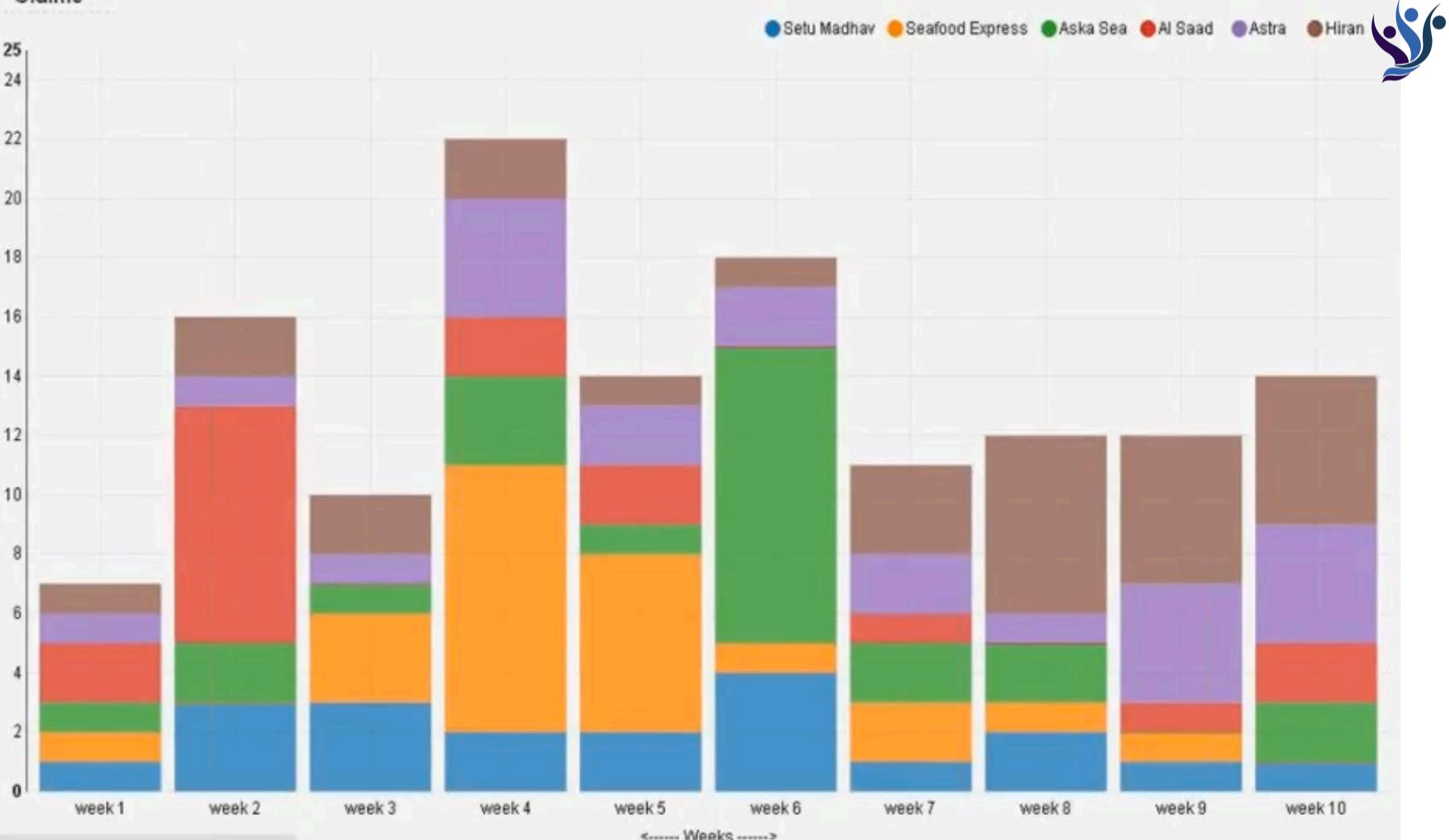


From Pickup to Destination:
Delivered Fresh, Delivered Safe.



