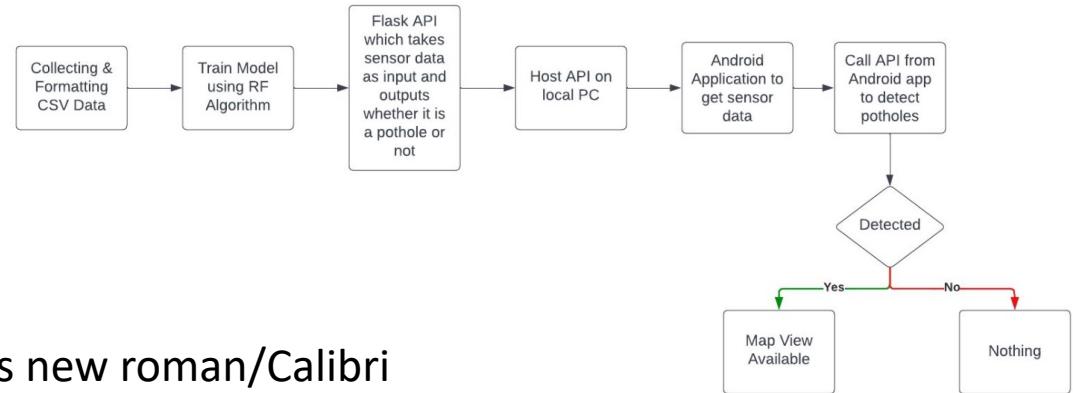


Projects

- Group formation
 - Due Oct 31st Via Google Drive
 - <https://docs.google.com/spreadsheets/d/10-RV4V60ajTHDWobIVtrf2Ck6Q3-pEn-E57OE36zuil/edit?usp=sharing>
- Project Discussion Day (zoom discussion)
 - Oct 27th, Schedule via Google Drive
 - https://docs.google.com/spreadsheets/d/1CoKyYJnABQ-wF8aoHeqmxPrj0NR6PbMMY8hIC3w_zfw/edit?usp=sharing
 - Zoom link: <https://uml.zoom.us/j/94459888925>
- Project Proposal Submission and Proposal Presentation
 - Nov 7th , In Class

Project Proposal Submission



- 1-2 Pages PDF, 10 pt, times new roman/Calibri
- Title
- Names of the students and email addresses
- Introduction: Introduce the problem
- Tentative device requirements
- Tentative Overview
 - Overview diagram/Flow diagram, Must use <https://draw.io/>
 - Hardware Design Plan
 - Software Design Plan
 - Machine learning analysis Plan
- Tentative Final project outcome and future works

Project Proposal Presentation

- Duration 5 minutes per group/project
- 2 minutes questions/answers/suggestions
- Please incorporate multiple presenters
- Slides
 - Project title, team members (1 slide)
 - Introduction to the problem (1 slide)
 - Devices required/acquired (1 slide)
 - Techniques used/will be used (2-3 slides)
 - Hardware integration
 - Signal processing
 - Machine learning (Must be included)
 - Final demo plan (1 slide)
 - What you are going to show in the final presentation

COMP.5650/4150.201

Internet of Things

Lecture 8: Project Discussions

Instructor
Mohammad Arif UI Alam

List of Instruments

- Google drive view
- <https://drive.google.com/file/d/1d2uHHDjPu6Esh-2-qseKpoQU6Or1E-4u/view?usp=sharing>

Available Technologies

Empatica E4

E4 Sensors



PPG Sensor
Photoplethysmography Sensor - Measures Blood Volume Pulse (BVP), from which heart rate, heart rate variability (HRV), and other cardiovascular features may be derived

3-axis Accelerometer
Captures motion-based activity

Event Mark Button
Tags events and correlate them with physiological signals

EDA Sensor (GSR Sensor)
Electrodermal Activity Sensor
- Used to measure sympathetic nervous system arousal and to derive features related to stress, engagement, and excitement.

Infrared Thermopile
Reads peripheral skin temperature

Internal Real-Time Clock
Temporal resolution up to 0.2 seconds in streaming mode

Android Connectivity, preliminary data collection, preliminary data analysis, ML implementation codes will be provided

Potential Projects Empatica E4

- Dehydration/Hydration status detection
- Mobile phone addiction detection
 - Short form video addiction
- Driving Stress estimation
- Smoking craving detection
- Sleep quality monitoring
- Over-exercise prediction
- Asthma episode prediction
- Opioid Craving Detection
- Alcohol consumption detection
- Stress/Anxiety detection

EEG Sensor (brain signal)

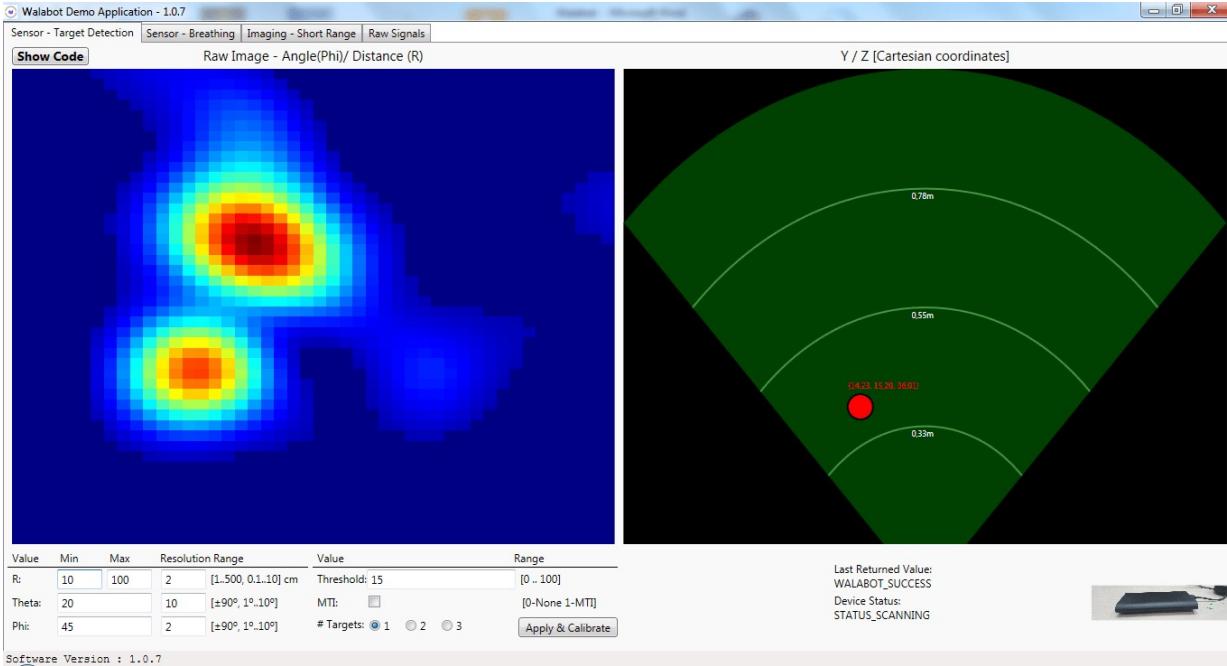
- 4 Different Channels
- Implementation code with Raspberry PI, Arduino and Android will be provided



Potential Projects

- Music Recommendation
- Robotic Arm Control using Brain Signal
- Emotion Detection
 - Emotional videos playing and differentiate them
- Video rating
 - Watch a video and automatically rate (like or not like) the video

Radar



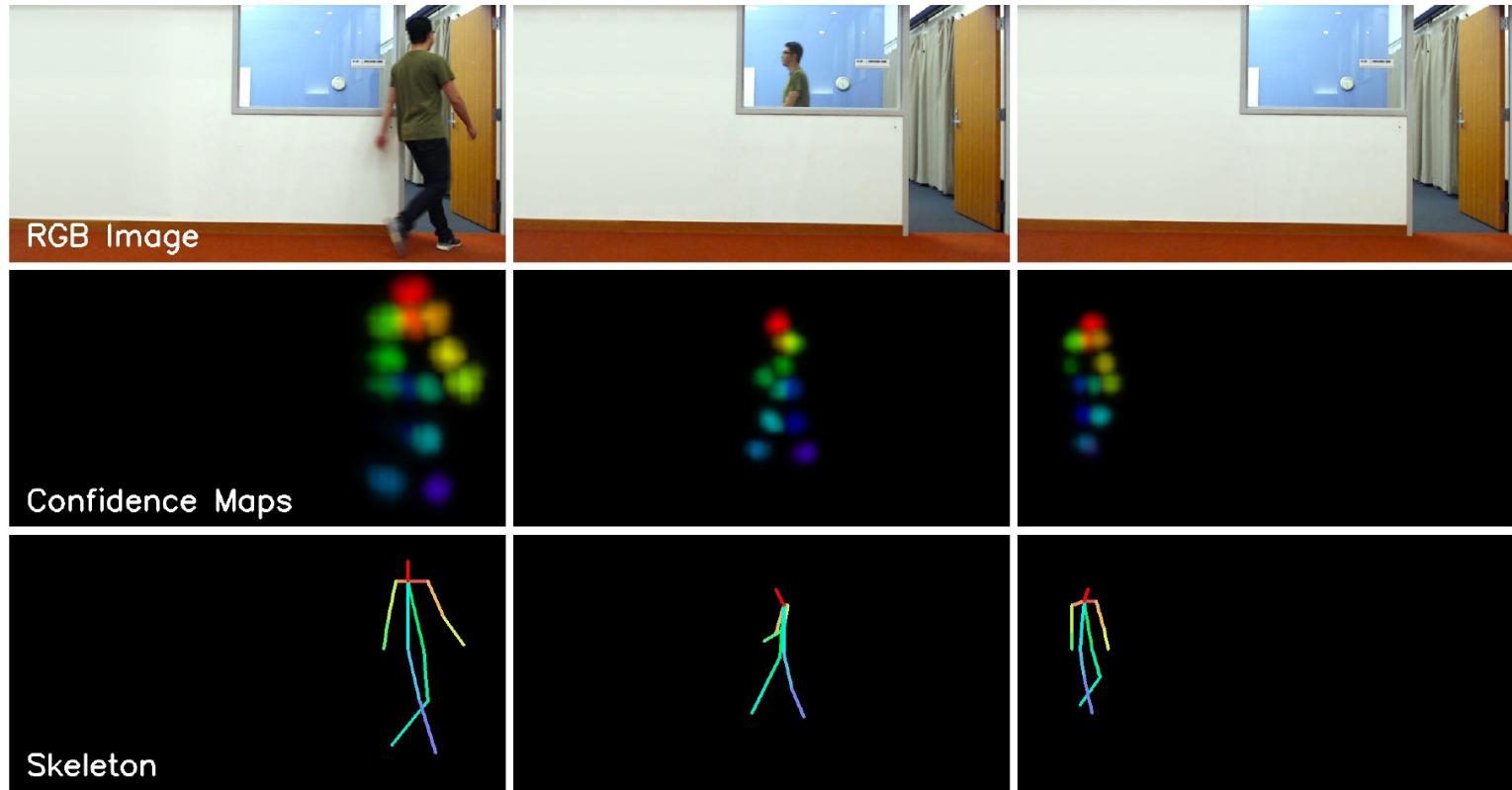
- Raspberry PI implementation codes, basic data collection code and basic ML codes will be provided



Potential Projects

- People count in Hospital Waiting Room/Crowd
- Fall Detection
- Gate detection (distance between your two step/ stride size)
- Activity Recognition (sitting, standing, lying)
- Sleep quality detection

High Definition Radar (One unit)



Potential Projects

- Eavesdropping, detecting activities of your neighbor through wall (privacy violation)
- Listening speech through wall (Privacy violation)

Palmsens Sensit Smart

Potentiostat Sensor

- A potentiostat is an electronic instrument that controls the voltage difference between a Working Electrode and a Reference Electrode.
- Both electrodes are contained in an electrochemical cell.
- The potentiostat implements this control by injecting current into the cell through an Auxiliary, or Counter, electrode.



Water Contaminants Detection



Example Projects

1. Multiple Sensors (Lidar, Radar, RF, Infrared and Camera) fusion for activity recognition (dataset will be created)
2. Eye Tracker and Brain Signal (EEG) signal analytics for human subconscious attention signal processing towards explainable AI (few dataset have already been created)
3. Deep MRI processing for Alzheimer's disease detection (data will be provided)
4. Electronic Health Record analysis towards Disease cause inference (data will be provided)
5. AI Fairness towards older adults' cognitive health assessment (data will be provided)
6. Improving Breast Cancer detection from mammography and ultrasound images (data will be provided)
7. Smart home appliance energy consumption data based health anomaly detection (dataset will be provided)
8. Galvanik Skin Response sensor processing for fear, stress, depression detection (data will be provided, but you can also collect some data, in that case, device will be provided with proper training how to collect data)
9. Sleep assessment using Sleep Pad (Withings Sleeping Pad) that detects heart rate, breathing rate, movements and snoring (data needs to be collected, device will be provided)

Source of Projects

- <https://www.hackster.io/>

COMP.5650/4150.201

Internet of Things

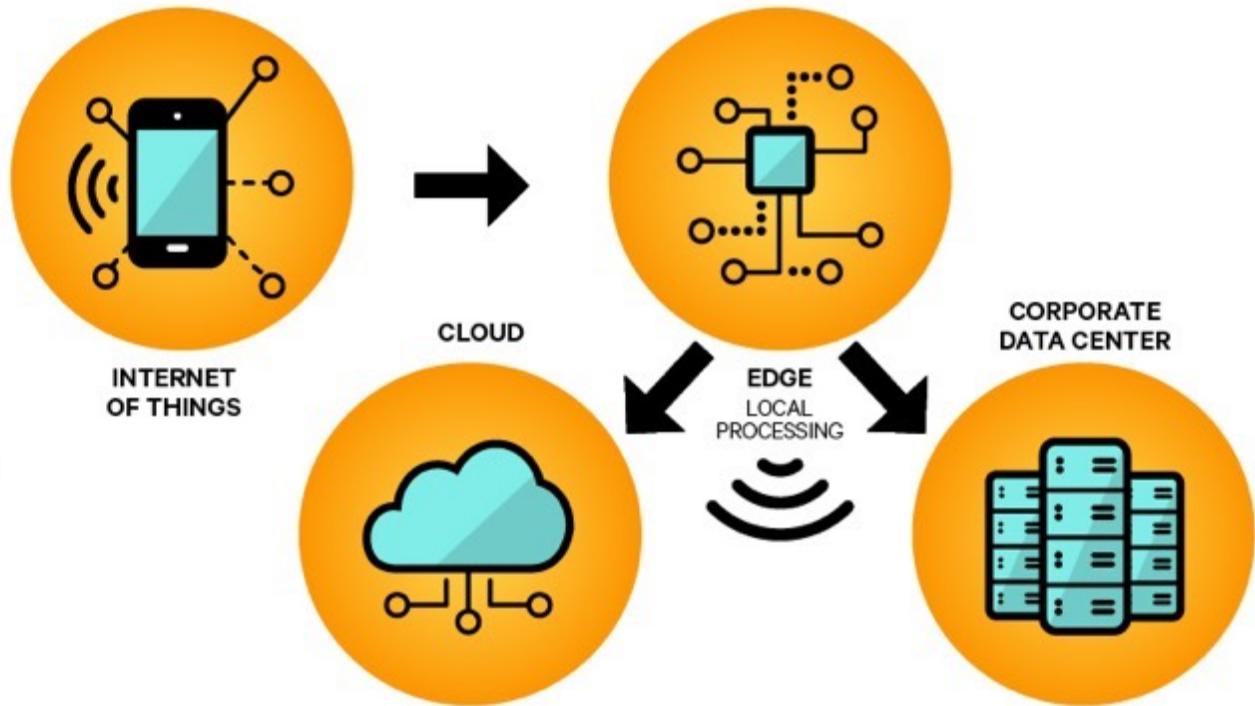
Lecture 8.1: Edge and Cloud Computing

Instructor

Mohammad Arif UI Alam

HOW EDGE COMPUTING WORKS

Edge computing allows data from Internet of Things devices to be analyzed at the edge of the network before being sent to a data center or the cloud.





Cloud Computing

[klaud kəm-'pyüt-in]

The delivery of different services through the Internet which includes tools and applications like data storage, servers, databases, networking, and software.

When to Use Edge Computing vs Cloud Computing?

Cloud Computing

Non-time-sensitive data processing

Reliable internet connection

Dynamic workloads

Data in cloud storage

Edge Computing

Real-time data processing

Remote locations with limited or no internet connectivity

Large datasets that are too costly to send to the cloud

Highly sensitive data and strict data laws



EDGE IMPULSE

A world-class team of industry leaders and investors, creating the platform and ecosystem for embedded ML, enabling developers and enterprises from research to production.