Claims from Clients

Claims

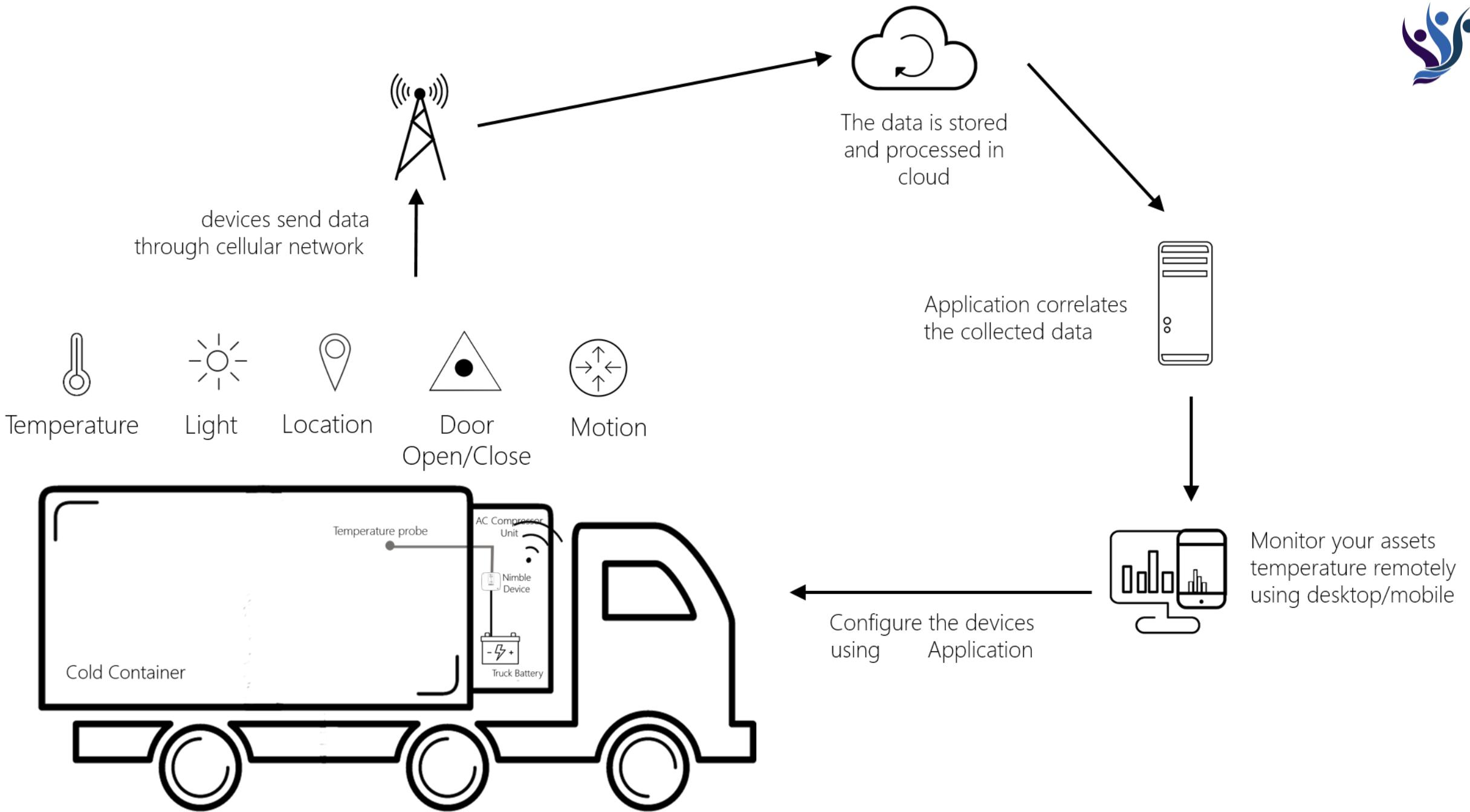




How To Filter False Claims



Solution

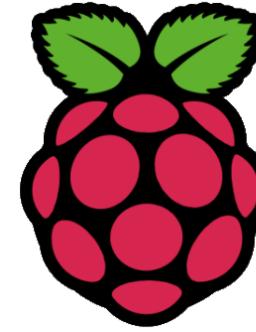
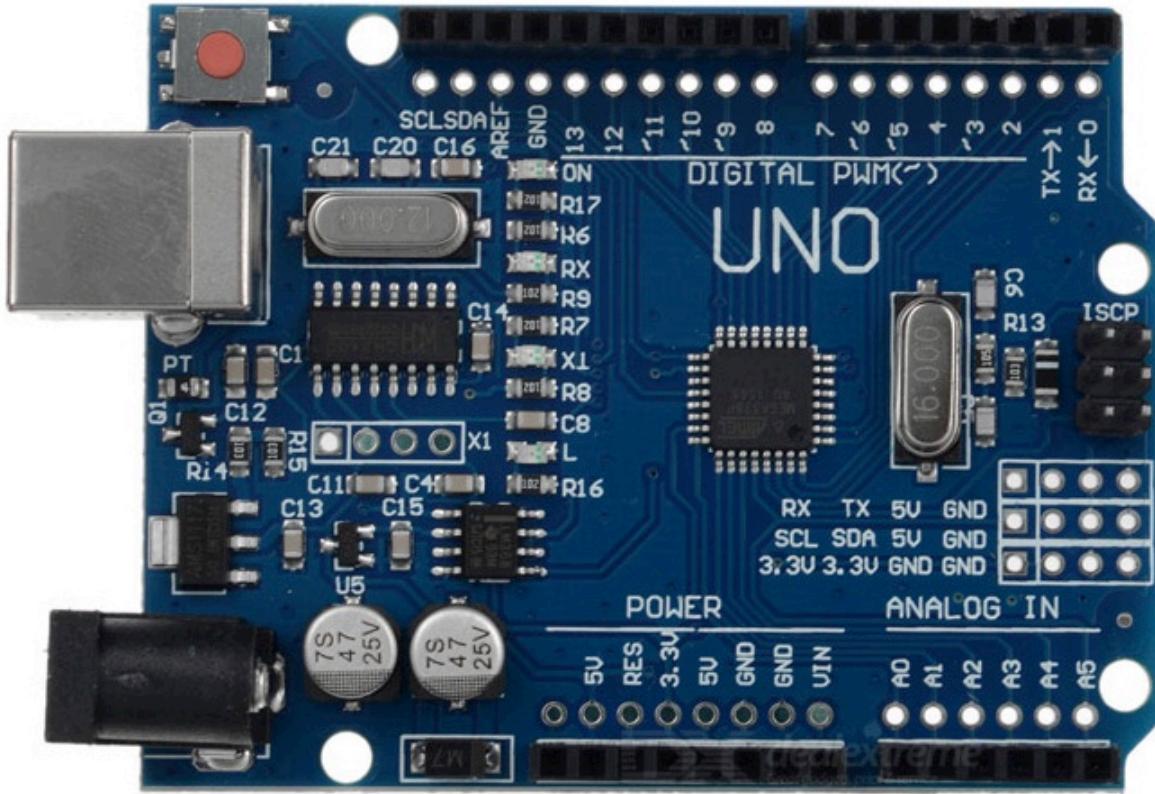




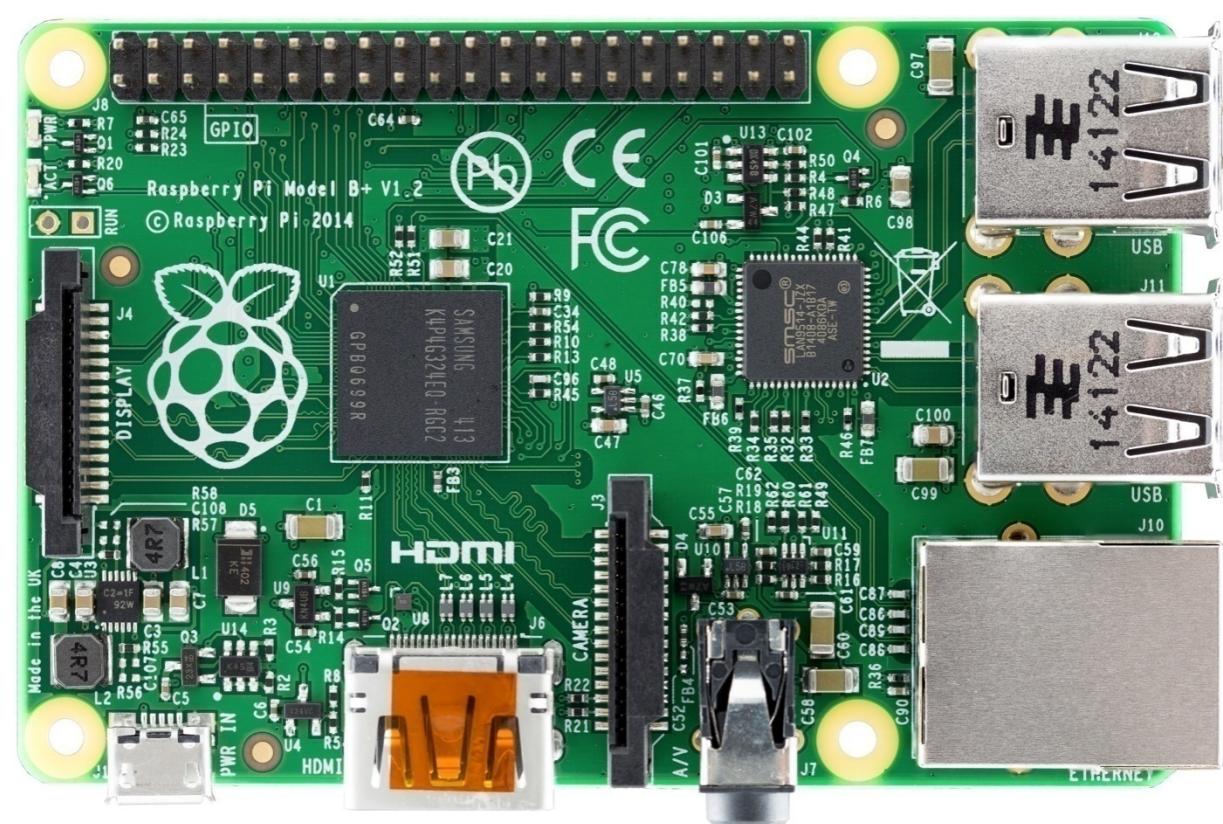
Execution



Development Boards



RaspberryPi





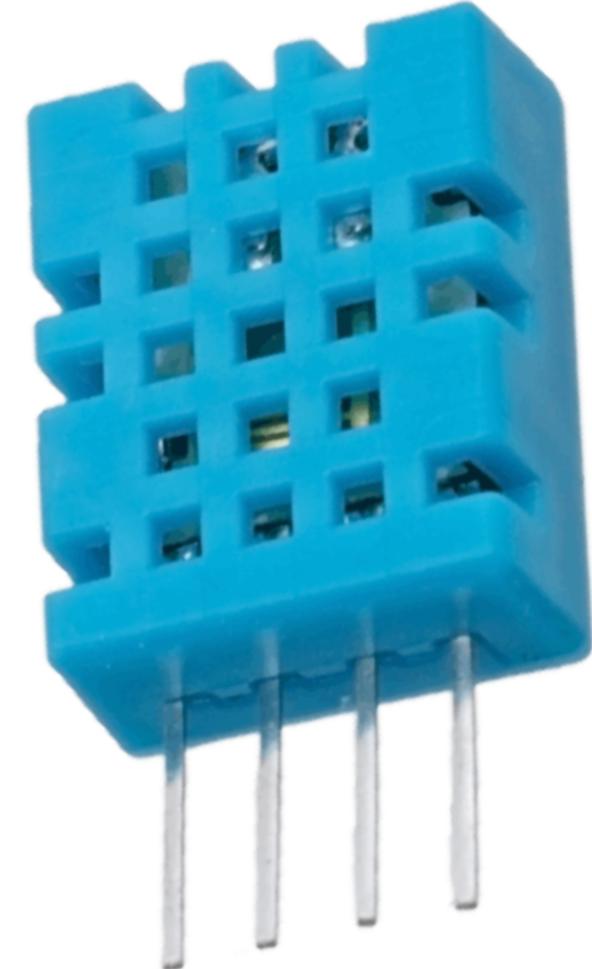
Hardware Requirements



LDR



LM35D



DHT11



Hardware Requirements



PIR



PC817



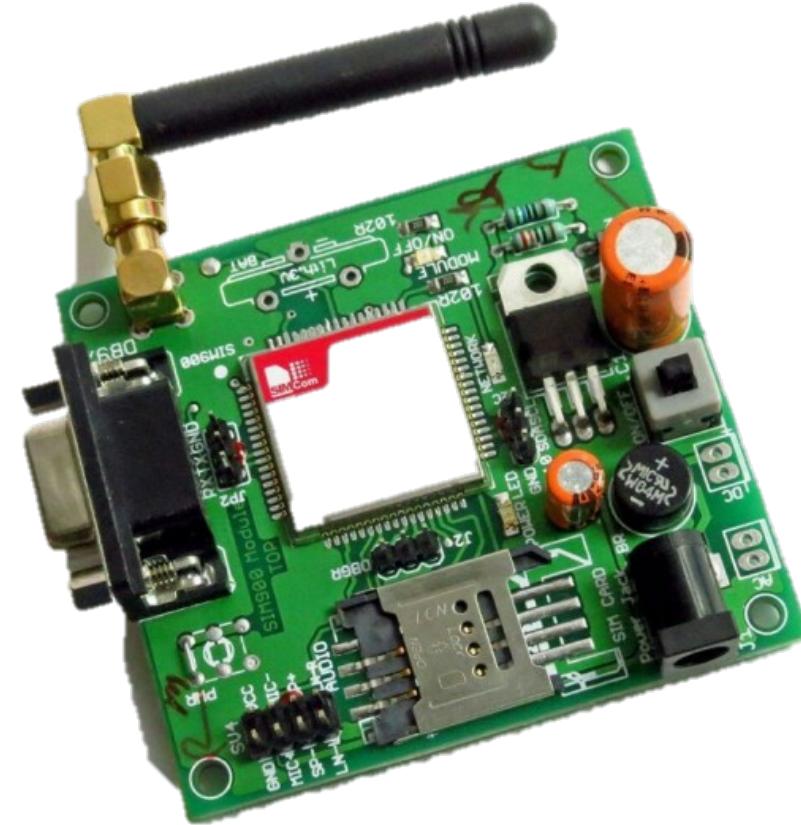
REED



Hardware Requirements

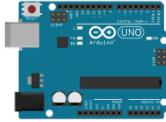
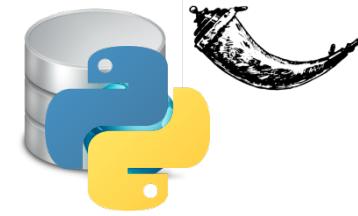
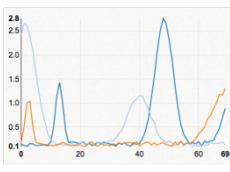


GPS



SIM800

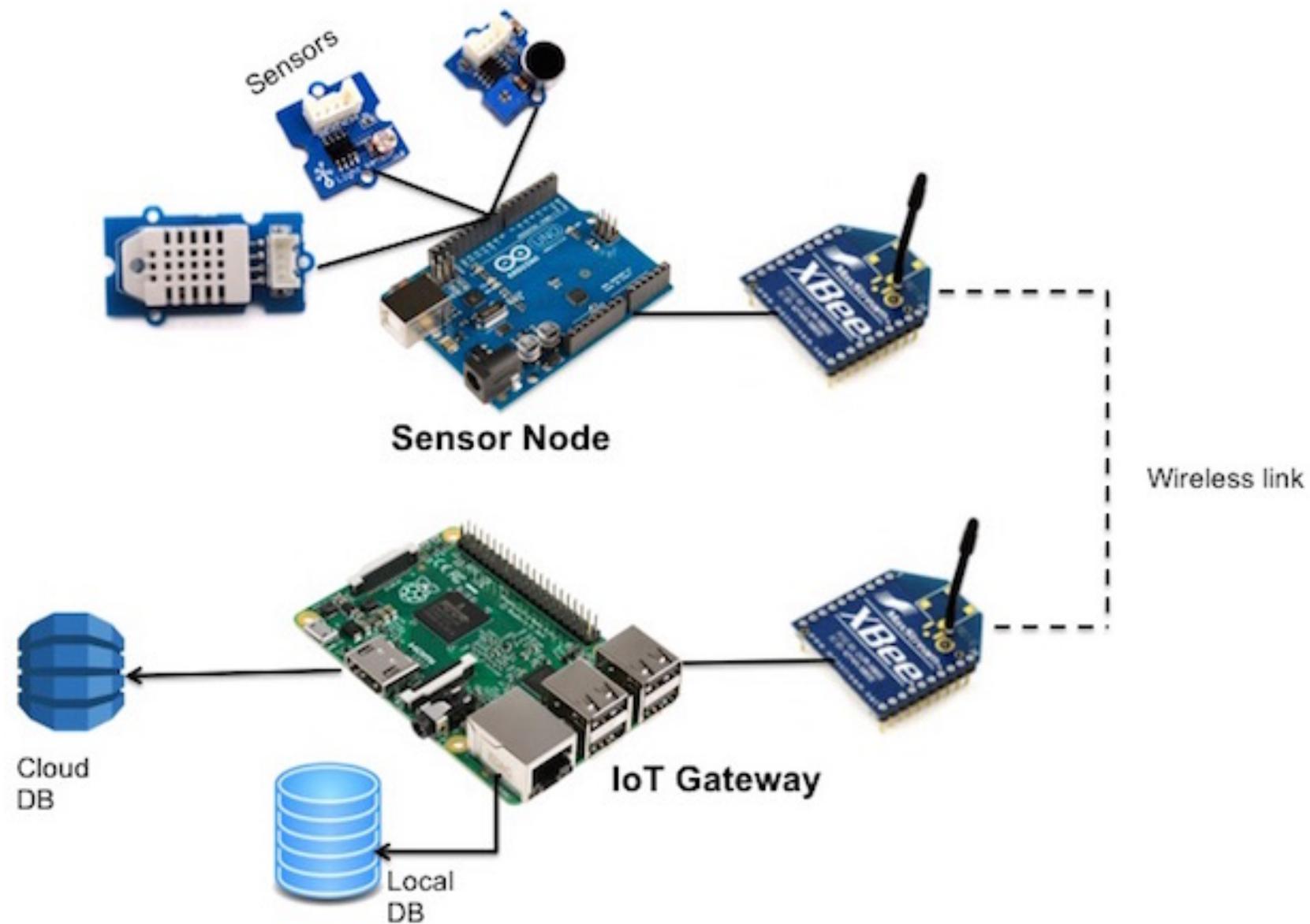
Practical Flow for Best Cold Chain Problem

Sensors	Nodes	Gateway	Web Services	Data Stores	Analytics	Visualization
 Read, Calibrate and Display data from sensors   	 Develop code for reading SMS through Arduino and trigger the Alarm.	 Send the data to cloud using GPRS	 Create a REST based Web Service to send truck sensor data on the cloud    Learn how to create cloud based programs	 Write the program to store the truck sensor data in NoSql Database.	 Analyze the data to solve business problem.	 Generate a neat report to visualize the data for management.  View and Control from Mobile App  



Introduction to Sensor, Nodes & Gateway

Sensor Node and Gateway





IoT Clouds

- Device management and integration support
- Information security
- Data collection protocols
- Data analytics



IoT Software Platforms (Cloud)

Artik Cloud

ThingWorx

Autodesk Fusion Connect

Salesforce IoT Cloud

AWS IOT

Telit DeviceWise

GE Predix

Xively

Google Cloud IoT

Zebra Zatar Cloud

Microsoft Azure IoT Suite

Kaa (**Open Source**)