Zach Shelby
Founder, CEO



Jan Jongboom
Founder, CTO

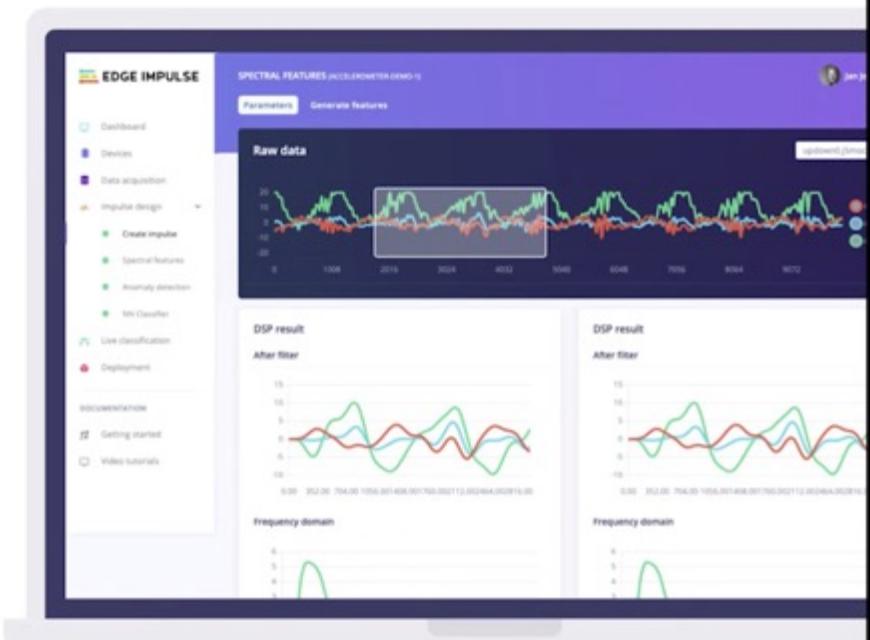
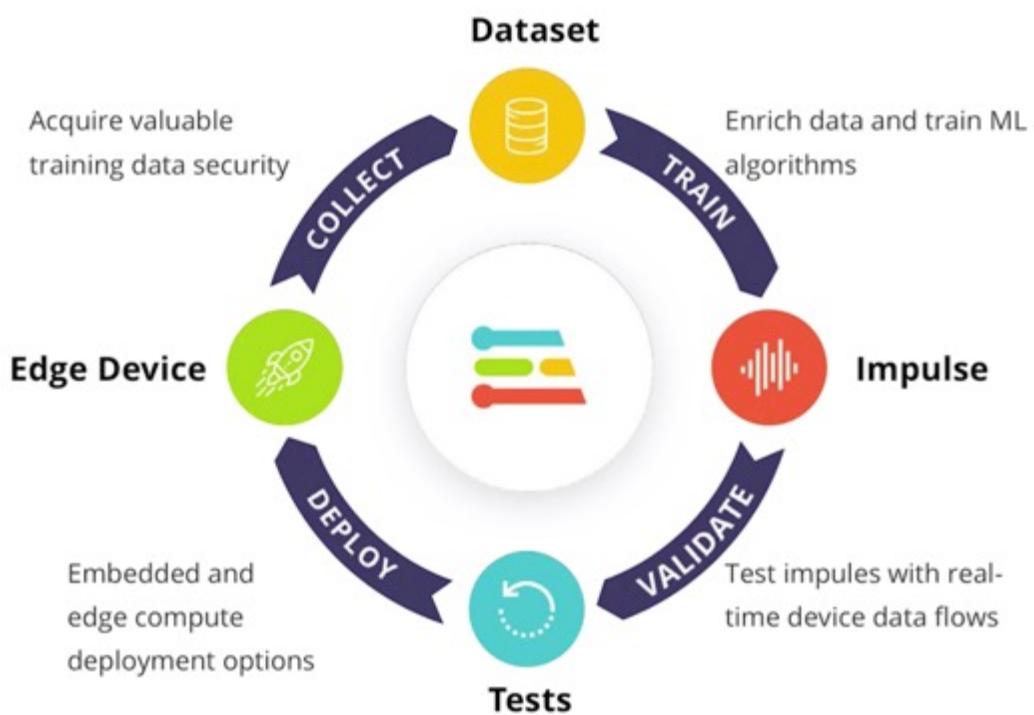


"2.5 billion devices and
3 million developers will
use embedded ML by
2030"



Edge Impulse

The ML development platform



Learn more at <http://edgeimpulse.com>

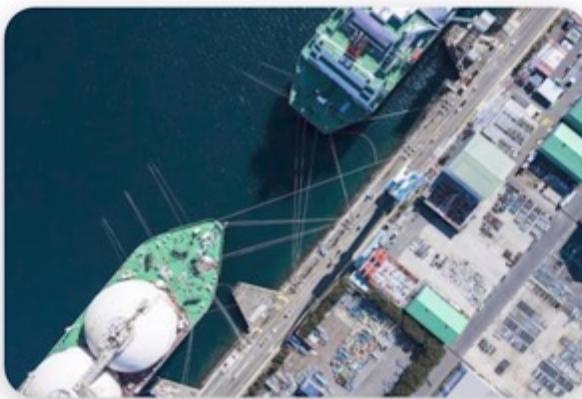


Industrial



- Predictive maintenance
- Process and logistical optimization
- Remote asset management

Enterprise



- Asset tracking and utilization analysis
- Facility usage optimization
- Metering and infrastructure monitoring

Wearables

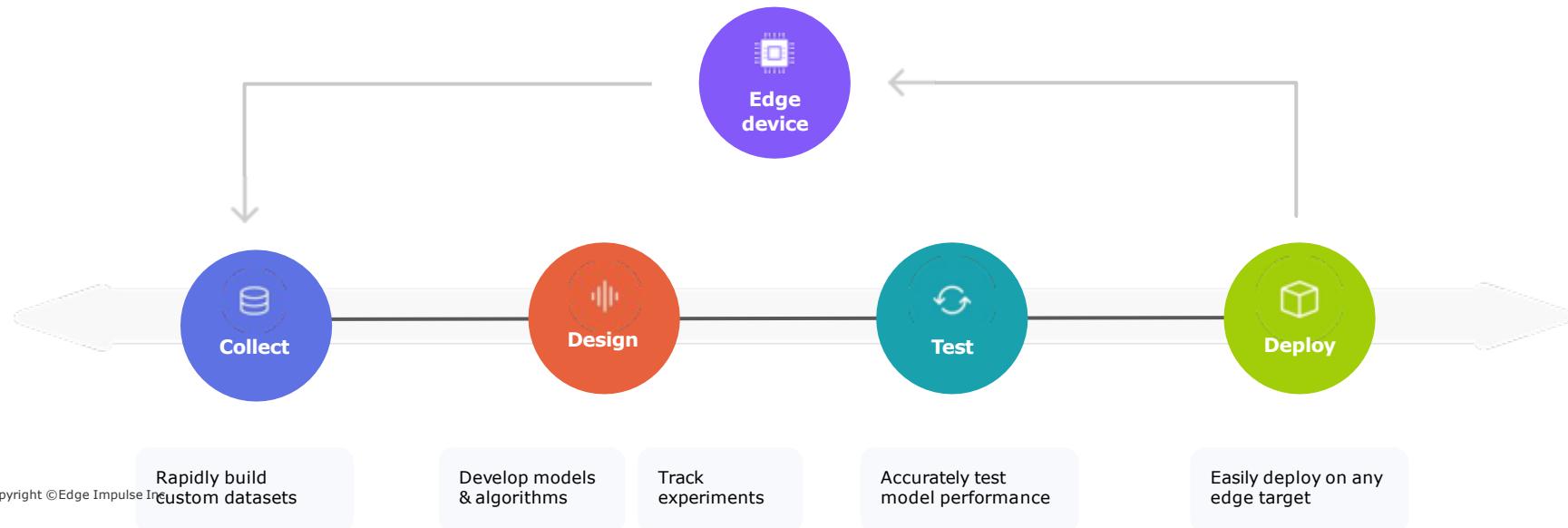


- Health monitoring
- Worker safety monitoring
- Occupancy and foot traffic analytics



Develop edge ML applications [with Edge Impulse](#)

The infrastructure and integrations your data science and ML teams need.



Copyright ©Edge Impulse Inc.

Activity Recognition in Smart Phone

The screenshot shows the Edge Impulse Studio interface, specifically the 'Devices' section. The left sidebar contains navigation links: Dashboard, Devices (selected), Data acquisition, Impulse design, Create impulse, Spectral features, NN Classifier, Anomaly detection, Retrain model, Live classification, and Model testing. The main content area has a purple header bar with the text 'DEVICES (MOBILE CLIENT)' and a user profile for 'Jan Jongboom (demo)'. Below this, a section titled 'Your devices' displays a message: 'These are devices that are connected to the Edge Impulse remote management API, or have posted data to the ingestion SDK.' It also states 'No devices connected yet.' and includes a green button labeled 'Learn how to connect a new device'. At the bottom, a copyright notice reads '© 2020 EdgeImpulse Inc. All rights reserved'.

Activity Recognition in Smart Phone

The screenshot shows the Edge Impulse studio interface. On the left is a sidebar with various icons and labels: Dashboard, Devices, Data acquisition, Impulse design, Connect (partially visible), Sensors (partially visible), Noise (partially visible), Acceleration (partially visible), Retrain, Live class, Model, and Deploy. The main area has a purple header bar with the text "DEVICES (MOBILE CLIENT)" and a user profile for "Jan Jongboom (demo)". Below this, a white card titled "Your devices" contains the message "These are devices that are connected to the Edge Impulse remote management API, or have posted data to the ingestion SDK." and "No devices connected yet." To the right of this card is a button "+ Connect a new device". A modal window is open in the foreground, titled "Connect a new device". It contains the text "You can collect data from development boards, from your own devices, or by uploading a prebuilt data set." Below this are four options: "Browse development boards" (with a small icon of a development board), "Use your mobile phone" (with a large circular icon containing a smartphone icon and a hand cursor), "Porting guide" (with a small icon of two overlapping documents), and "Import prebuilt data sets" (with a small icon of a database). The URL in the browser bar is <https://studio.edgeimpulse.com/studio/1566/devices>.

Activity Recognition in Smart Phone

EDGE IMPULSE

DEVICES (MOBILE CLIENT)

Jan Jongboom (demo)

Your devices

+ Connect a new device

These are devices that are connected to the Edge Impulse remote management API, or have posted data to the ingestion SDK.

Connect a new device

You can collect data from any smartphone. From your smartphone go to this URL, or scan the QR code below.



GETTING STARTED

Data collection

Activity Recognition in Smart Phone

EDGE IMPULSE

Dashboard

Devices

Data acquisition

Impulse design

- Create impulse
- Spectral features
- NN Classifier
- Anomaly detection

Retrain model

Live classification

Model testing

Deployment

DEVICES (MOBILE CLIENT)

Your devices

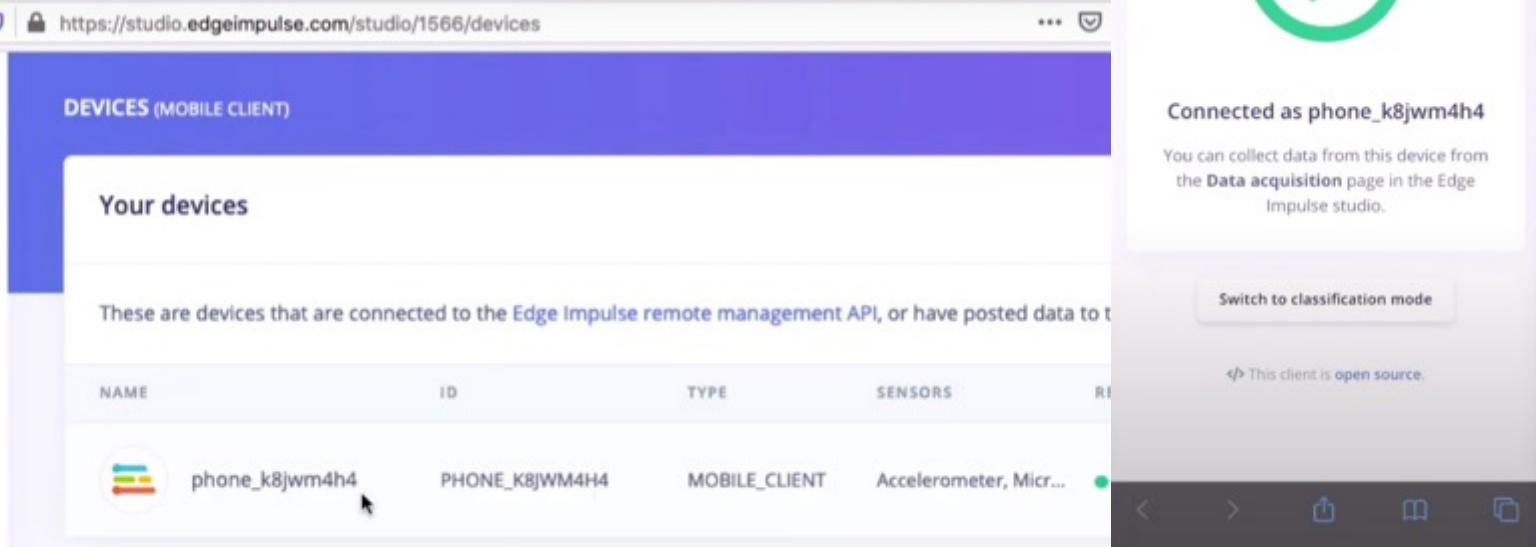
These are devices that are connected to the Edge Impulse remote management API, or have posted data to the Edge Impulse cloud.

NAME	ID	TYPE	SENSORS	RECENTLY UPLOADED
phone_k8jwm4h4	PHONE_K8JWM4H4	MOBILE_CLIENT	Accelerometer, Mic...	

Switch to classification mode

</> This client is open source.

© 2020 EdgeImpulse Inc. All rights reserved



Activity Recognition in Smart Phone

The screenshot shows the Edge Impulse Data Acquisition interface for a mobile client. The top navigation bar includes back, forward, refresh, and search icons, along with the URL <https://studio.edgeimpulse.com/studio/1566/acquisition/training>. The top right corner shows a user profile for "Jan Jongboom (demo)".

The main header "DATA ACQUISITION (MOBILE CLIENT)" is displayed above two tabs: "Training data" (selected) and "Test data". A prominent message box says: "Did you know? You can add data from any device through the Ingestion API." with a close button.

The left sidebar lists various features: Dashboard, Devices, Data acquisition, Impulse design, Create impulse, Spectral features, NN Classifier, Anomaly detection, Retrain model, Live classification, Model testing, and Deployment. Below this is a "GETTING STARTED" section.

The central area displays data collection statistics: "DATA COLLECTED 0s" and "LABELS 0". A "Record new data" section contains fields for "Device" (set to "phone_k8jwm4h4"), "Label" (set to "updown"), "Sample length (ms.)" (set to 5000), "Sensor" (set to "Accelerometer"), and "Frequency" (set to 62.5Hz). A button at the bottom right says "Sampling... X3s left".

The "Collected data" section shows a message: "No data collected yet" and a "Learn how to connect a new device" button.

Activity Recognition in Smart Phone

The screenshot shows the Edge Impulse Studio interface for data acquisition. On the left, a sidebar lists various features: Dashboard, Devices, Data acquisition, Impulse design, Create impulse, Spectral features, NN Classifier, Anomaly detection, Retrain model, Live classification, Model testing, and Deployment. Below this is a 'GETTING STARTED' section.

The main area is titled 'DATA ACQUISITION' and shows 'Training data'. A large green circle with a white checkmark is displayed, indicating data collection status. The text 'Connected as phone_k8jwm4h4' is shown below it. A button labeled 'Switch to classification mode' is highlighted with a red box.

A central window titled 'Data collection' shows '0s' collected data. To the right, a 'Record new data' section is visible, showing settings for 'Device' (set to 'phone_k8jwm4h4'), 'Label' ('updown'), 'Sample length (ms.)' (set to 5000), 'Sensor' ('Accelerometer'), 'Frequency' (set to 62.5Hz), and a 'Sampling...' button with 'X3s left'.

The top bar shows the URL <https://studio.edgeimpulse.com/studio/1566/acquisition/training>. The top right corner shows the user profile 'Jan Jongboom (demo)'.

Activity Recognition in Smart Phone

The screenshot shows the Edge Impulse Studio interface with the following components:

- Left Sidebar:** Lists various features: Dashboard, Devices, Data acquisition, Impulse design, Create impulse, Spectral features, NN Classifier, Anomaly detection, Retrain model, Live classification, Model testing, Deployment, and Getting Started.
- Central Panel (Data Acquisition):** A blue header says "DATA ACQUISITION". It shows "Training data" and a "Did you..." button. Below is a circular progress bar labeled "Sampling..." with a "2s" center. To its left, it says "DATA COLLECTED 0s" and "Collected data". At the bottom is a table with columns IDLE, SNAKE, UPDOWN, WAVE, ANO..., and values 1, 0.99, 0.01, 0.00, 0.00, -0.06. A button "Switch to data collection mode" is at the bottom right.
- Central Panel (Classifier):** Shows a "Classifier" section with a purple header. It displays "Ingestion API." and a "Record new data" section.
- Right Panel (Smartphone Interface):** Shows a smartphone screen with the URL "smartphone.edgeimpulse.com". The phone's status bar shows "10:41 Camera". The phone's home screen has a purple header with the user "Jan Jongboom (demo)" and a circular icon.
- Bottom Right:** A dark blue button says "Sampling... X3s left".

Edge Impulse for Arduino and Raspberry PI

- Arduino
 - <https://docs.edgeimpulse.com/docs/development-platforms/officially-supported-mcu-targets/arduino-nano-33-ble-sense>
- Raspberry PI
 - <https://docs.edgeimpulse.com/docs/development-platforms/officially-supported-cpu-gpu-targets/raspberry-pi-4>

- Edge Impulse is for High School Students
- We are computer scientist



Cloud Computing

- Cloud computing provides **shared pool of configurable computing resource** to end users **on demand**
- Three service models
 - ❑ **IaaS (Infrastructure as a Service):** virtual machines, storage, network ...



- ❑ **PaaS (Platform as a Service):** execution runtime, middleware, web server, database, development tool ...



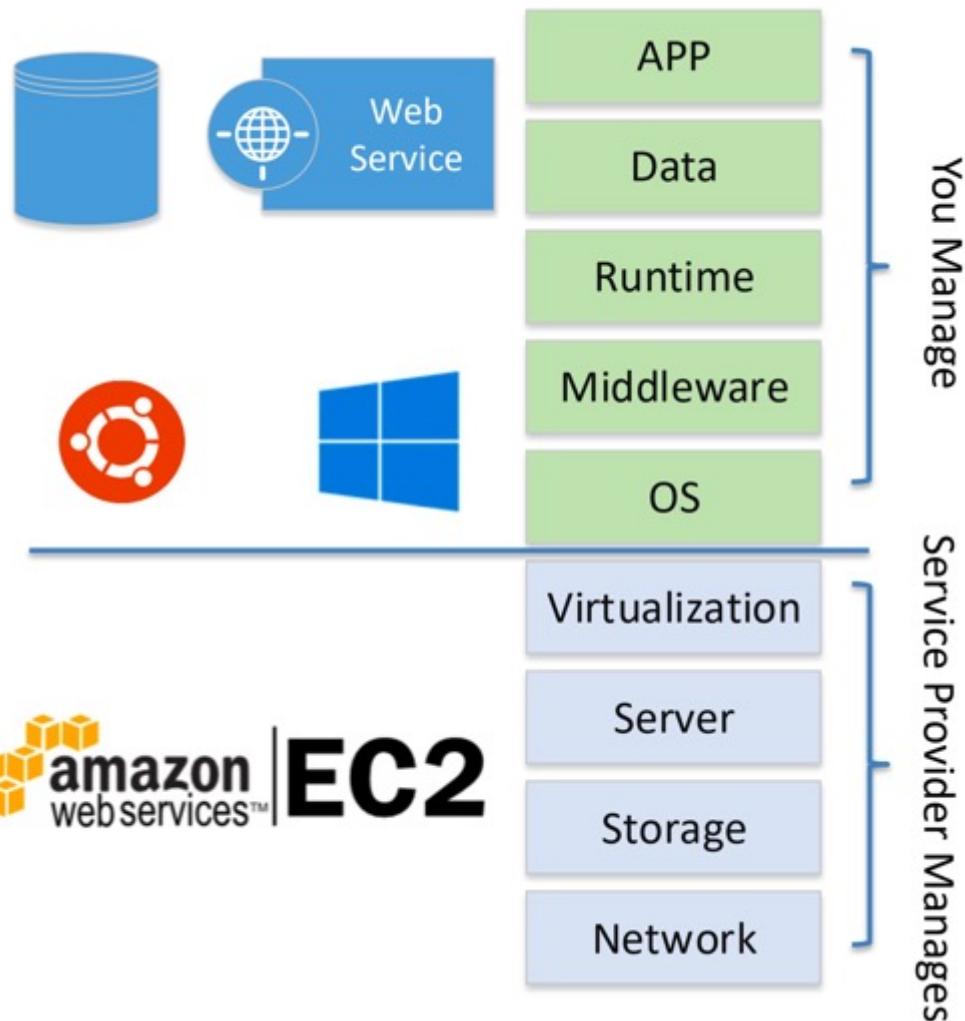
- ❑ **SaaS (Software as a Service):** email, virtual desktop, games ...