IBM Watson IoT

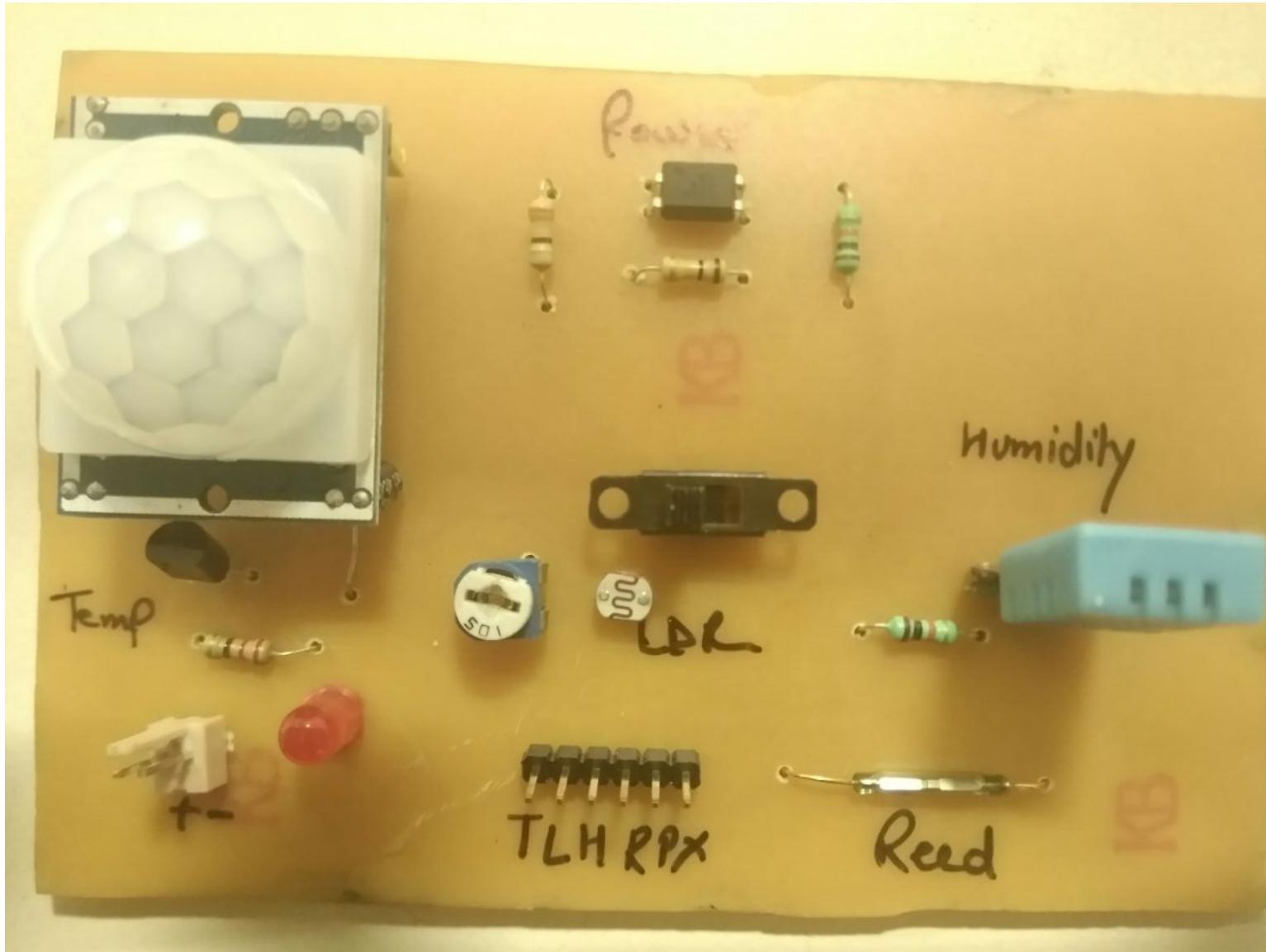
✓ ThingSpeak (**Open Source**)



Introduction to Sensor Board

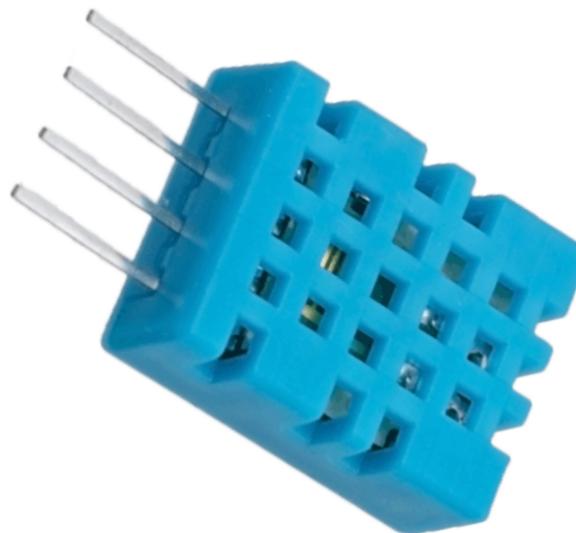
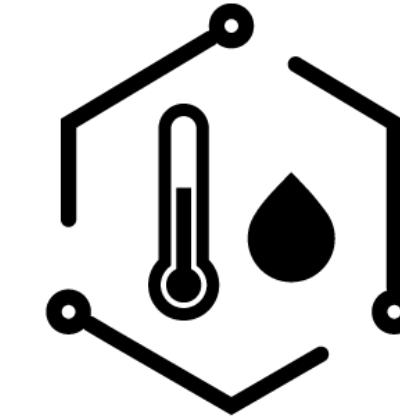
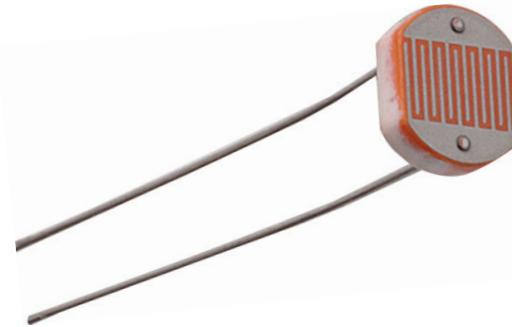
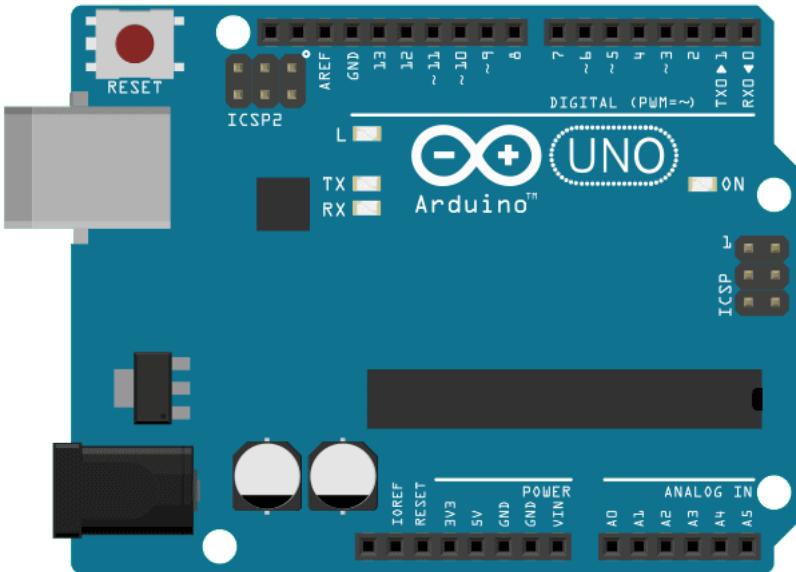


Sensor Board



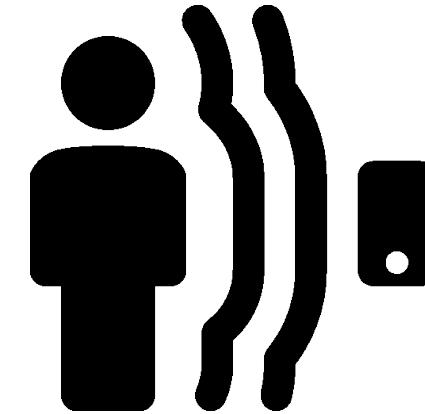
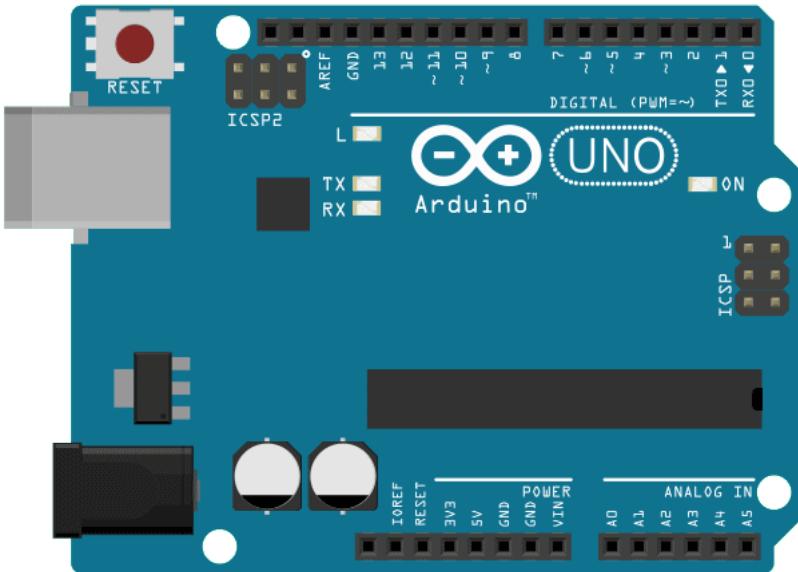


Analog Sensors





Digital Sensors

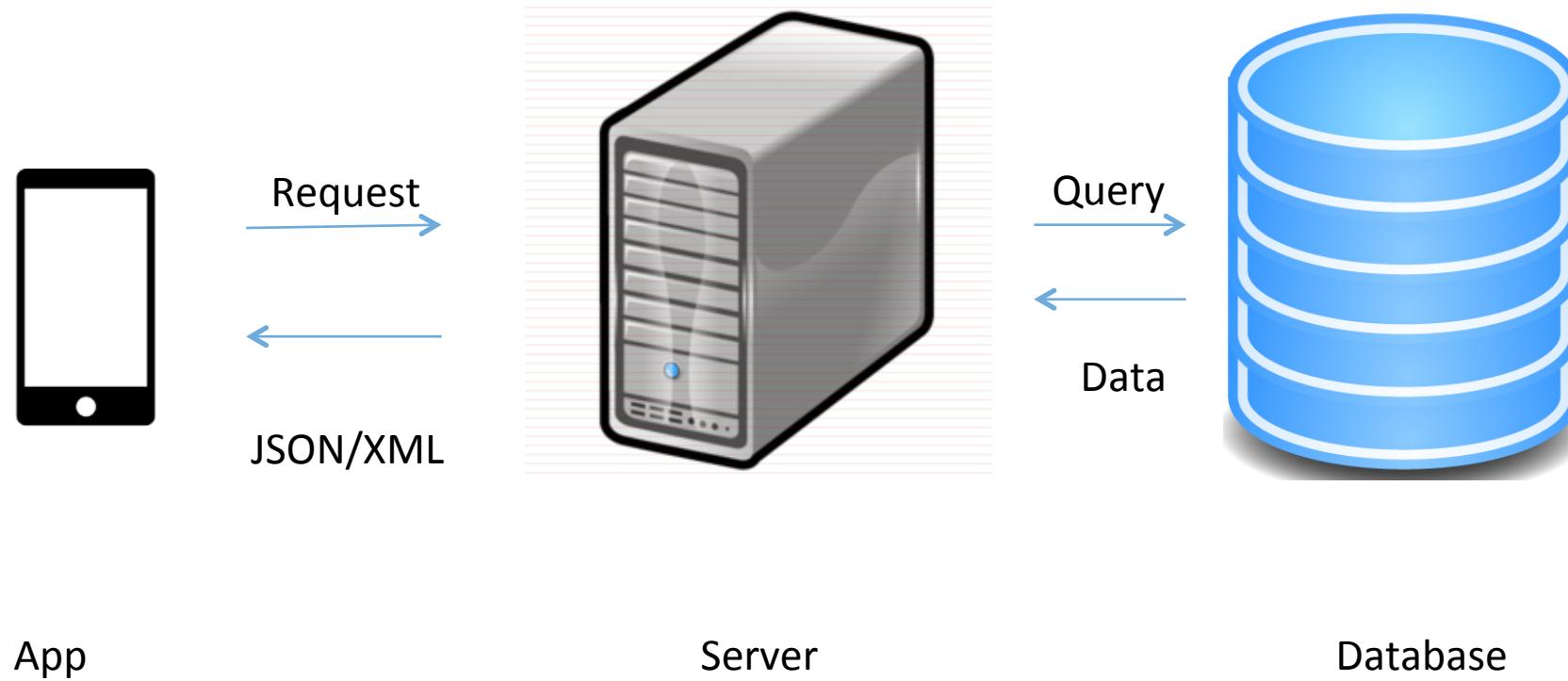




Data Exchange Mechanisms

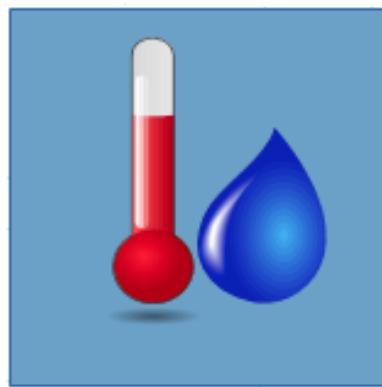


Data Exchange

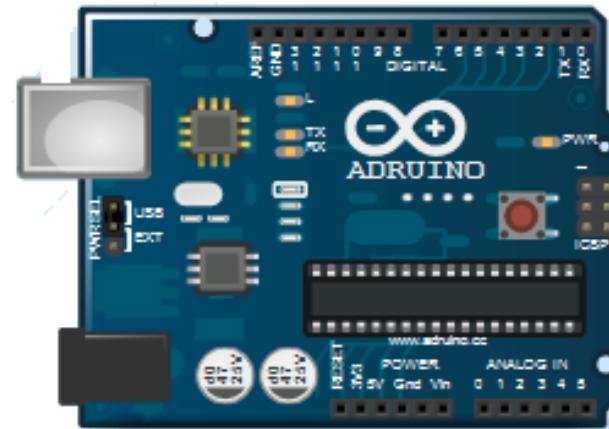
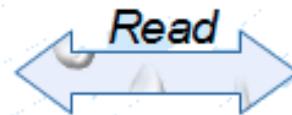




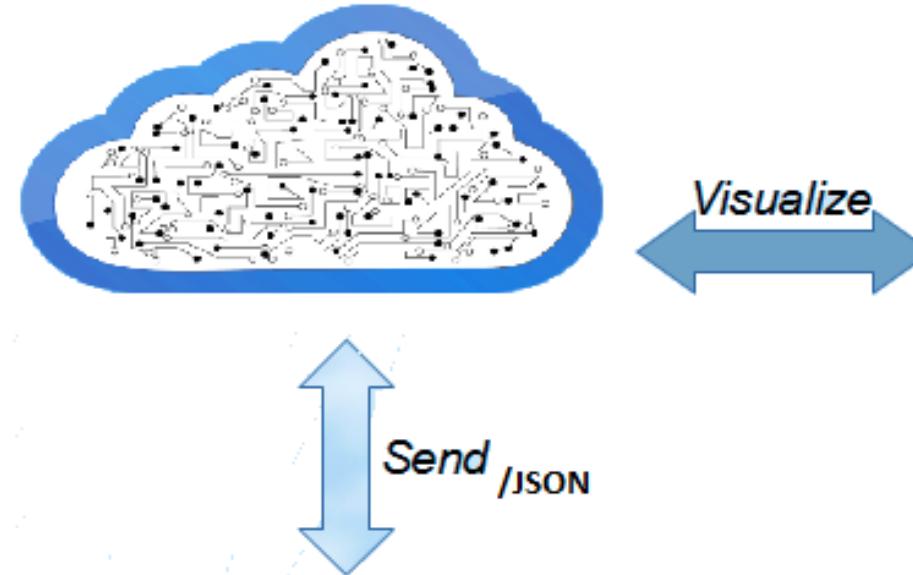
Summary



Sensor



Arduino





Cloud Signup



Step 1: Open <https://thingspeak.com/>

The screenshot shows the ThingSpeak website homepage. At the top, there's a navigation bar with links for File, Edit, View, History, Bookmarks, Tools, Window, Help, and a search bar. Below the navigation is a banner featuring a city skyline at night with a network of glowing nodes overlaid, symbolizing connectivity. The banner includes the text "ThingSpeak™", "Channels", "Apps", "Community", "Support", "How to Buy", "Log In", and a prominent "Sign Up" button, which is highlighted with a large red circle. Below the banner, the main headline reads "Understand Your Things" followed by the subtext "The open IoT platform with MATLAB analytics." There are two call-to-action buttons: a green "Get Started For Free" button and a white "Learn More" button. At the bottom of the page, there are three sections: "Collect" (with an icon of a cloud and arrows), "Analyze" (with an icon of a bar chart), and "Act" (with an icon of a person running). Each section has a brief description: "Send sensor data to the cloud.", "Analyze and visualize your data with MATLAB.", and "Trigger a reaction." respectively.