Infrastructure as a Service (IaaS)

➤ IaaS

- ❑ "physical server box"
- ❑ Virtual Machine
 - Memory
 - Storage
 - CPU
 - Network



➤ Example

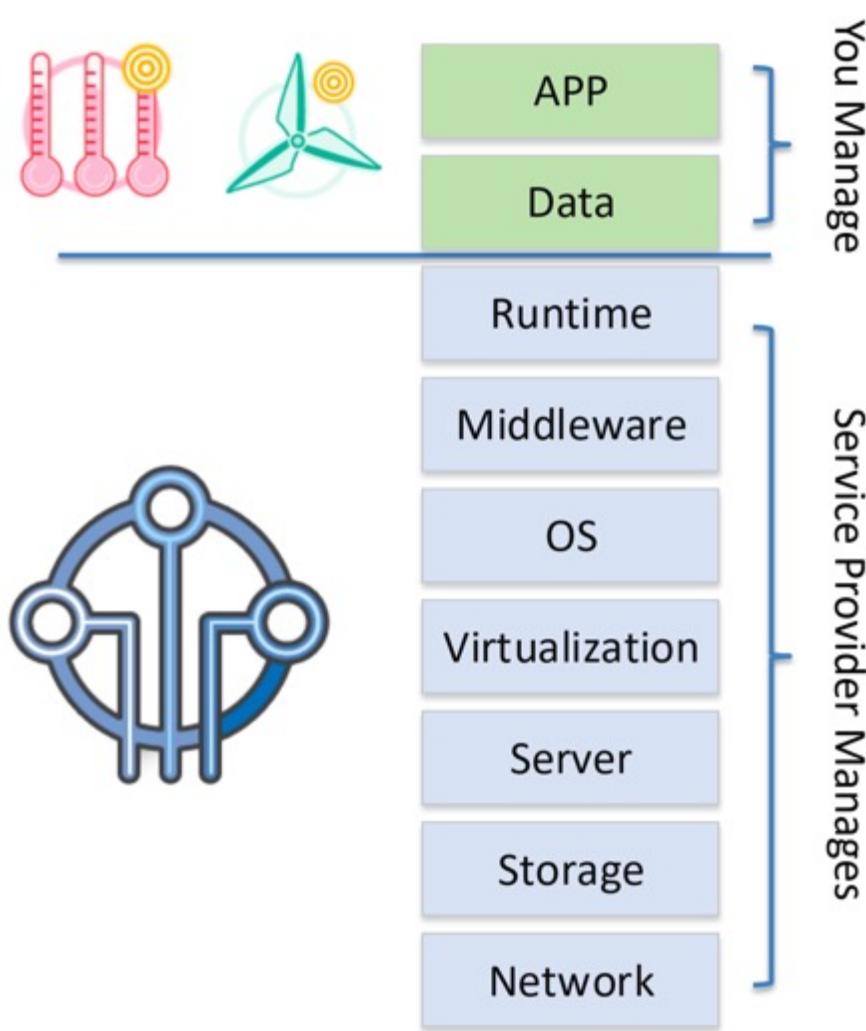
- ❑ AWS EC2
- ❑ AWS EFS

➤ Use case

- ❑ Build up your VM cluster

Platform as a Service (PaaS)

- PaaS
 - ❑ You get a framework
 - ❑ Host Application
 - ❑ Tools
- Example
 - ❑ AWS IoT
- Use case
 - ❑ Build up you're smart A/C controller



PaaS Example: AWS IoT



The essence is **MESSAGING MIDDLEWARE**
Send messages between **sensors** and **servers**...

Software as a Service (SaaS)

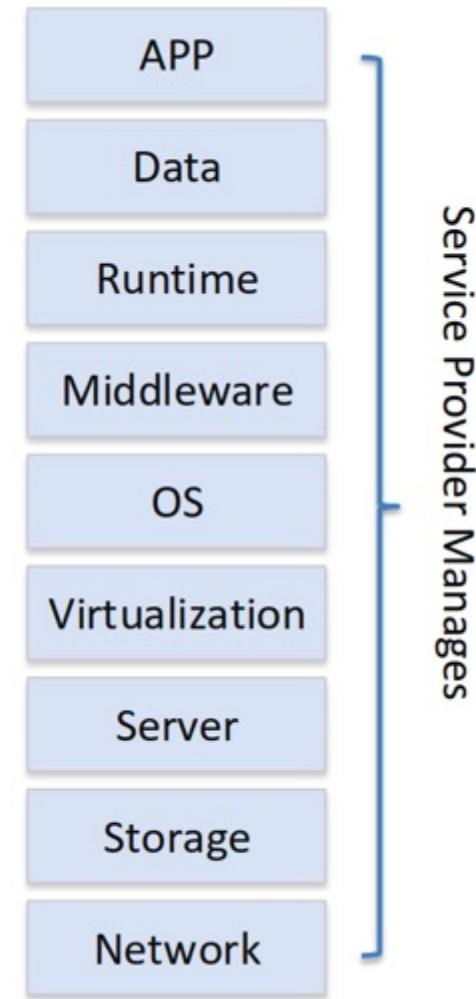
➤ SaaS

- ❑ You get a whole solution

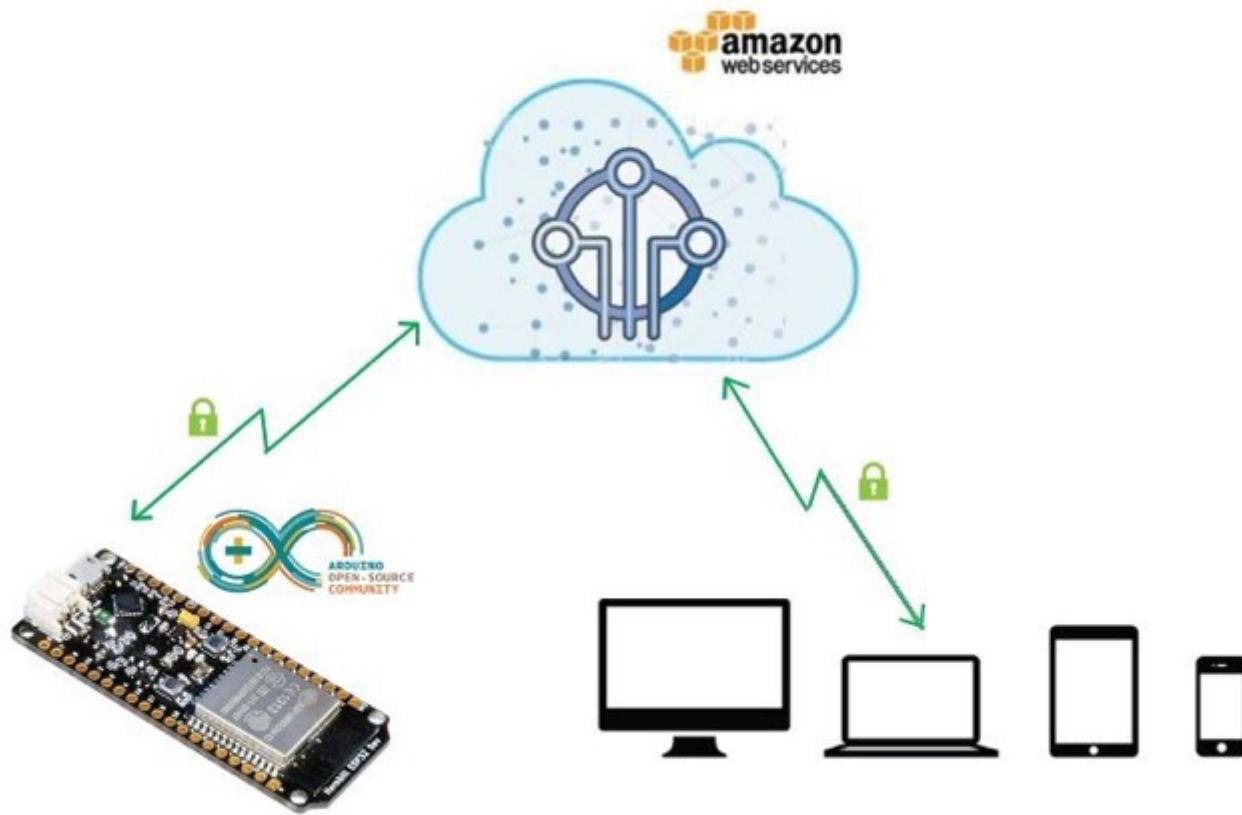


➤ Example

- ❑ Gmail
- ❑ Dropbox
- ❑ Office365

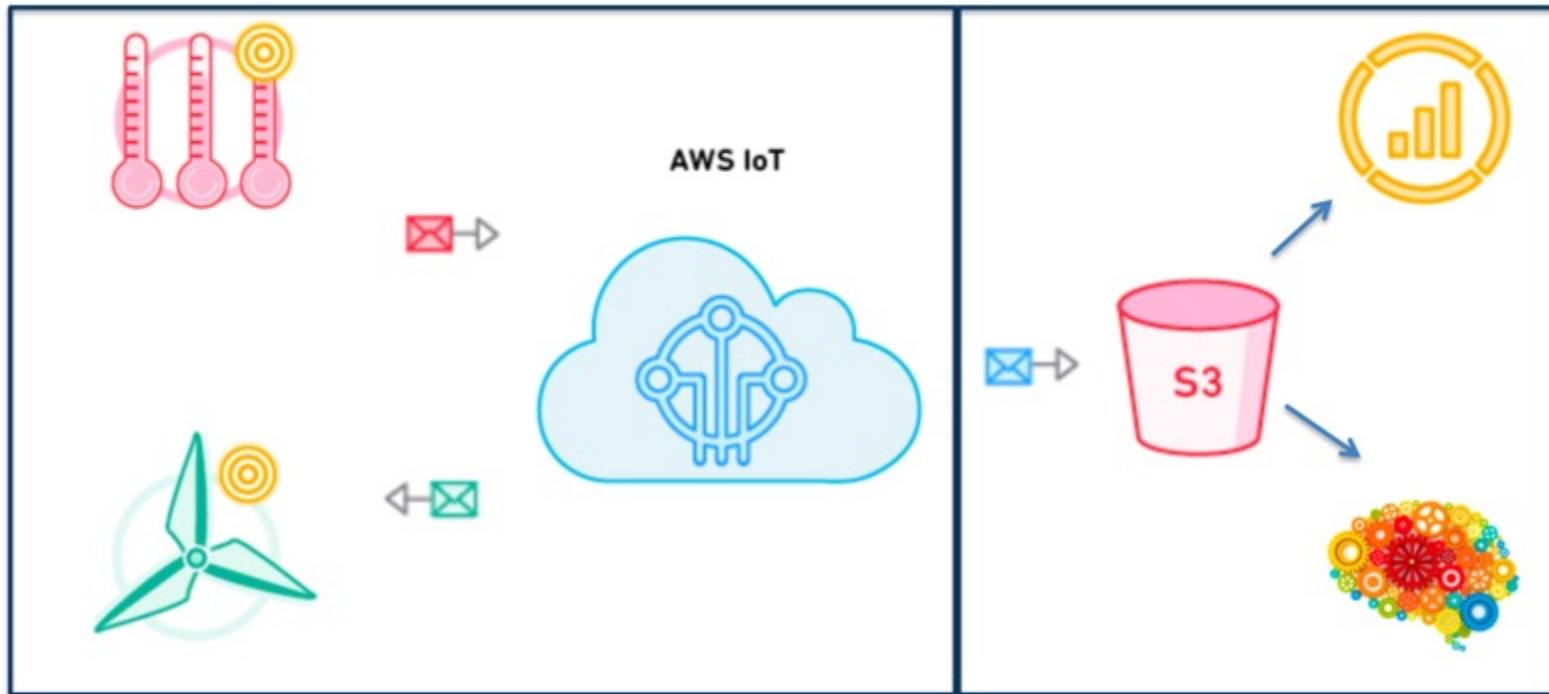


Amazon Web Service (AWS) for IoT Core



https://exploreembedded.com/wiki/Secure_IOT_with_AWS_and_Hornbill_ESP32

AWS IoT

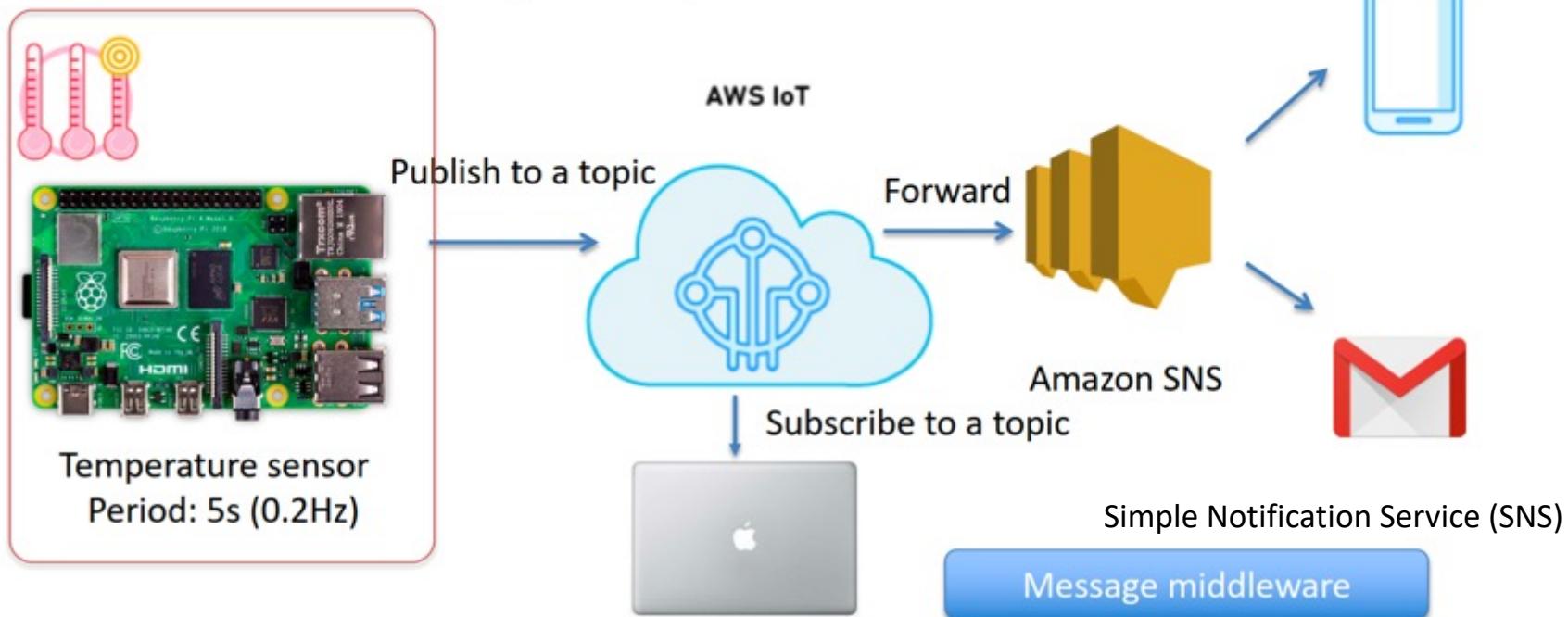


United: Connect + Communication
Stated: "Thing Shadow"

Smart: Other Cloud Service
Data Storage
Machine Learning

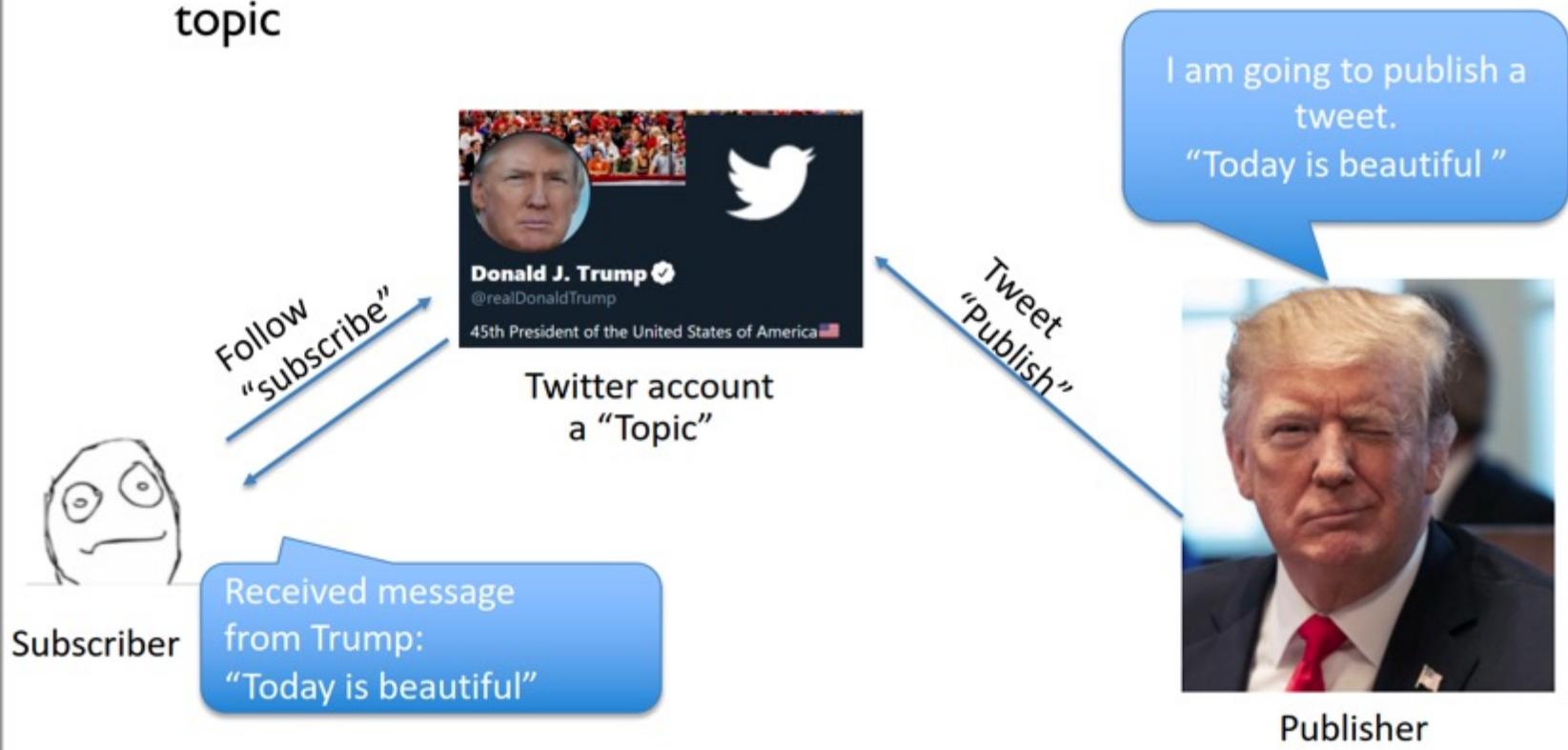
Tutorial: Hello AWS IoT!

- Key concepts:
 - ❑ Publisher (e.g. Sensor), Subscriber (e.g. Server), **Topic**
 - **Topic is used to identify the message.**
- Not a traditional “peer-to-peer” communication.

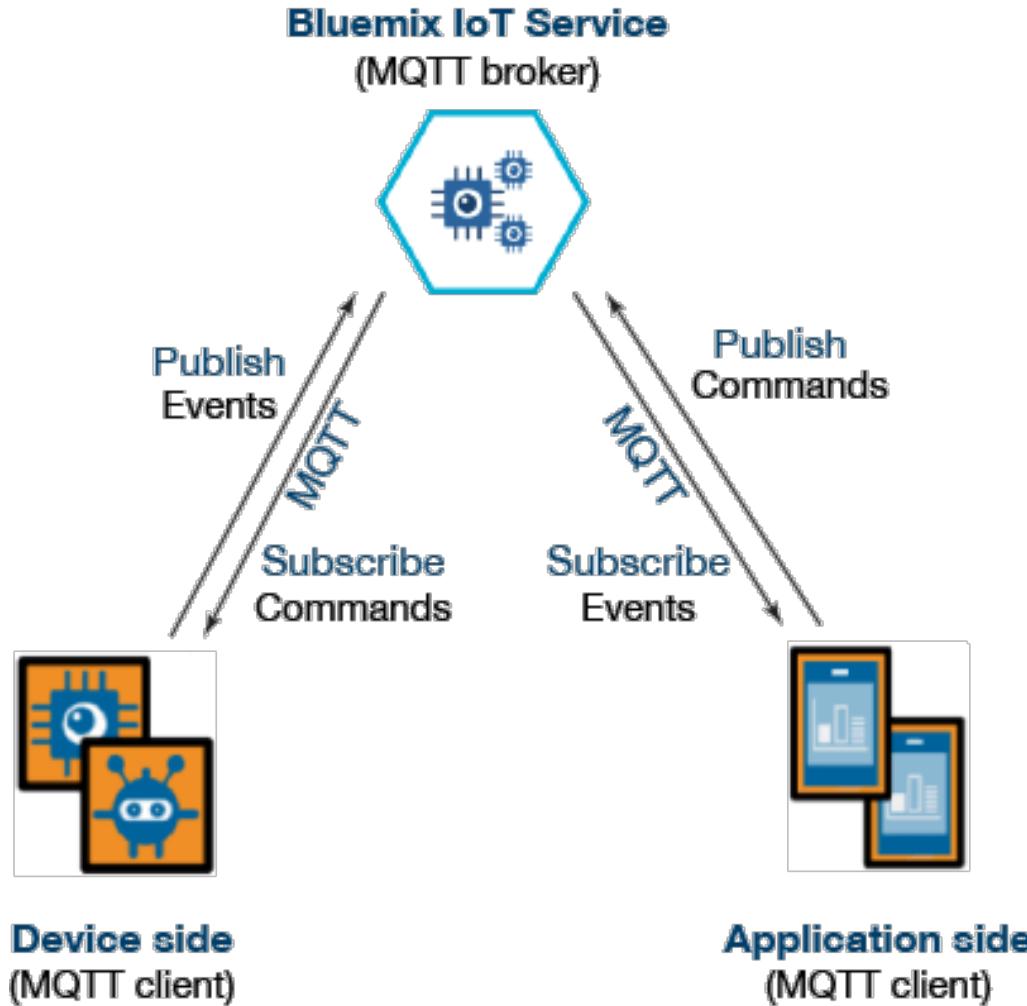


Basic Interact: Subscribe/Publish

- You can define your own Topic (Twitter Account)
- Subscriber can receive the message you published to your topic



MQTT: Message Queuing Telemetry Transport

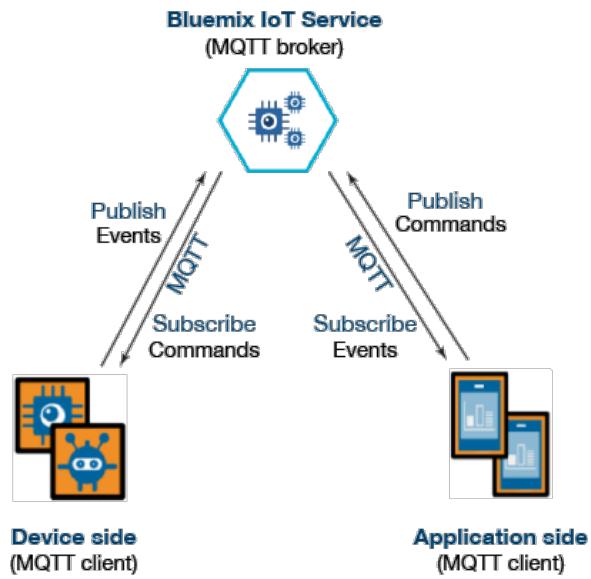


MQTT

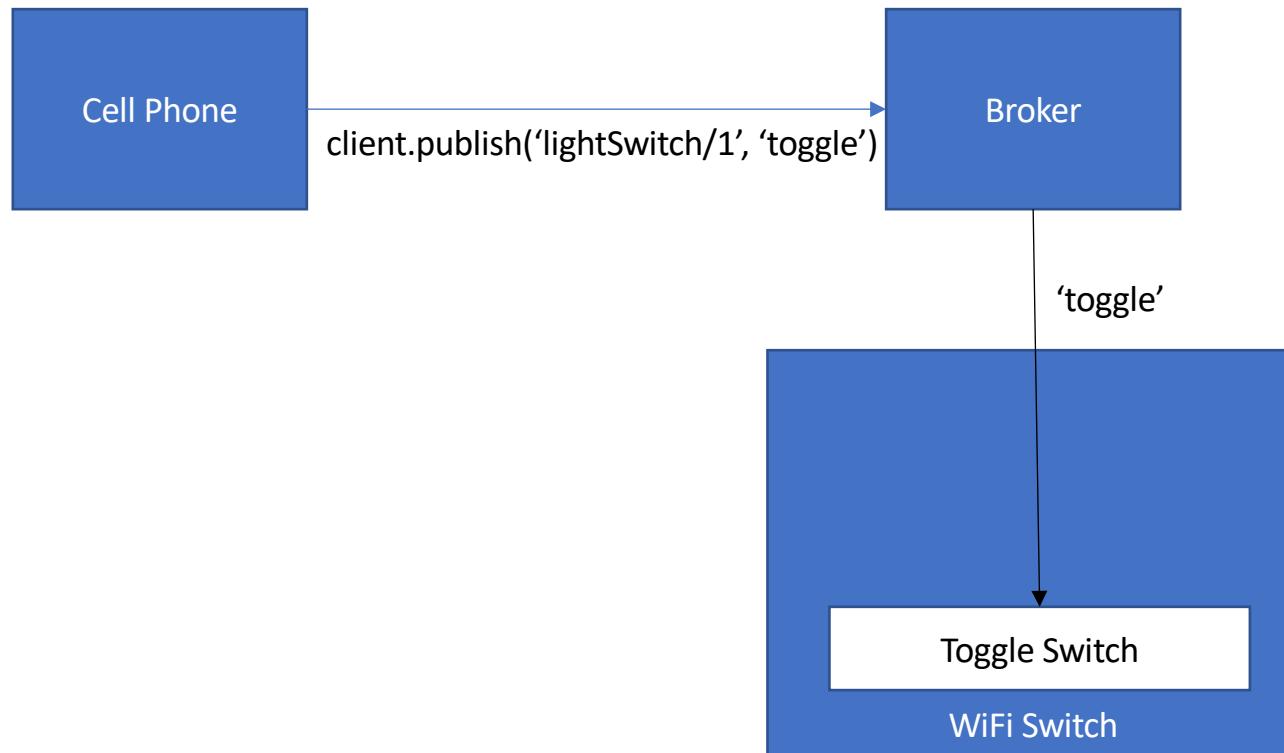
- MQTT was invented by Andy Stanford-Clark (IBM) and Arlen Nipper (Arcom, now Cirrus Link) back in 1999, where their use case was to create a protocol for minimal battery loss and minimal bandwidth connecting oil pipelines over satellite connection. They specified the following goals, which the future protocol should have:
 - Simple to implement
 - Provide a Quality of Service Data Delivery
 - Lightweight and Bandwidth Efficient
 - Data Agnostic
 - Continuous Session Awareness

MQTT: Pub/Sub

- Clients connect to a “Broker”
- Clients subscribe to topics eg,
 - `client.subscribe('toggleLight/1')`
 - `client.subscribe('toggleLight/2')`
 - `client.subscribe('toggleLight/3')`
- Clients can publish messages to topics:
 - `client.publish('toggleLight/1', 'toggle');`
 - `client.publish('toggleLight/2', 'toggle');`
- All clients receive all messages published to topics they subscribe to
- **Messages can be anything**
 - Text
 - Images
 - etc



MQTT Hypothetical Light Switch



Apply for \$40 credits for Amazon AWS

- <https://aws.amazon.com/education/awseducate/apply>
- <https://www.youtube.com/watch?v=idf-gGXvlu4>

AWS User Account

Secure | https://www.amazon.com/ap/signin?openid.assoc_handle=aws&openid.return_to=https%3A%2F%2Fsignin.aws.amazon.com%2Foauth%3Fresponse_type%3Dcode%26client_id%26 ☆



Sign In or Create an AWS Account

What is your email (phone for mobile accounts)?

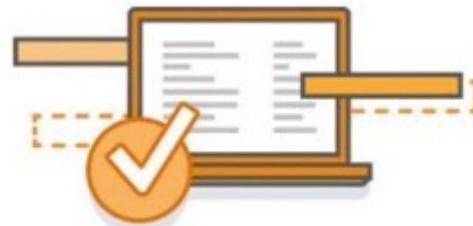
E-mail or mobile number:

I am a new user.

I am a returning user
and my password is:

[Sign in using our secure server](#)

[Forgot your password?](#)



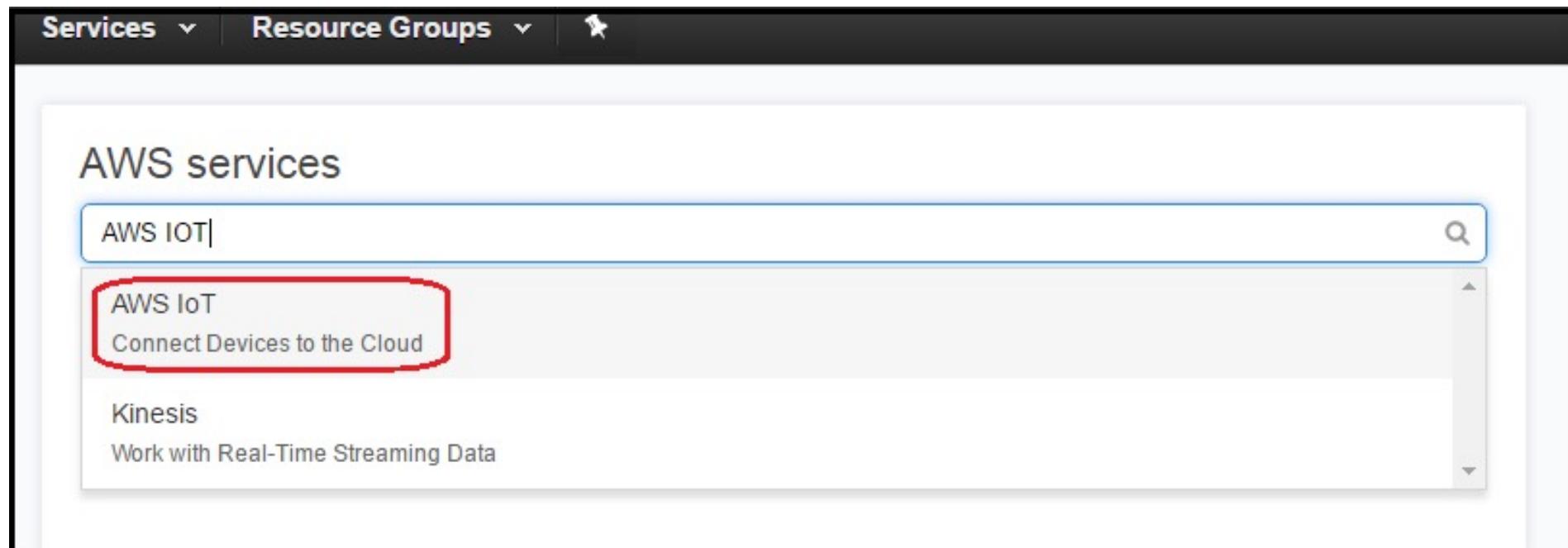
AWS Accounts Include
12 Months of Free Tier Access

Including use of Amazon EC2,
Amazon S3, and Amazon DynamoDB

[Visit aws.amazon.com/free for full offer terms](#)

Creating a new thing

- After Login, click on **Signin To Console** to go to AWS dash board. Now type AWS IOT and Select the AWS IOT (Connect Devices to Cloud) from drop down as shown below





AWS IoT

AWS IoT is a managed cloud platform that lets connected devices - cars, light bulbs, sensor grids, and more - easily and securely interact with cloud applications and other devices.

[Get started](#)

Click on Register-->Things for registering a new Thing as shown below.

Welcome to the AWS IoT Console

To get started, you can jump into the recommended starting points below, or explore other learning resources as needed.

Dashboard

Connect

Registry

Things

Types

Security

Rules

Test

See how AWS IoT works

Connect to AWS IoT

Explore documentation

The image shows the AWS IoT Console's welcome screen. On the left, there's a sidebar with various navigation options: Dashboard, Connect, Registry, Things (which is highlighted with a red border), Types, Security, Rules, and Test. Below these are three large cards: 'See how AWS IoT works' (with icons for a speech bubble, a car, a shield, and a sensor), 'Connect to AWS IoT' (with a globe icon), and 'Explore documentation' (with icons for a book, a laptop, and a smartphone). The top right of the screen has a message: 'Welcome to the AWS IoT Console' and 'To get started, you can jump into the recommended starting points below, or explore other learning resources as needed.'

Click on Create to create a new Thing.

The screenshot shows the AWS IoT Things dashboard. On the left, there's a sidebar with various icons and links: Dashboard, Connect, Registry (which is selected and shown in bold), Things (under Registry), Types, Security, Rules, and Test. The main area is titled "Things" and contains a search bar with the placeholder "Search things". In the top right corner, there's a blue "Create" button. A red rectangular box highlights this "Create" button, indicating where the user should click to create a new Thing.

Provide a unique **Thing** name and click on Create to proceed.

The screenshot shows a user interface for registering a new thing. At the top, a blue header bar contains the text "Register a thing". To the left of the header is a small gray square button with a white left-pointing arrow. Below the header, a descriptive text states: "This step creates an entry in the thing registry and a thing shadow for your device." Underneath this text is a form field labeled "Name" with a text input box containing the value "myTestThing". Below the name input is a link labeled "Show optional configuration (this can be done later) ▾". In the bottom right corner of the main content area, there is a teal-colored button with the text "Create thing". This button is highlighted with a red rounded rectangular border.

Register a thing

This step creates an entry in the thing registry and a thing shadow for your device.

Name

myTestThing

Show optional configuration (this can be done later) ▾

Create thing

myTestThing is SuccessFul created as shown below.

The screenshot shows the AWS IoT Things console interface. At the top, it displays the title "THING myTestThing NO TYPE" and an "Actions" dropdown menu. On the left, there's a vertical navigation bar with tabs: Details (selected), Security, Shadow, Interact, and Activity. The main content area has two sections: "Thing ARN" and "Type". The "Thing ARN" section contains the text "arn:aws:iot:us-west-2:242173728165:thing/myTestThing". The "Type" section shows a search input field with the placeholder "No type" and an "Edit" button. Below these sections, a green banner at the bottom left says "Successfully created thing." with a close button. At the bottom right, there's another "Edit" button next to the text "0 Attributes".

THING
myTestThing
NO TYPE

Actions ▾

Details Thing ARN

Security A thing Amazon Resource Name uniquely identifies this thing.
arn:aws:iot:us-west-2:242173728165:thing/myTestThing

Shadow

Interact

Activity

Type Edit

No type

0 Attributes Edit

Successfully created thing. ✎

Get the MQTT HOST name and thing name from Interact menu.

- These will be used to connect with AWS MQTT and also to subscribe/publish the topics.

The screenshot shows the AWS IoT Thing Details page for a thing named "myTestThing". The "Interact" tab is selected in the sidebar. The main content area displays the MQTT configuration:

MQTT

Use topics to enable applications and things to get, update, or delete the state information for a Thing (Thing Shadow) [Learn more](#)

Update to this thing shadow

`$aws/things/myTestThing/shadow/update`

Update to this thing shadow was accepted

`$aws/things/myTestThing/shadow/update/accepted`

A red box highlights the text "MQTT HOST name that will be used in MQTT Setup" next to the endpoint URL, and another red box highlights the topic prefix "\$aws/things/myTestThing/shadow/update" under "Update to this thing shadow".

Generate certificates for new thing

- Go to Security menu and click on Create Certificate to register the certificates.

The screenshot shows the AWS IoT console interface for a thing named "myTestThing". The "Security" tab is selected and highlighted with a red box. On the right, under the "Certificates" section, there is a callout box with the following text:

To securely connect to AWS IoT, your thing will need a certificate and policy.
Certificates help things establish a secure connection. AWS IoT policies give things permission to access AWS IoT resources (like other things, MQTT topics, or thing shadows).

Below the text are two buttons: "Create certificate" (highlighted with a red box) and "View other options".

THING
myTestThing
NO TYPE
Actions ▾

Details

Security

Shadow

Interact

Activity

Certificates

Create certificate

View other options

Click on the activate button to activate the certificates

Certificate created!

Download these files and save them in a safe place. Certificates can be retrieved at any time, but the private and public keys cannot be retrieved after you close this page.