ThingSpeak™

Channels Apps Community Support

How to Buy Log In Sign Up

Understand Your Things

The open IoT platform with MATLAB analytics.

Get Started For Free Learn More

Collect

Analyze

Act

Send sensor data to the cloud.

Analyze and visualize your data with MATLAB.

Trigger a reaction.



How to Export Data in csv Format

Screenshot of a Firefox browser window showing the ThingSpeak channel page for "coldchain".

The browser toolbar at the top includes: Apple icon, Firefox, File, Edit, View, History, Bookmarks, Tools, Window, Help, and several status icons.

The address bar shows: Data Import / Export - Thin... | https://thingspeak.com/channels/222415/import_export

The ThingSpeak navigation bar includes: ThingSpeak™, Channels, Apps, Community, Support, How to Buy, Account, and Sign Out.

The main content area displays:

- Channel ID: 222415**
- Author: sylvesterf**
- Access: Public**
- Captures the data for the cold chain problem**

A red oval highlights the **Data Import / Export** tab in the navigation bar.

Import: Upload a CSV file to import data into this channel. A "Browse..." button shows "No file selected." Below it is a "Time Zone" dropdown set to "(GMT+00:00) UTC". A green "Upload" button is present.

Export: Download all of this Channel's feeds in CSV format. A green "Download" button is present.

Help: Select a CSV file on your hard drive and import all of its data directly into this channel. Your CSV file should contain a date field in the first column. If your data doesn't contain timezone info, select one appropriately. [Learn More](#)

API Requests:

- Update Channel Feed - GET**: GET https://api.thingspeak.com/update?api_key=0C277GP1H26YHUJ0&field1=73
- Update Channel Feed - POST**: POST <https://api.thingspeak.com/update.json>
api_key=0C277GP1H26YHUJ0
field1=73
- Get a Channel Feed**: GET <https://api.thingspeak.com/channels/222415/feeds.json?results=1>
- Get a Channel Field Feed**: GET <https://api.thingspeak.com/channels/222415/fields/1.json>



Mapping

Field 1

id

Field 2

start_location

Field 3

end_location

Field 4

driver_name

Field 5

customer

Field 6

material

Field 7

device_no

Field 8

type

trips

Field 1

door_operation

Field 2

power_status

Field 3

temperature

Field 4

humidity

Field 5

trip_id

Field 6

device_type

Field 7

device_no

Field 8

type

transaction



Postman



POST Method

- URL: http://api.thingspeak.com/update?api_key=J0WJOEXGJNF0QNC8

- Choose type from dropdown: **POST**

- Send data in body

- Choose **raw** type, **application/json**

- JSON data

- Change your phone number(**field7**)
 - Change your trip_id(**field5**)
 - Last 3 digits of your mobile number

```
{  
    "field1" : "1",                      // door_operation  
    "field2" : "0",                      // power_status  
    "field3" : "23",                     //temperature  
    "field4" : "45.0",                   //humidity  
    "field5" : "120",                    //trip_id  
    "field6" : "Cold Chain",            //device_type  
    "field7" : "9928492120",           //device_no  
    "field8" : "transaction"          //type  
}
```



GET Method

- URL:
https://api.thingspeak.com/channels/223662/feeds.json?api_key=35FCI04R5Y01A7VQ
- Choose type from dropdown: **GET**
- Result
 - JSON response from the cloud

To get only last 3 results

- [https://api.thingspeak.com/channels/223662/feeds.json?
api_key=NB5UI22JS0VTPVEE&results=3](https://api.thingspeak.com/channels/223662/feeds.json?api_key=NB5UI22JS0VTPVEE&results=3)