In order to connect a device, you need to download the following:

A certificate for this thing	7054b132c8.cert.pem	Download
A public key	7054b132c8.public.key	Download
A private key	7054b132c8.private.key	Download

You also need to download a root CA for AWS IoT from Symantec:

A root CA for AWS IoT [Download](#)

Deactivate

Successfully activated certificate. ✘

```
graph LR; A[Download] --> B[Certificate]; C[Download] --> D[Private Key]; E[Download] --> F[root CA]
```

Create Policy

- A policy is attached to a key/cert
 - It tells what this key/cert can do

The screenshot shows the AWS IoT Policies management interface. On the left, a navigation sidebar lists various AWS IoT services: Activity, Onboard, Manage (Things, Types, Thing groups, Billing groups, Jobs, Tunnels), Greengrass, Secure (Certificates, Policies, CAs, Role Aliases, Authorizers), Defend, Act, and Test. The 'Policies' link under 'Secure' is highlighted with a red box. The main content area is titled 'Policies' and contains a search bar labeled 'Search policies'. Below the search bar is a table listing four policies: 'Name' (checkbox), 'mqtt_gps_uploading' (checkbox), 'cse520_policy' (checkbox), and 'RuiPi_policy' (checkbox). Each policy row has a '...' button on the right. A large red box highlights the blue 'Create' button in the top right corner of the main content area.

Create a policy

Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the [AWS IoT Policies documentation page](#).

Name

cse521_policy

Add statements

Policy statements define the types of actions that can be performed by a resource.

[Advanced mode](#)

Action

iot:*

Here, we grant it all permissions for demo!

Resource ARN

*

Effect

Allow Deny

[Remove](#)

[Add statement](#)

[Create](#)

Create

Policies

Search policies



Name

mqtt_gps_uploading

...

cse520_policy

...

RuiPi_policy

This is the policy you created

...

cse521_policy

...

Attach Policy

➤ Attach Policy to the key/cert

- A policy tells what this key/cert can do

The screenshot shows two panels of the AWS IoT console. On the left, the 'Things' section of the 'Manage' menu is selected. A blue arrow points from the 'Things' list to the 'Certificates' section on the right. In the 'Certificates' section, a certificate with the ID 'e1edf9915e9da5a73d...' is listed. A red box highlights the 'Security' tab in the navigation bar of the right panel, and another red box highlights the certificate ID in the list.

Activity

► Onboard

▼ Manage

Things

Types

Thing groups

Billing groups

Jobs

Tunnels

► Greengrass

▼ Secure

Certificates

Policies

Search things

Name

COREY_LAB_PC

cse520

RuiPi

locationmqtt

cse521

Details

Security

Thing groups

Billing Groups

Shadows

Interact

Activity

Jobs

Violations

Defender metrics

Create certificate

View other

e1edf9915e9da5a73d...

Click the cert you just created

The screenshot shows the AWS IoT Certificate Details page. At the top, there's a breadcrumb navigation: 'CSE_521_Temp > 208f60eb4fab1b02f5d656963382b05cd5c690b481e710c6a3e899a...'. Below it, the word 'CERTIFICATE' is followed by the certificate ID. A status indicator 'INACTIVE' is shown. To the right is a 'Actions' dropdown menu with several options: 'Activate', 'Deactivate', 'Revoke', 'Accept transfer', 'Reject transfer', 'Revoke transfer', 'Start transfer', 'Attach policy' (which is highlighted with a red box), 'Attach thing', 'Download', and 'Delete'. The 'Actions' button itself is also highlighted with a red box.

Details

Certificate ARN

Policies

A certificate Amazon Resource Name (ARN) uniquely identifies this certificate.

Things

Non-compliance

`arn:aws:iot:us-east-1:006025899016:cert/208f60eb4fab1b02f5d656963382b05cd5c690b481e710c6a3e899a...`

Details

Issuer

OU=Amazon Web Services O=Amazon.com Inc. L=Seattle ST=Washington C=US

Subject

CN=AWS IoT Certificate

Create date

Aug 25, 2019 12:56:24 PM -0500

Effective date

Aug 25, 2019 12:54:24 PM -0500

Expiration date

Dec 31, 2049 5:59:59 PM -0600

X

Attach policies to certificate(s)

Policies will be attached to the following certificate(s):

e1edf9915e9da5a73da0f9385df408d76d28ec364eff1c3e9d4dc5c4b8b34e08

Choose one or more policies

 Search policies

mqtt_gps_uploading

[View](#)

cse520_policy

[View](#)

RuiPi_policy

[View](#)

cse521_policy

[View](#)

0 policies selected

[Cancel](#)

[Attach](#)

September 16, 2020, 11:11:39 (UTC-0400)

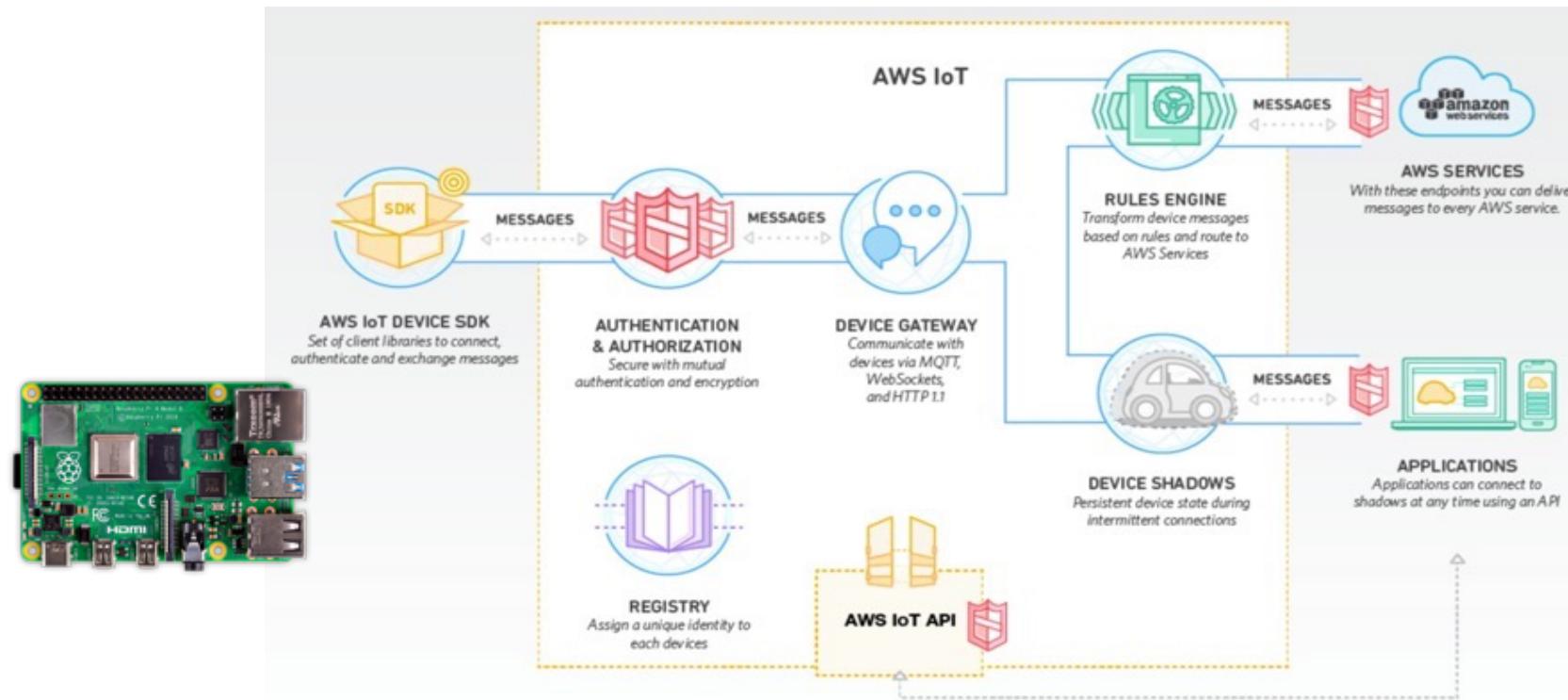
Expiration date

December 31, 2049, 18:59:59 (UTC-0500)

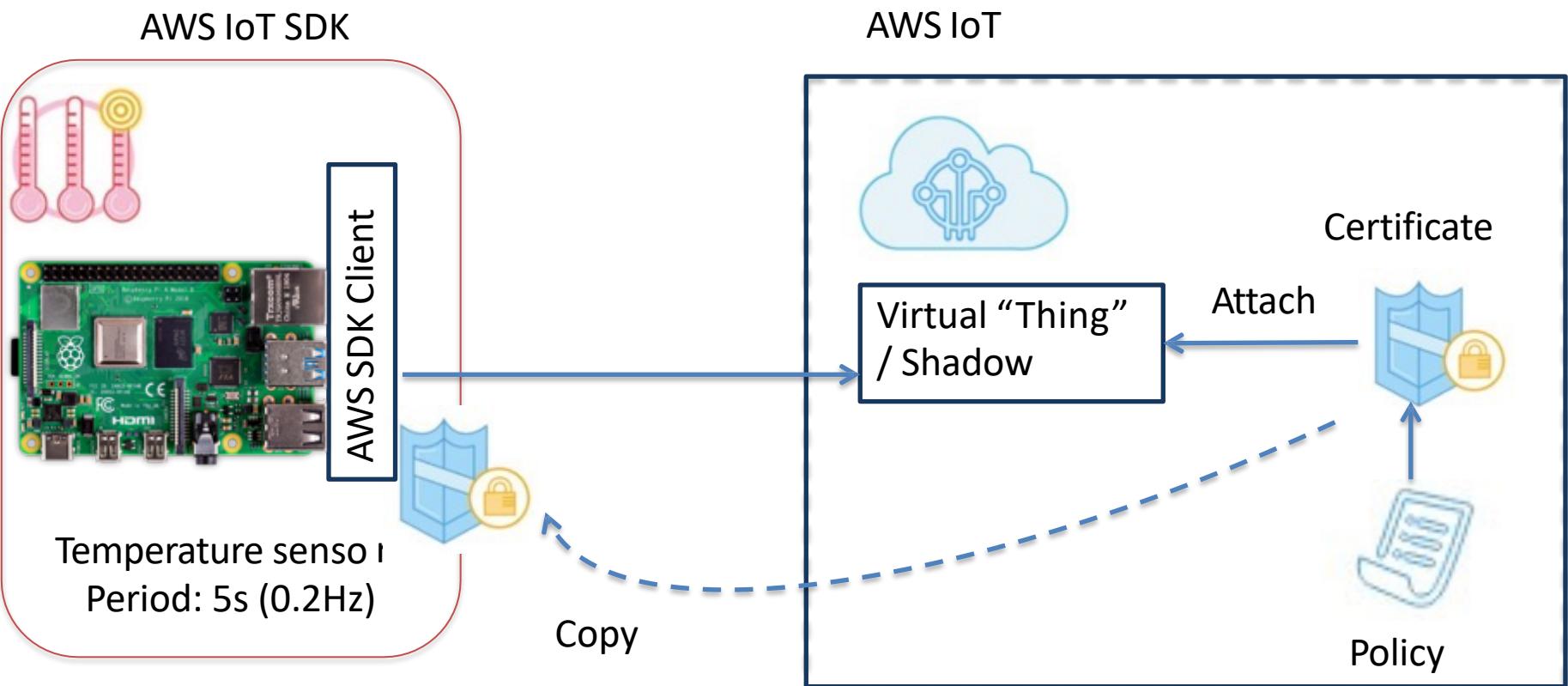
Now, you have a virtual thing on AWS!

AWS Things Summary

- ❑ Certificate: authenticate the device
- ❑ Policy: define the roles/permissions of the certificate
- ❑ Virtual copy of the thing (Shadow): Store/retrieve some information



Step 2: Connect a “Physical” Device



[Connect AWS IoT with ESP32:](#)

https://exploreembedded.com/wiki/AWS_IOT_with_Arduino_ESP32

Connect your Device

➤ Copy certificates to your **physical things**

- ❑ Downloaded before!

In order to connect a device, you need to download the following:

A certificate for this thing	208f60eb4f.cert.pem	Download
A public key	208f60eb4f.public.key	Download
A private key	208f60eb4f.private.key	Download

You also need to download a root CA for AWS IoT:
A root CA for AWS IoT [Download](#)

[Download all keys and root CA](#)

➤ Choose your AWS SDK (support MQTT)

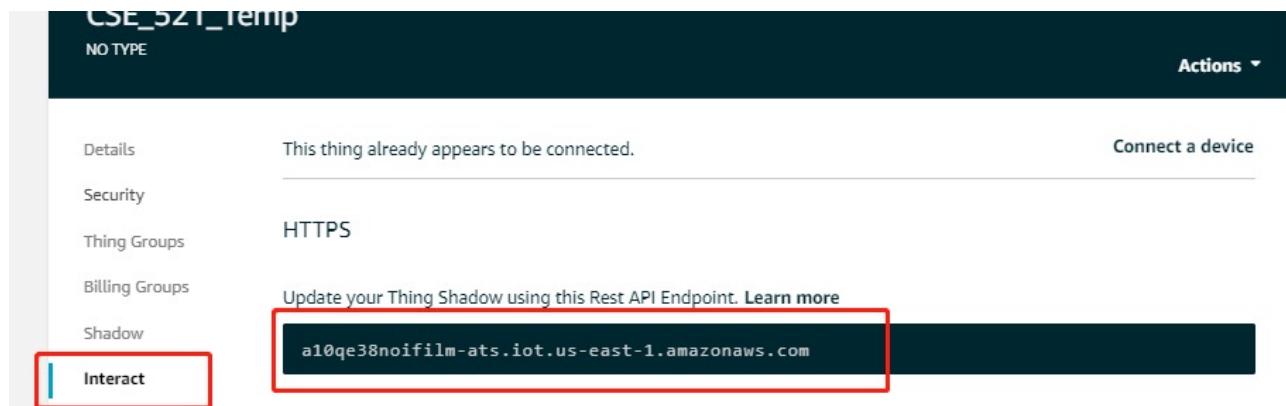
- ❑ Node JS
- ❑ Python (pip install AWSIoTPythonSDK)
- ❑ Java
- ❑ ...

➤ Set up your client with SDK and the certificates



Some Notes

- 1. Copy the certificates/keys to your real thing
- 2. You will need the endpoint and port (8883)



Host(Endpoint)

- 3. Set up your configuration of the code with SDK

```
host = "a10qe38noifilm-ats.iot.us-east-1.amazonaws.com" # Your thing's endpoint. See tutorial slides
rootCAPath = "root-CA.crt"
certificatePath = "e1edf9915e-certificate.pem.crt"
privateKeyPath = "e1edf9915e-private.pem.key"
port = 8883
clientId = "CSE521"
topic = "$aws/things/cse521/shadow/update" # Shadow topic of your Thing
```

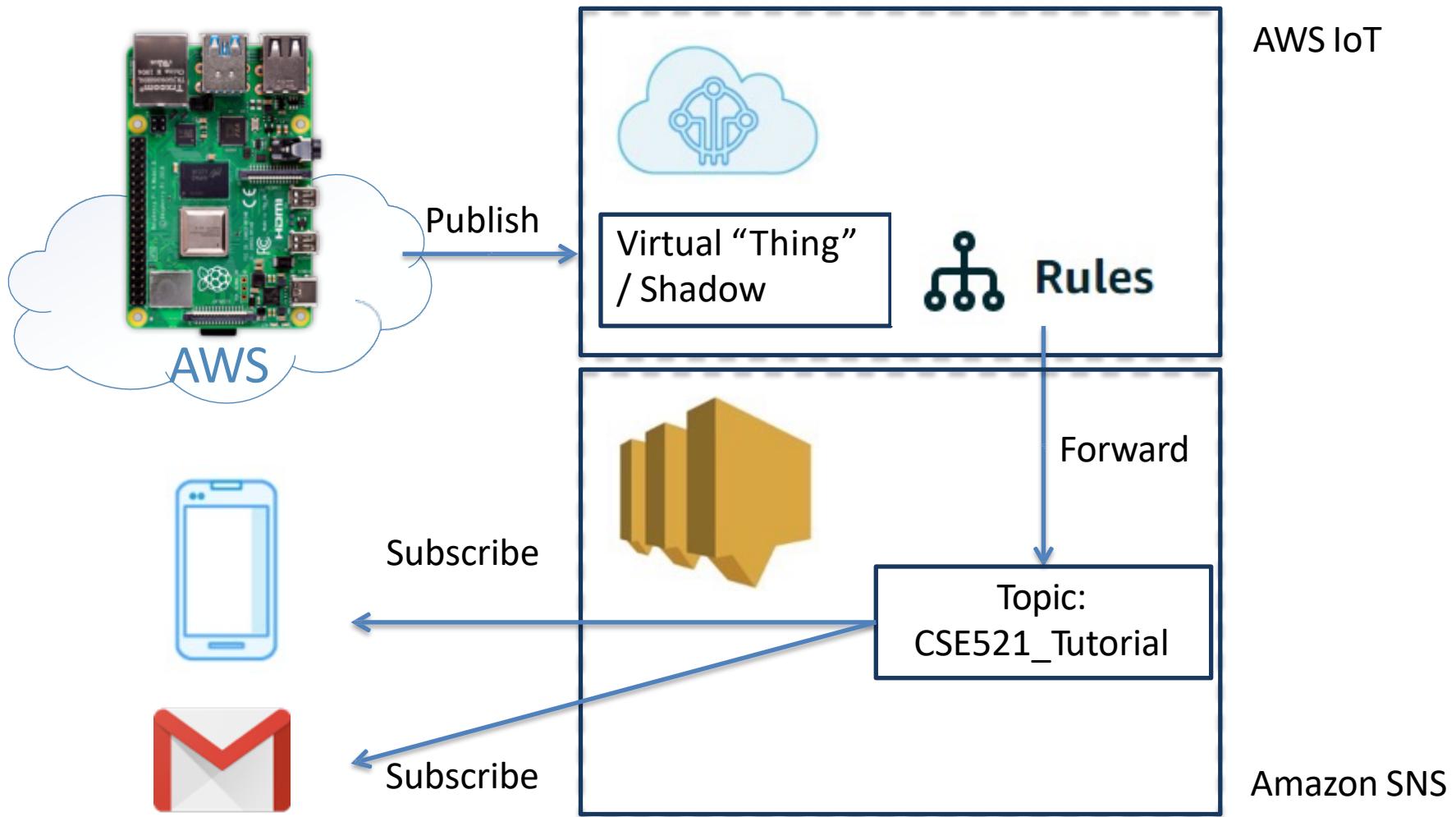
Change your code
accordingly!

SDK and Demo Codes

- <https://github.com/aws/aws-iot-device-sdk-python>
- <https://github.com/aws/aws-iot-device-sdk-python/blob/master/samples/basicPubSub/basicPubSub.py>

More: Rule Engine, Link with SNS services

➤ Simple Notification Service



Create a Rule in Amazon IoT

➤ Add a query to filter your interesting topic (event)

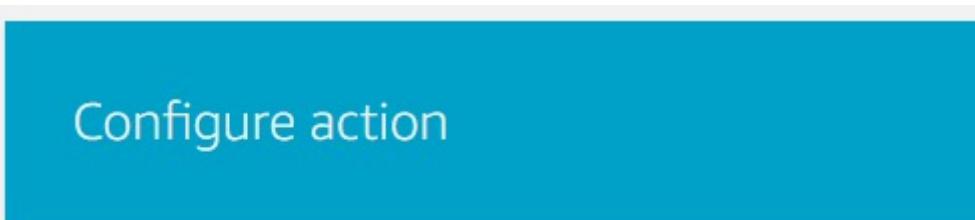
Rule query statement

SELECT <Attribute> FROM <Topic Filter> WHERE <Condition>. For example: SELECT temperature FROM 'iot/topic' WHERE temperature > 50. To learn more, see [AWS IoT SQL Reference](#).

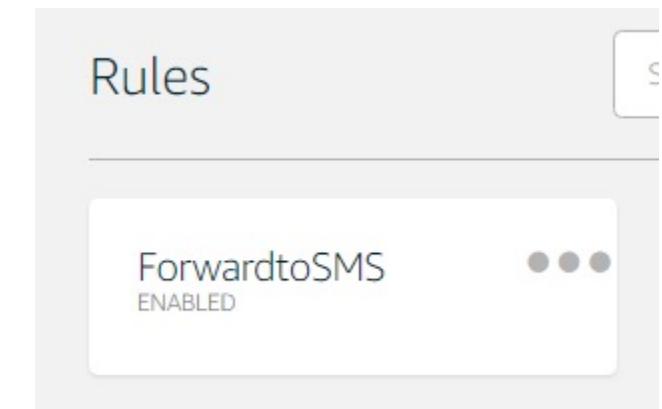
```
1 SELECT * FROM '$aws/things/cse521/shadow/update'
```

➤ Add an Action:

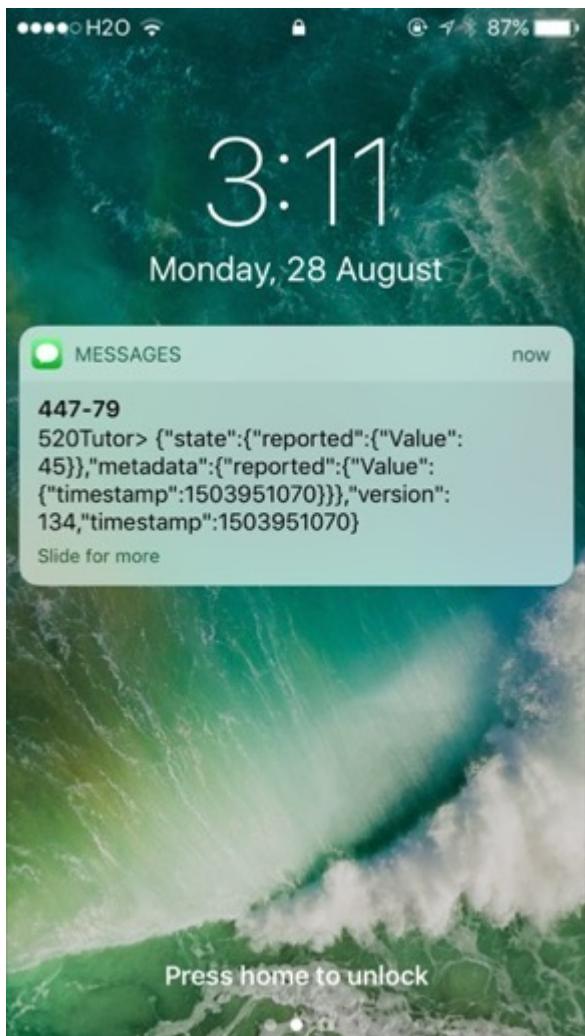
- Forward this message to SNS
- Specify Dest ARN
- Enable Rule



SNS
Send a message as an SNS push notification



Notification on SMS & Email



AWS Notification Message

Inbox

520Tutor no-reply@sns.amazonaws.com 3:11 PM (28 minutes ago)

to me

{"state":{"reported":{"Value":45}}, "metadata":{"reported":{"Value":{"timestamp":1503951070}}}, "version":134, "timestamp":1503951070}

--

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:

https://sns.us-west-2.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-west-2:401317363811:CSE520S_Tutorial:00c54352-7d1a-4c09-9cc1-15aed3c395e3&Endpoint=naroahlee@gmail.com

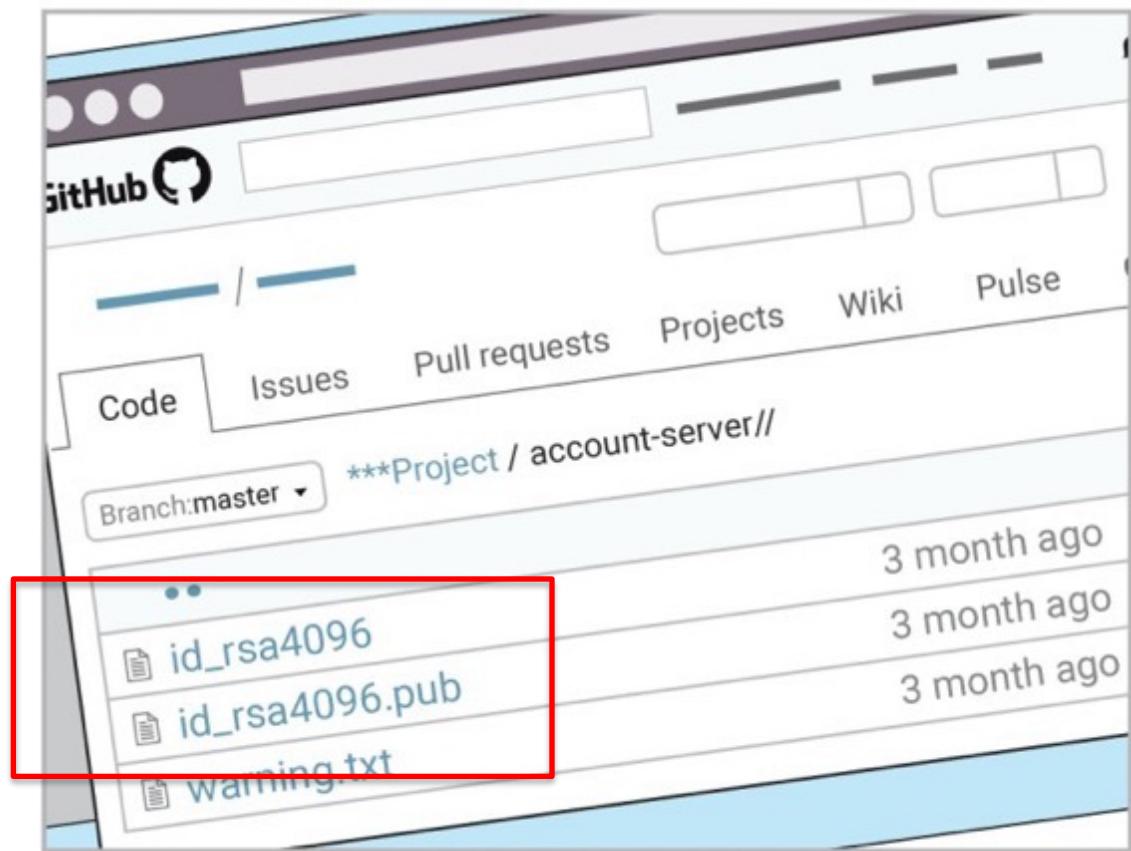
Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at

<https://aws.amazon.com/support>

One More Thing: Account Security

➤ DON'T UPLOAD YOUR KEY PUBLICLY!!!

Time to Open Source!



What if... \$50,000 AWS Bill!

Quora

Ask or Search Quora

Ask Question

Fraud

Amazon Web Services

Amazon.com (product)

Hackers

+3

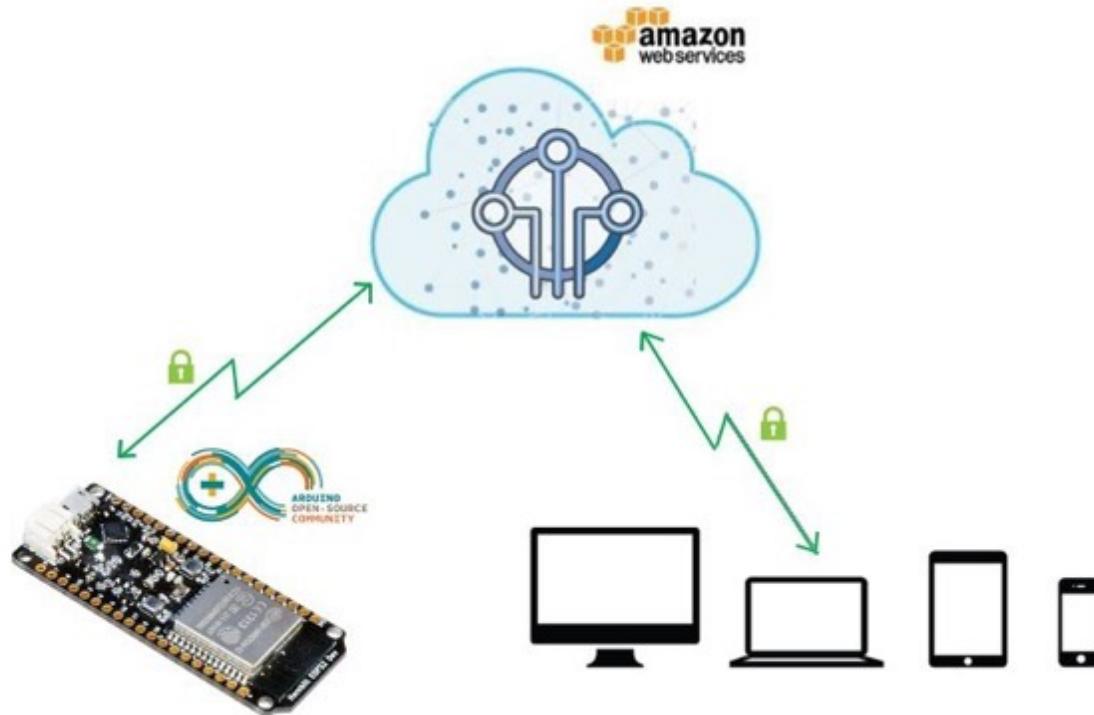


My AWS account was hacked and I have a \$50,000 bill, how can I reduce the amount I need to pay?

For years, my bill was never above \$350/month on my single AWS instance. Then over the weekend someone got hold of my private key and launched hundreds of instances and racked up a \$50,000 bill before I found out about it on Tuesday. Amazon had sent a warning by email at \$15,000 saying they had found **our key posted publicly**, but I didn't see it. Naturally, this is a devastating amount of money to pay. I'm not saying I shouldn't pay anything, but this just a crazy amount in context. Amazon knew the account was compromised, that is why they sent an email, they knew the account history and I had only spent \$213 the previous month. I almost feel they deliberately let it ride to try to earn more money. Does anyone have any experience with this sort of problem?

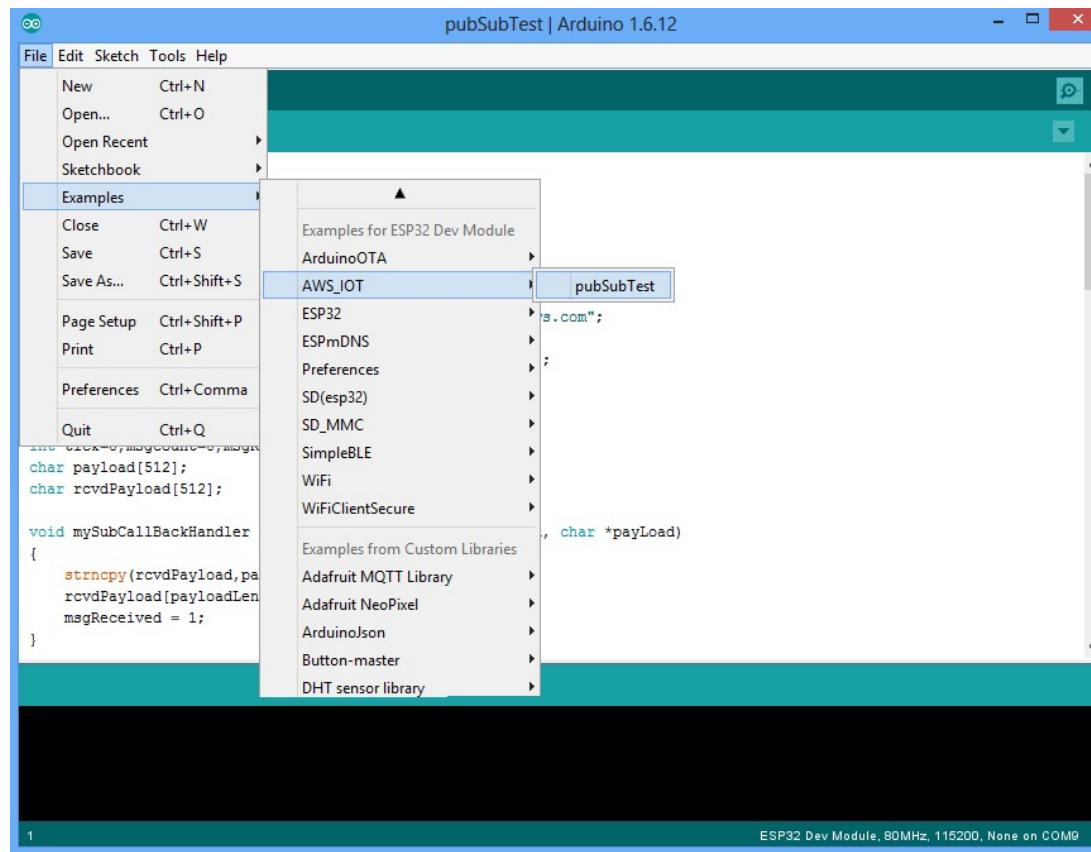
AWS IoT for ESP32

- https://exploreembedded.com/wiki/AWS_IOT_with_Arduino_ESP32



AWS IoT for ESP32

- Download the AWS IOT library for Arduino ESP32 from [this link](#). Now open the first example pubSubTest.ino as shown below.





File Edit Sketch Tools Help



pubSubTest §

```
#include <AWS_IOT.h>
#include <WiFi.h>

AWS_IOT hornbill;

char WIFI_SSID[]="your Wifi SSID";
char WIFI_PASSWORD[]="Wifi Password";
char HOST_ADDRESS[]="AWS host address";
char CLIENT_ID[]="client id";
char TOPIC_NAME[]="your thing/topic name";

int status = WL_IDLE_STATUS;
int tick=0,msgCount=0,msgReceived = 0;
char payload[512];
char rcvdPayload[512];

void mySubCallBackHandler (char *topicName, int payloadLen, char *payLoad)
{
    strncpy(rcvdPayload,payLoad,payloadLen);
    rcvdPayload[payloadLen] = 0;
    msgReceived = 1;
}
```

AWS IoT for ESP32

- The below certificate and private key needs to be copied to `aws_iot_certificate.c` file in the form of array.
 1. `aws-root-ca.pem`
 2. `certificate.pem.crt`
 3. `private.pem.key`

Upload the sketch and verify the output

- Publish and subscribe(received) messages printed on terminal.

The screenshot shows a terminal window titled "COM3" displaying MQTT communication between an ESP32 and a broker. The window includes a text input field with a "Send" button and two status indicators at the bottom: "Autoscroll" checked and "No line ending" with a baud rate of "115200 baud".

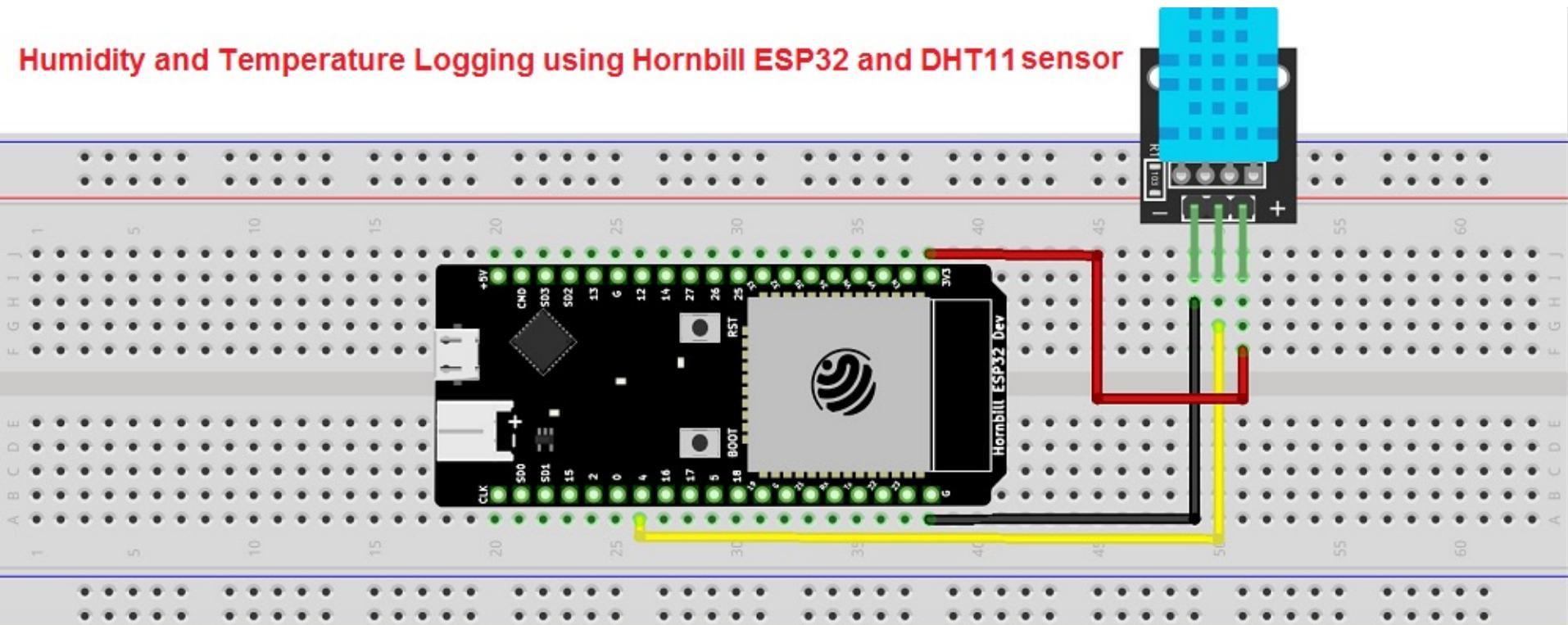
```
Attempting to connect to SSID: ExploreEmbedded.com
Attempting to connect to SSID: ExploreEmbedded.com
Connected to wifi
Connected to AWS
Subscribe Successfull
Publish failed
Publish Message:Hello from hornbill ESP32 : 1
Received Message:Hello from hornbill ESP32 : 1
Publish Message:Hello from hornbill ESP32 : 2
Received Message:Hello from hornbill ESP32 : 2
Publish Message:Hello from hornbill ESP32 : 3
Received Message:Hello from hornbill ESP32 : 3
Publish Message:Hello from hornbill ESP32 : 4
Received Message:Hello from hornbill ESP32 : 4
Publish Message:Hello from hornbill ESP32 : 5
Received Message:Hello from hornbill ESP32 : 5
Publish failed
Publish Message:Hello from hornbill ESP32 : 7
Received Message:Hello from hornbill ESP32 : 7
Publish Message:Hello from hornbill ESP32 : 8
Received Message:Hello from hornbill ESP32 : 8
Publish Message:Hello from hornbill ESP32 : 9
Received Message:Hello from hornbill ESP32 : 9
Publish Message:Hello from hornbill ESP32 : 10
Received Message:Hello from hornbill ESP32 : 10
```

Annotations highlight specific lines:

- A red box labeled "Published Message" surrounds the line "Publish Message:Hello from hornbill ESP32 : 1".
- A red box labeled "Same message is received back as we have subscribed to **same topic**" surrounds the line "Received Message:Hello from hornbill ESP32 : 1".