To get specific field data

<https://api.thingspeak.com/channels/223662/fields/1.json?results=2>



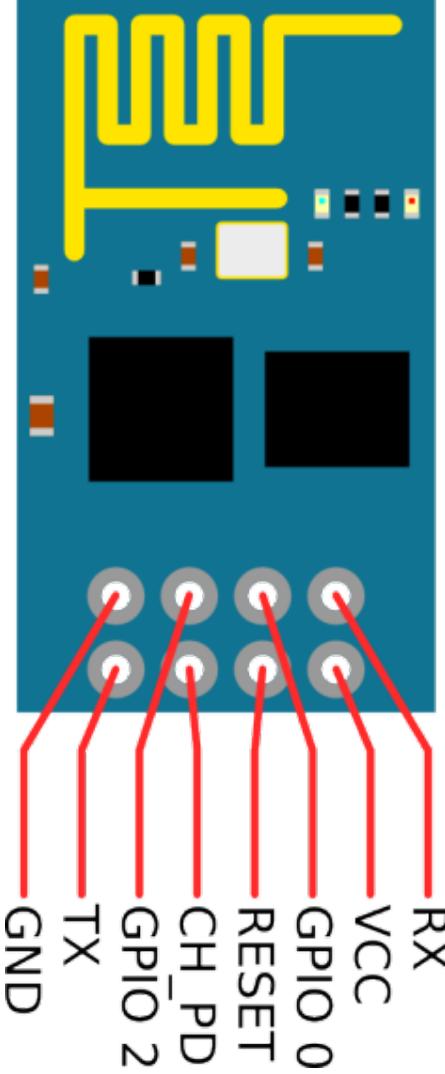
Gateway using WiFi

To View the Public Data

<https://thingspeak.com/channels/223662>



ESP8266 (WiFi) Notation

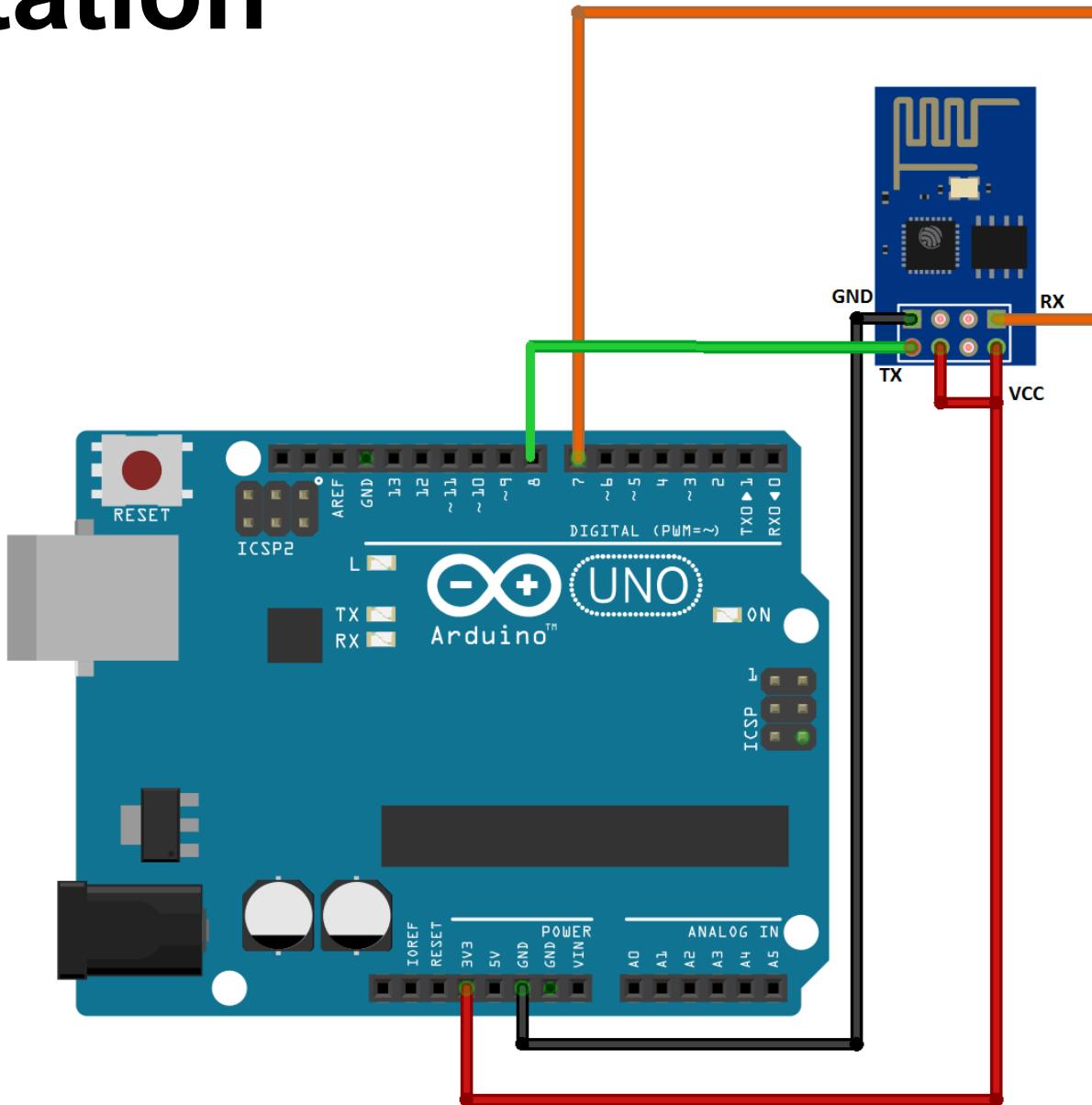


VCC to 3.3V of Arduino

GND to GND of Arduino

Rx – to Arduino PIN 7

Tx – to Arduino PIN 8





What is MongoDB

- <http://www.mongodb.org/downloads>
 - Local installer available at `~/day2/Installers`
- Scalable High-Performance Open-source
- Document-orientated database
- Built for Speed
- Map / Reduce for Aggregation
- Supports JSON



Why MongoDB

- SQL was invented in the 70's to store data
- MongoDB stores documents (or) objects
- Now-a-days, everyone works with objects (Python/Ruby)
- And we need Databases to persist our objects
 - Then why not store objects directly ?



Analytics (MapReduce)



Map and Reduce

- Programming model for processing and generating large data sets
- **map function** : processes documents from your dataset into **key-value** pairs
- **reduce function** : **combines** the set returned by map



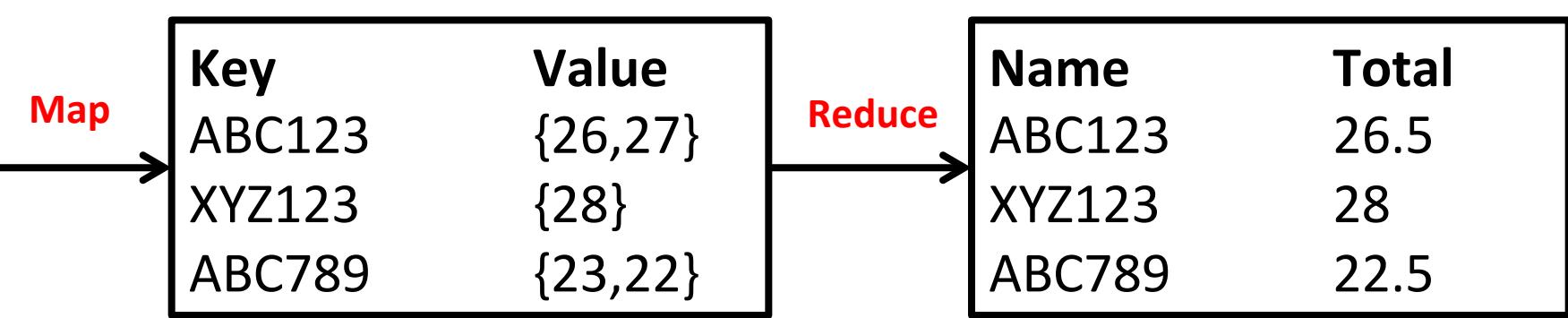
Map and Reduce

- A reduce function is passed a set of intermediate values and combines them **to a single value**
- Popular implementations of MapReduce: **Hadoop, Disco, Skynet**



Understand MapReduce

```
{  
  {  
    device:ABC123,  
    Temperature:26  
  },  
  {  
    device:ABC123,  
    Temperature:27  
  },  
  {  
    device:XYZ123,  
    Temperature:28  
  },  
  {  
    device:ABC789,  
    Temperature:23  
  },  
  {  
    device:ABC789,  
    Temperature:22  
  },  
}
```





Real Problem?

- Best cold chain company is facing huge claims for delivery of spoilt fish
- Analyse genuine claims using data stored on cloud
 - They have claimed many cases of spoilt fish
 - You need to check if this number is correct
 - Apply analytics using MapReduce concept to solve this



Visualization

From Data to Graphics

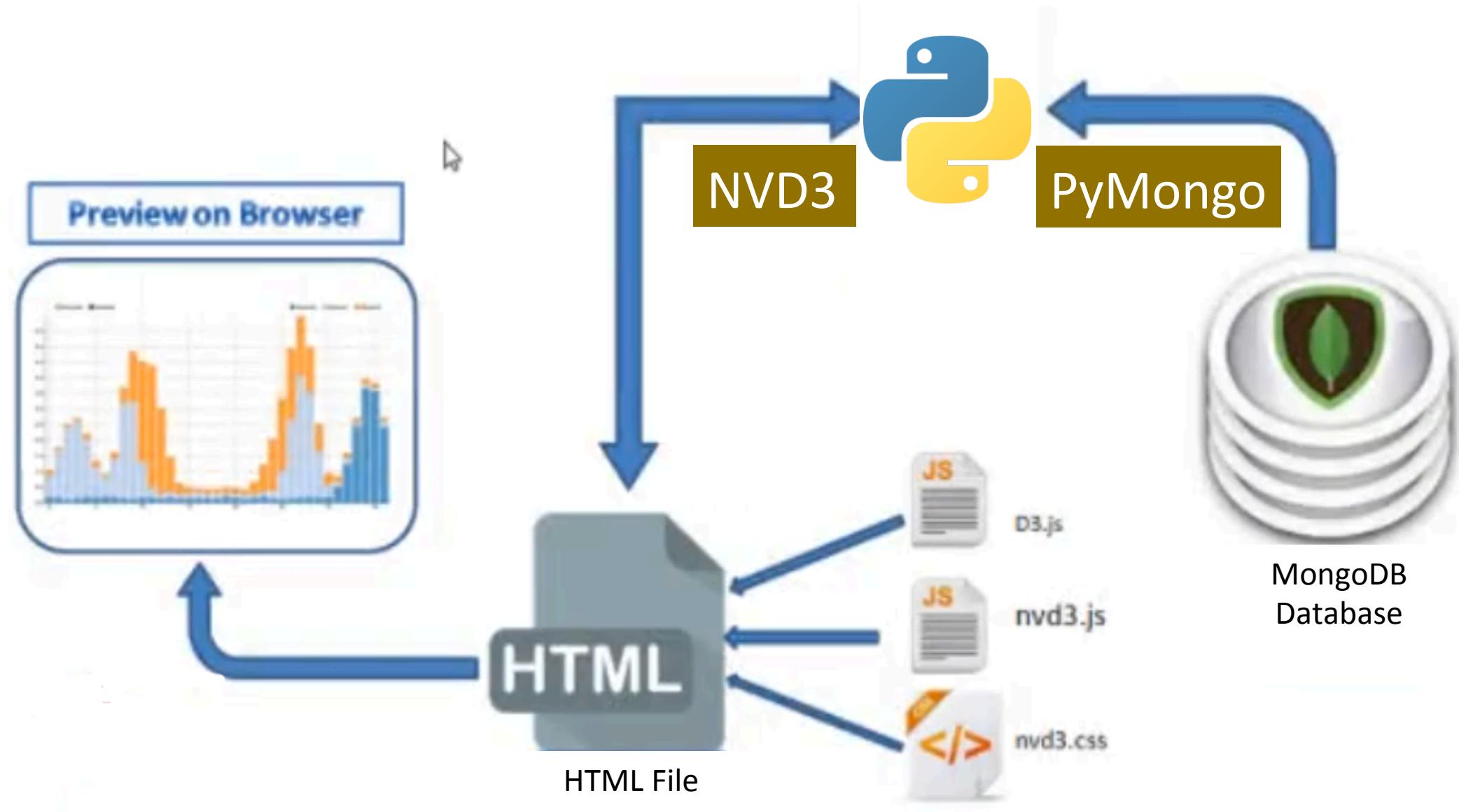


NVD3

- NVD3 is a **JavaScript** visualization library
- Internally it uses **d3.js** which is the one of best customizable visualization library
- This is a very young collection of components, with the goal of keeping these components very customizable