Logging Temperature and Humidity using DHT11 sensor

Humidity and Temperature Logging using Hornbill ESP32 and DHT11 sensor



fritzing

Logging Temperature and Humidity using DHT11 sensor

```
#include <WiFi.h>
#include "DHT.h"

#define DHTPIN 4 // what digital pin we're connected to

// Uncomment whatever type you're using!
#define DHTTYPE DHT11 // DHT 11
//#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
//#define DHTTYPE DHT21 // DHT 21 (AM2301)

DHT dht(DHTPIN, DHTTYPE);

AWS_IOT hornbill; // AWS_IOT instance

char WIFI_SSID[]="your Wifi SSID";
char WIFI_PASSWORD[]="Wifi Password";
char HOST_ADDRESS[]="AWS host address";
char CLIENT_ID[]="client id";
char TOPIC_NAME[]="your thing/topic name";

int status = WL_IDLE_STATUS;
int tick=0,msgCount=0,msgReceived = 0;
char payload[512];
char rcvdPayload[512];
```

```
Serial.begin(115200);
delay(2000);

while (status != WL_CONNECTED)
{
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(WIFI_SSID);
    // Connect to WPA/WPA2 network. Change this line if using open or WEP network:
    status = WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

    // wait 5 seconds for connection:
    delay(5000);
}

Serial.println("Connected to wifi");

if(hornbill.connect(HOST_ADDRESS,CLIENT_ID)== 0) // Connect to AWS using Host Address and Client ID
{
    Serial.println("Connected to AWS");
    delay(1000);
}
else
{
    Serial.println("AWS connection failed, Check the HOST Address");
    while(1);
}

delay(2000);

dht.begin(); //Initialize the DHT11 sensor
}
```

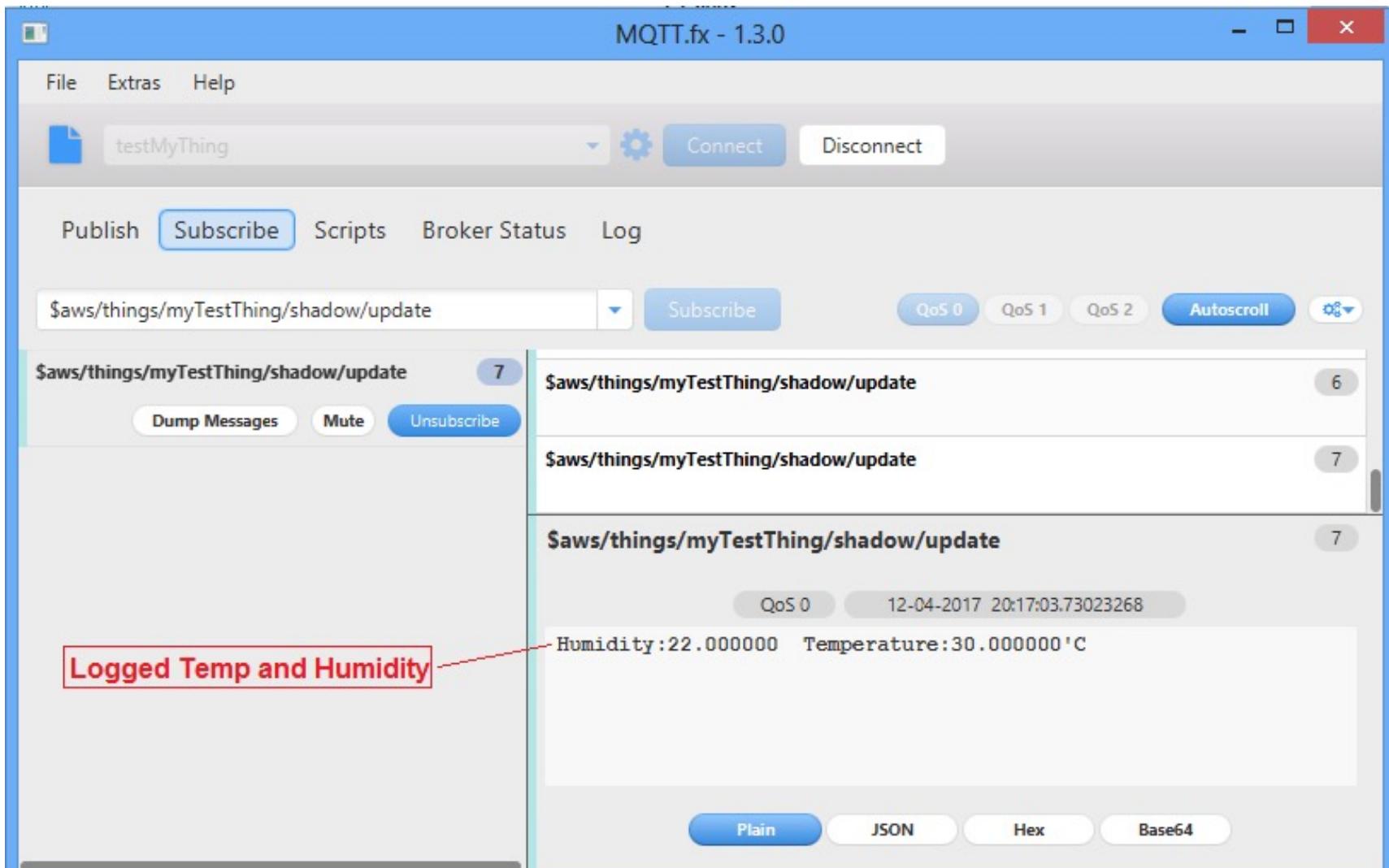
Logging Temperature and Humidity using DHT11 sensor

```
// Reading temperature or humidity takes about 250 milliseconds!
// Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
float h = dht.readHumidity();
// Read temperature as Celsius (the default)
float t = dht.readTemperature();
// Read temperature as Fahrenheit (isFahrenheit = true)
float f = dht.readTemperature(true);

// Check if any reads failed and exit early (to try again).
if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Failed to read from DHT sensor!");
}
else
{
    sprintf(payload,"Humidity:%f Temperature:%f'C",h,t); // Create the payload for publishing

    if(hornbill.publish(TOPIC_NAME,payload) == 0) // Publish the message(Temp and humidity)
    {
        Serial.print("Publish Message:");
        Serial.println(payload);
    }
    else
    {
        Serial.println("Publish failed");
    }
    // publish the temp and humidity every 5 seconds.
    vTaskDelay(5000 / portTICK_RATE_MS);
}
}
```

Temp and Humidity logged to AWS IOT server.



AWS IoT for Android

<https://www.linkedin.com/pulse/android-app-aws-iot-core-guide-felipe-ramos-da-silva/>

Next Week, AWS IoT Greengrass ML Inference

COMP.5650/4150.201
Internet of Things
Lecture 8.2: Speech Analysis

Instructor
Mohammad Arif UI Alam

Voice Based Analytics

- Voice can be analyzed, lots of useful information extracted
 - Who is talking? (Speaker identification): Human Speech Frequency ranges 100 – 3200Hz
 - How many social interactions a person has a day
 - Emotion of person while speaking
 - Anxiety, depression, intoxication, of person, etc.
- For speech recognition, voice analytics used to:
 - Discard useless information (background noise, etc)
 - Extract information useful for identifying linguistic content



Mel Frequency Cepstral Coefficients (MFCCs)

- MFCCs widely used in speech and speaker recognition for representing envelope of power spectrum of voice
- Popular approach in Speech recognition
 - MFCC features + Hidden Markov Model (HMM) classifiers

Human Speech

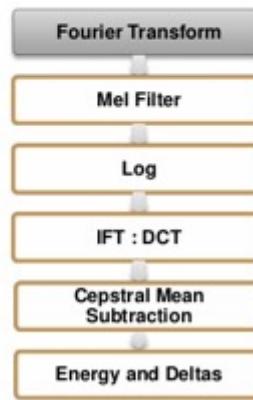
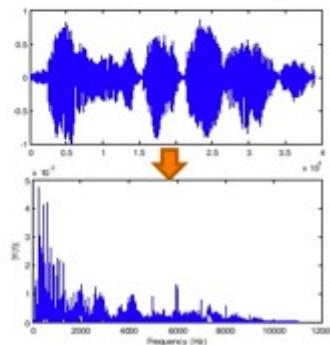
- Human perception of speech is linear until 1000 and logarithmic from there
- cepstrum is
 - Fourier Transform (logarithmic(Fourier Transform (Speech Signal)))

FEATURE EXTRACTION :

FOURIER TRANSFORM

- Gives information about the amount of energy at each frequency band

- FFT used
$$bin_k = \left| \sum_{n=1}^N s_w(n) e^{-j(n-1)k \frac{2\pi}{N}} \right|, \quad k = 0, 1, 2, \dots, N-1$$



MFCC Steps: Overview

1. Frame the signal into short frames.
2. For each frame calculate the periodogram estimate of the power spectrum.
3. Apply the mel filterbank to the power spectra, sum the energy in each filter.
4. Take the logarithm of all filterbank energies.
5. Take the DCT of the log filterbank energies. discrete cosine transform (DCT)
6. Keep DCT coefficients 2-13, discard the rest.

Step 1: Windowing

- Audio is continuously changing.
- Break into short segments (20-40 milliseconds)
- Can assume audio does not change in short window

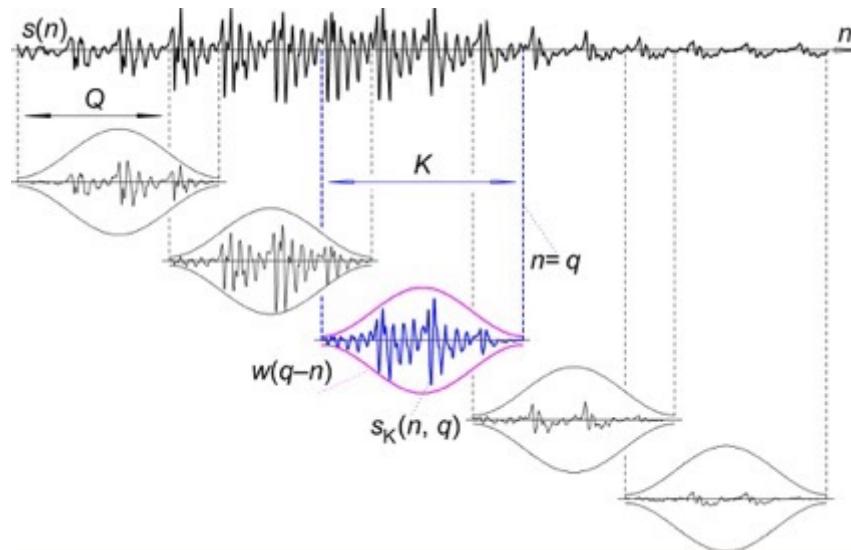
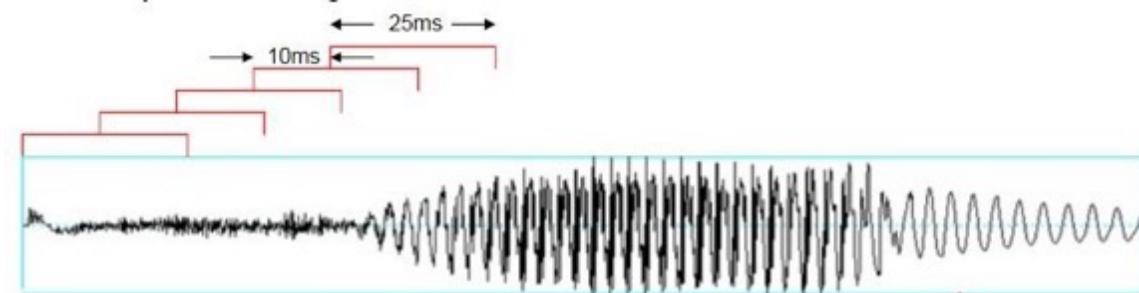


Image credits: <http://recognize-speech.com/preprocessing/cepstral-mean-normalization/10-preprocessing>

Step 1: Windowing

- Essentially, break into smaller overlapping frames
- Need to select frame length (e.g. 25 ms), shift (e.g. 10 ms)

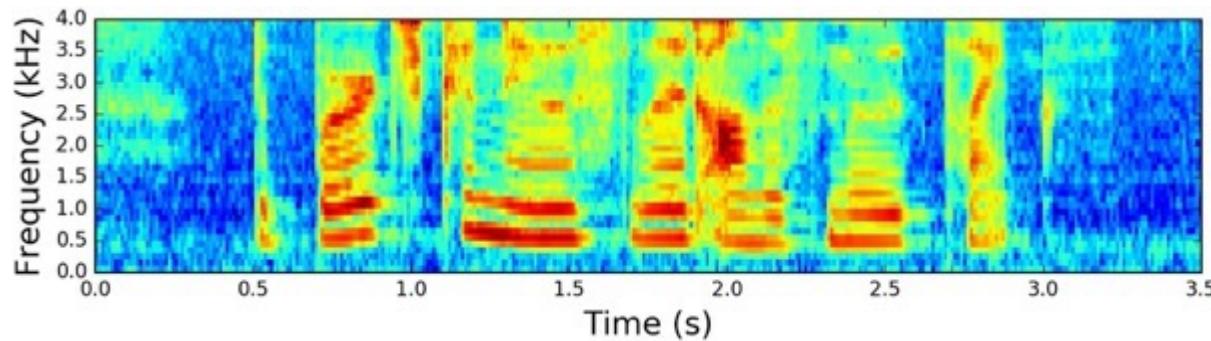


- So what? Can compare frames from reference vs test words (i.e. calculate distances between them)

<http://slideplayer.com/slide/7674116/>

Step 2: Calculate Power Spectrum of each Frame

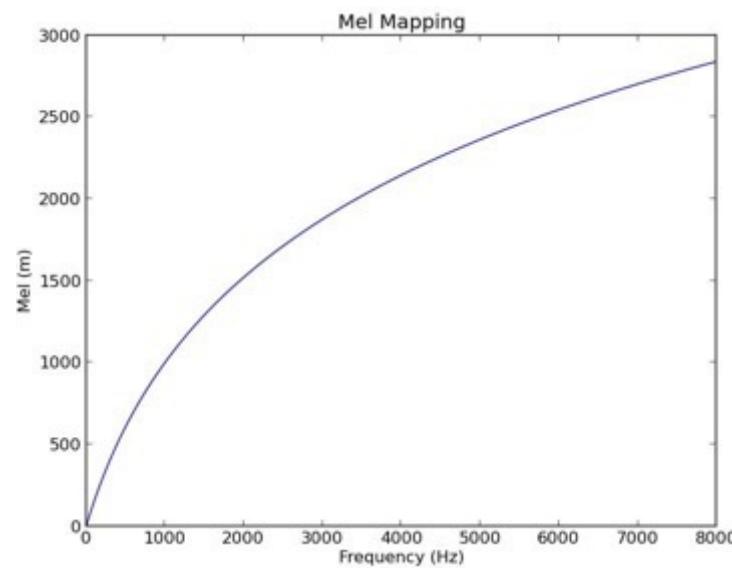
- Cochlea (Part of human ear) vibrates at different parts depending on sound frequency
- Power spectrum Periodogram similarly identifies frequencies present in each frame



Step 3: Apply Mel FilterBank

- Non-linear conversion from frequency to Mel Space

$$M(f) = 1125 \ln(1 + f/700) \quad (1)$$



Step 4: Apply Logarithm of Mel Filterbank

- Take log of filterbank energies at each frequency
- This step makes output mimic human hearing better
 - We don't hear loudness on a linear scale
 - Changes in loud noises may not sound different

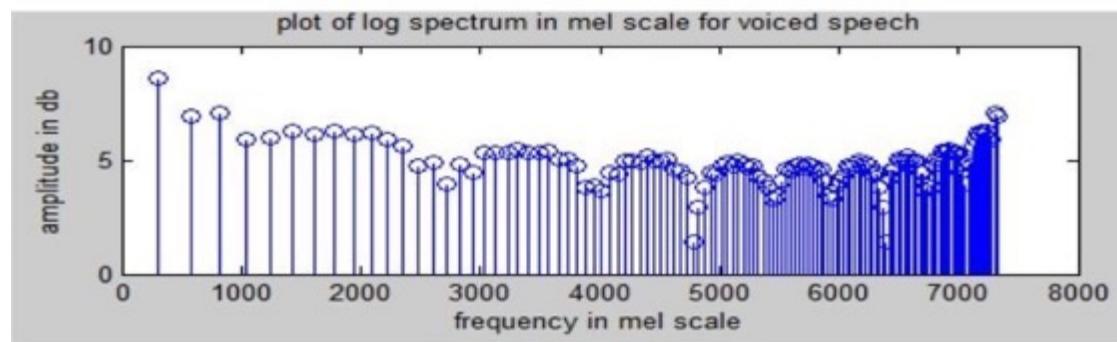


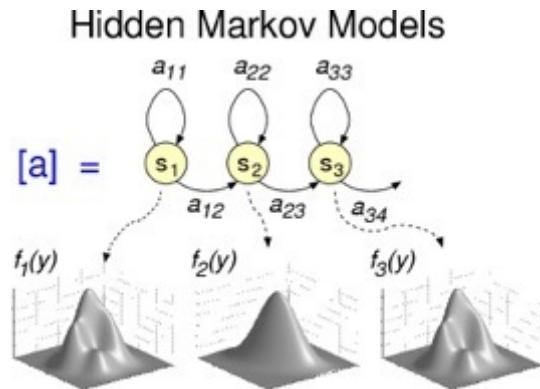
Fig.7. Spectrum of voiced speech

Step 5: **DCT of log filterbank**

- **Step 5: DCT of log filterbank:**
 - There are correlations between signals at different frequencies
 - Discrete Cosine Transform (DCT) extracts most useful and independent features
- Final result: 39 element acoustic vector used in speech processing algorithms