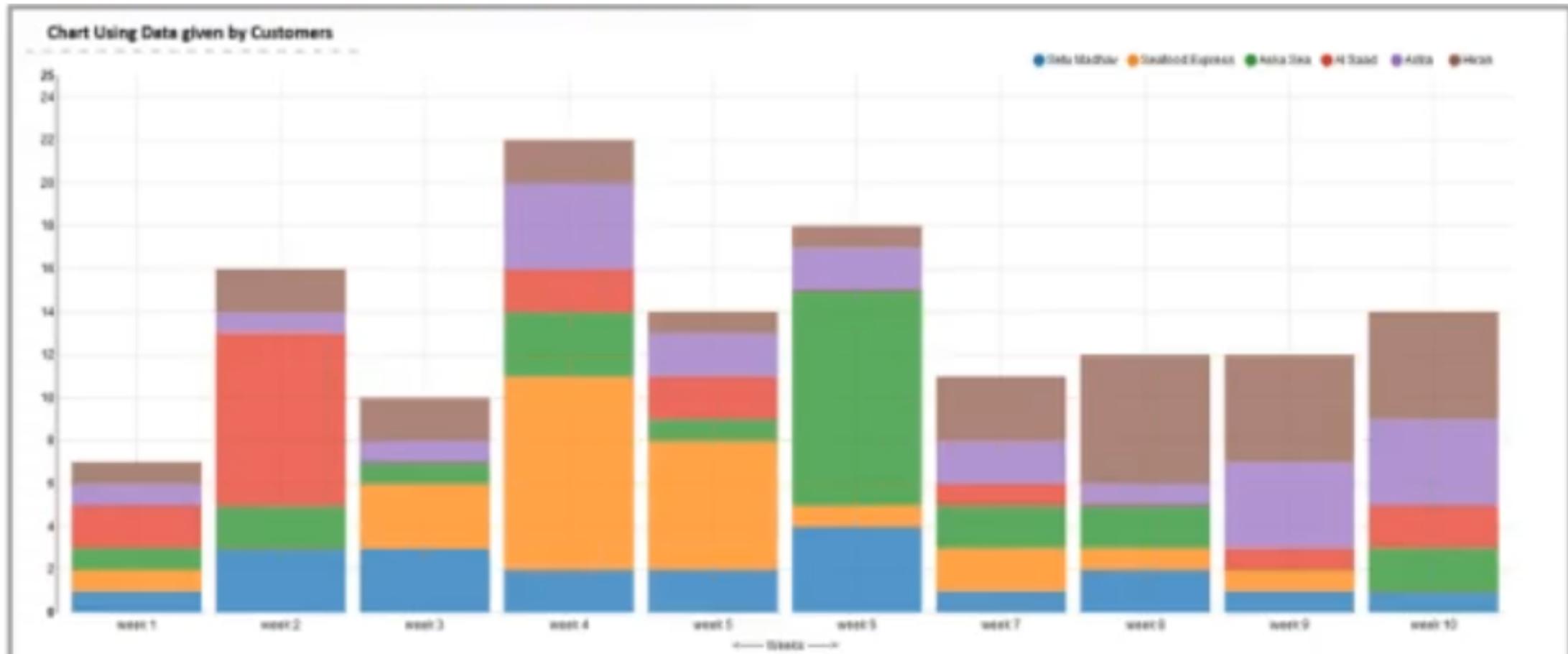
Flow Diagram





Business Scenario

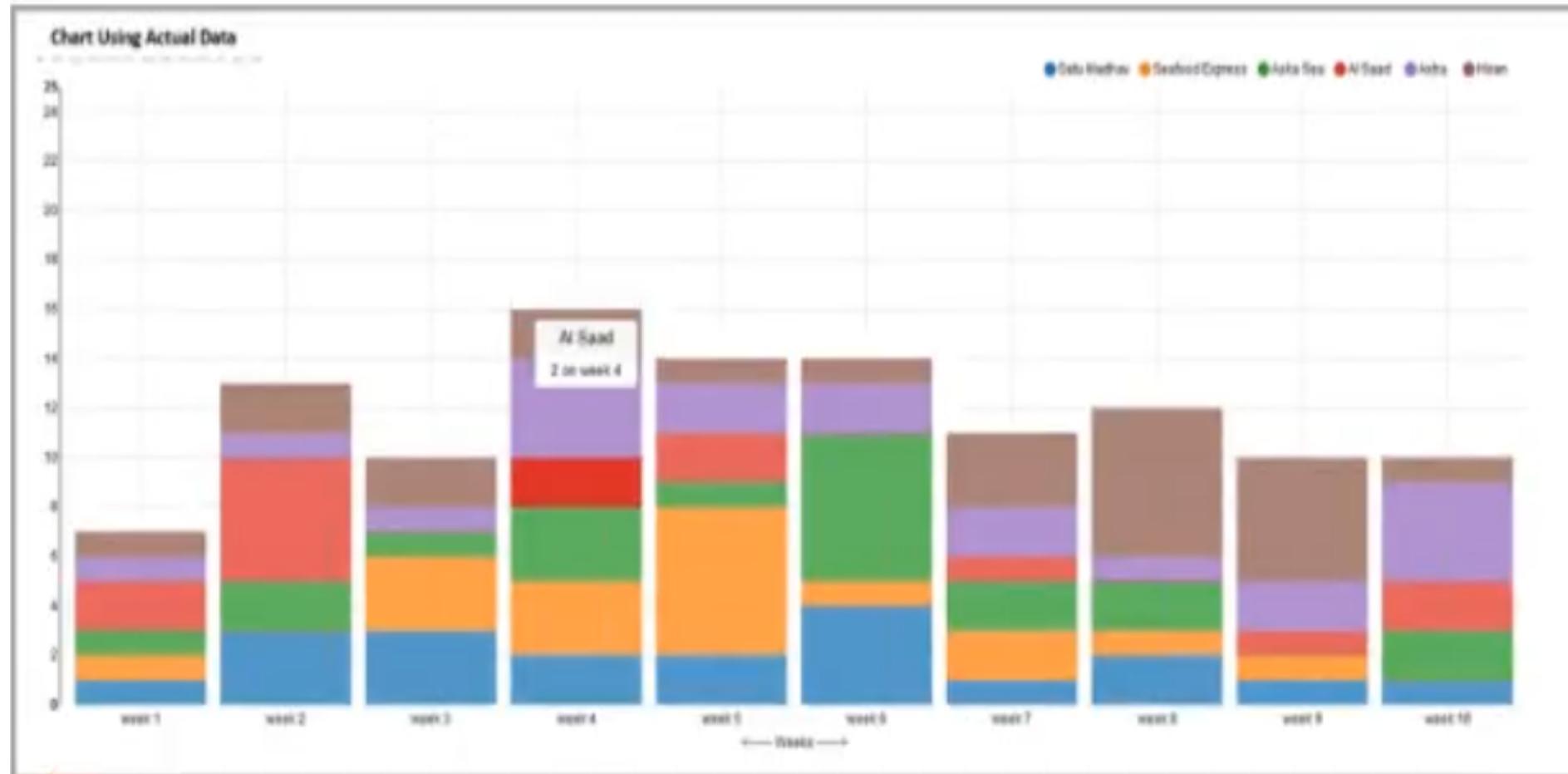
Below Chart describes the number of claims received in last 10 weeks





Business Scenario

Below chart describes the genuine claims which we got from the Cloud platform



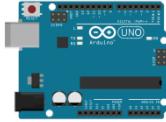
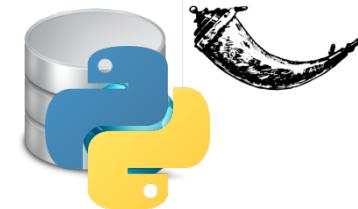


Business Scenario

Below chart describes the fault claims



Practical Flow for Best Cold Chain Problem

Sensors	Nodes	Gateway	Web Services	Data Stores	Analytics	Visualization
 Read, Calibrate and Display data from sensors   	 Develop code for reading SMS through Arduino and trigger the Alarm.	 Send the data to cloud using GPRS	 Create a REST based Web Service to send truck sensor data on the cloud    Learn how to create cloud based programs	 Write the program to store the truck sensor data in NoSql Database.	 Analyze the data to solve business problem.	 Generate a neat report to visualize the data for management.  View and Control from Mobile App  