Speech Classification

- Human speech can be broken into phonemes
- Example of phoneme is /k/ in the words (**cat, school, skill**)
- Speech recognition tries to recognize sequence of phonemes in a word
- Typically uses Hidden Markov Model (HMM)
 - Recognizes letters, then words, then sentences



Speech Classification

- Human speech can be broken into phonemes
- Example of phoneme is /k/ in the words (**cat, school, skill**)
- Speech recognition tries to recognize sequence of phonemes in a word
- Dynamic Time Warping

Practice 8. 1 Speech Recognition and Print Texts Using Android

- Using SpeechRecognizer service

```
protected Intent intent;
protected SpeechRecognizer recognizer;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    textViewResults = (TextView)findViewById(R.id.textViewResults);

    intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, "en-US");
    intent.putExtra(RecognizerIntent.EXTRA_WEB_SEARCH_ONLY, "false");
    intent.putExtra(RecognizerIntent.EXTRA_SPEECH_INPUT_MINIMUM_LENGTH_MILLIS, "3000");

    recognizer = SpeechRecognizer.createSpeechRecognizer(this);
    recognizer.setRecognitionListener(this);
    recognizer.startListening(intent);
}
```

Practice 8. 1 Speech Recognition and Print Texts Using Android

- Using SpeechRecognizer service

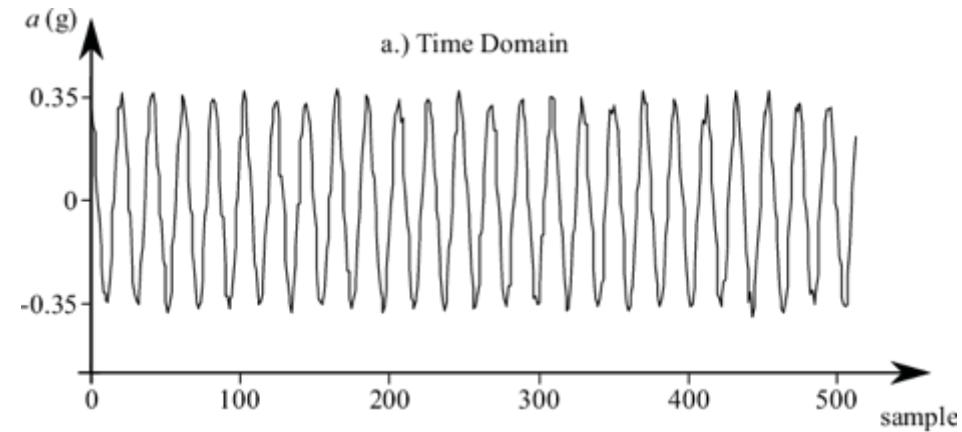
```
@Override  
public void onResults(Bundle results) {  
    ArrayList<String> words = results.getStringArrayList(SpeechRecognizer.RESULTS_RECOGNITION);  
  
    String text = "";  
  
    for (String word : words) {  
        text += word + " ";  
    }  
  
    textViewResults.setText(text);  
  
    //recognizer.stopListening();  
    //recognizer.startListening(intent);  
}
```

COMP.5650/4150.201
Internet of Things
Lecture 8.3: Sensor Signal
Processing

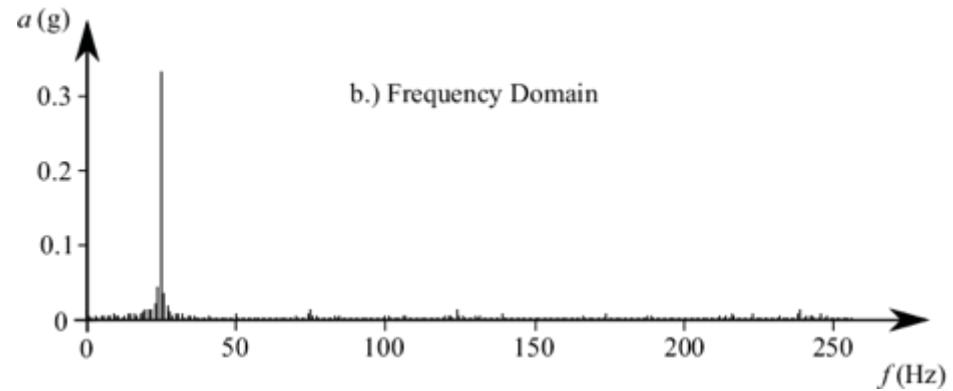
Instructor
Mohammad Arif UI Alam

Sensor Signals Two Types

- Time Domain
- Frequency Domain



Example: Accelerometer



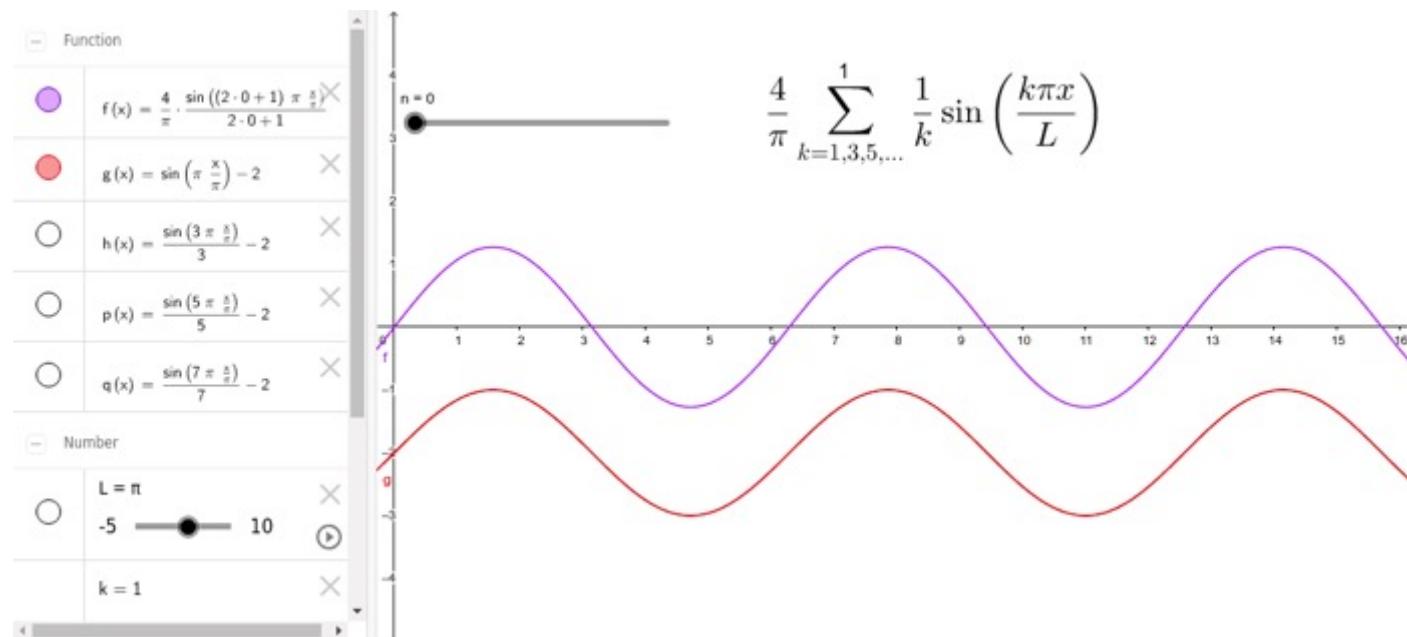
Transform Time Domain to Frequency Domain

- Fourier transform – nonrepetitive signals, transients.
- Laplace transform – electronic circuits and control systems.
- Z transform – discrete-time signals, digital signal processing.
- Wavelet transform — image analysis, data compression.

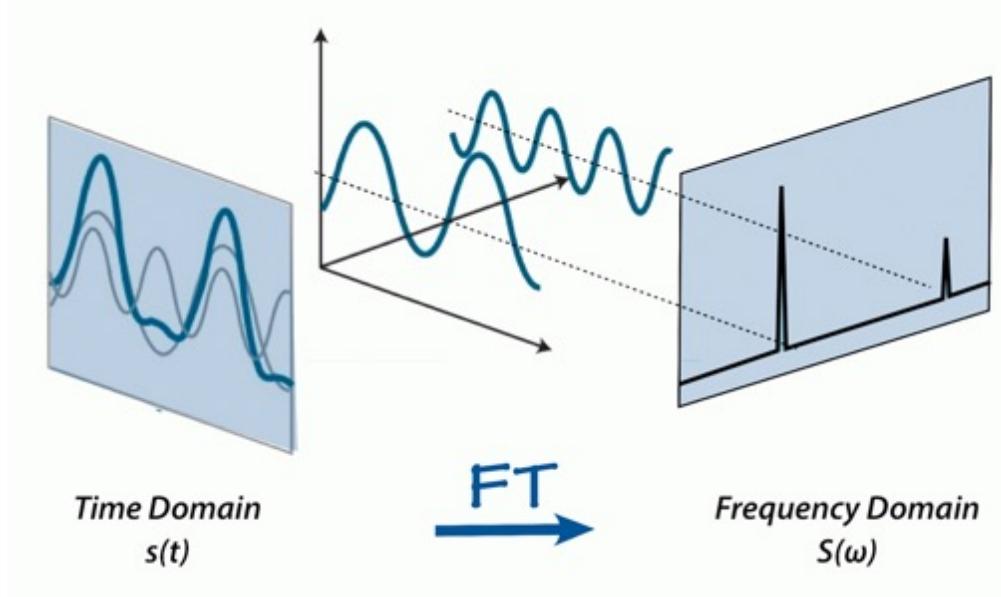
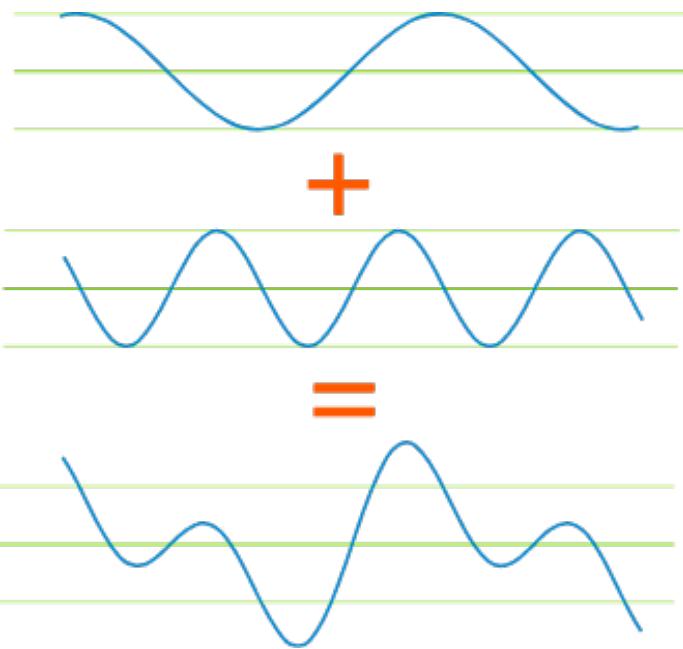
Fourier Series

Fourier Series

- A series of sinusoidal signals

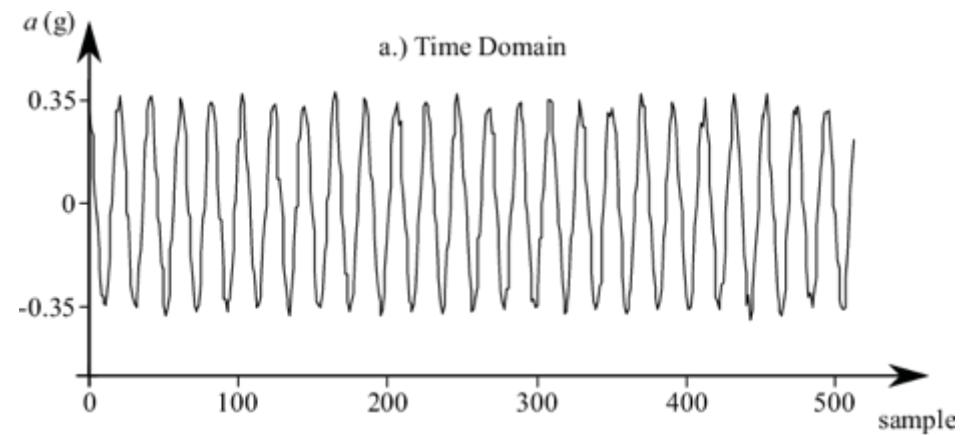


Fourier Transform

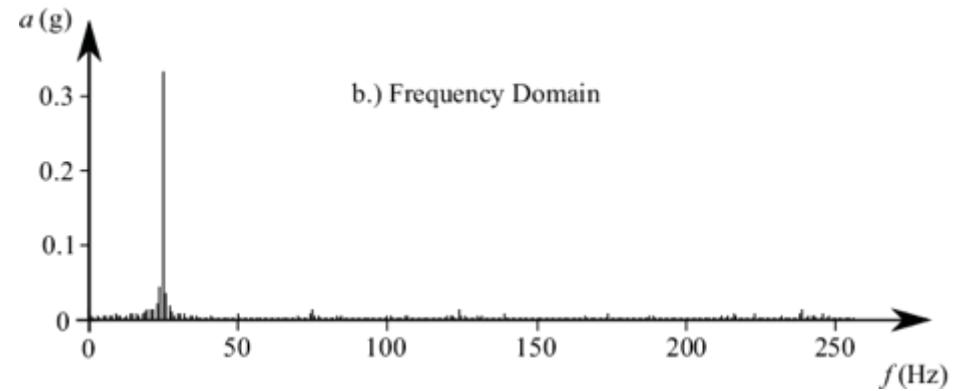


Accelerometer Frequency Domain Transform

- Time Domain
- Frequency Domain



Example: Accelerometer



Pass Filters

- Low pass filter
 - A low-pass filter (LPF) is a filter that passes signals with a frequency lower than a selected cutoff frequency and attenuates signals with frequencies higher than the cutoff frequency.
- High pass filter
 - A high-pass filter (HPF) is an electronic filter that passes signals with a frequency higher than a certain cutoff frequency and attenuates signals with frequencies lower than the cutoff frequency. The amount of attenuation for each frequency depends on the filter design.
- Band Pass filter
 - Band Pass Filters passes signals within a certain "band" or "spread" of frequencies without distorting the input signal or introducing extra noise. This band of frequencies can be any width and is commonly known as the filters Bandwidth.

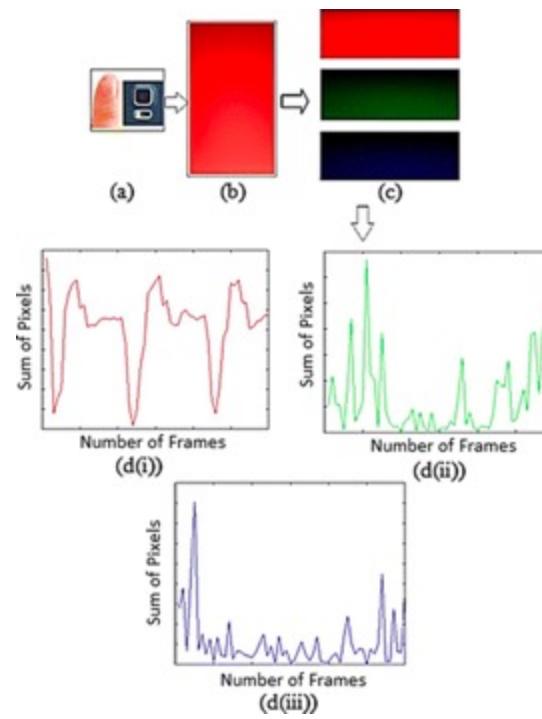
Usage

- Audio filter
 - Loud sound reduction, enhance small sounds, Audio amplifier
- Music filter
 - Instrument, voice separation
- Motion filter
 - Filter out large motion or small motion
- Noise reduction
 - Noises are unusual amplification or reduction of frequencies (in general)

Demo: Smartphone Camera Based Heart Rate Detection



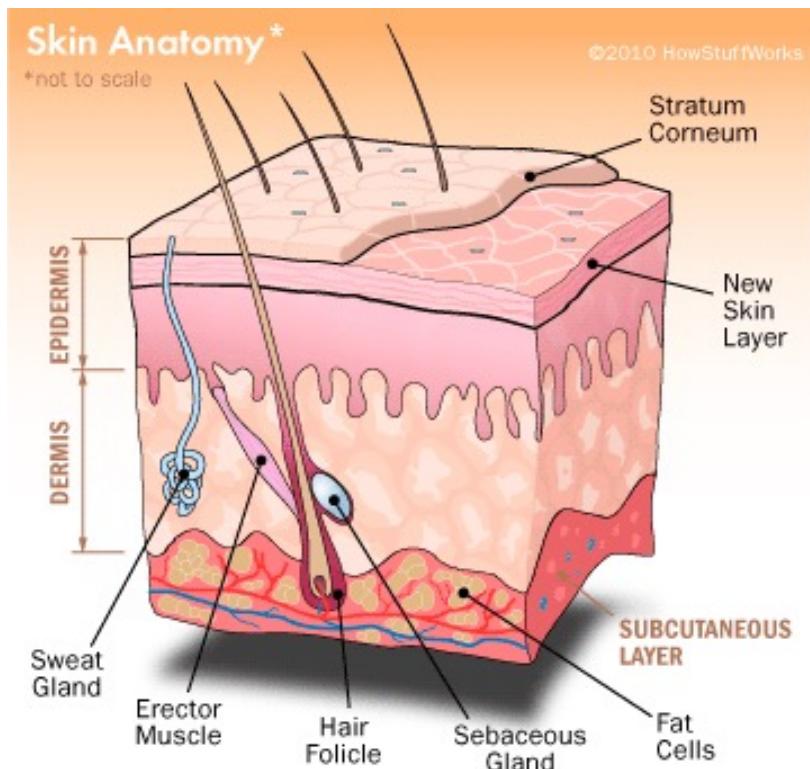
Demo: Smartphone Camera Based Heart Rate Detection



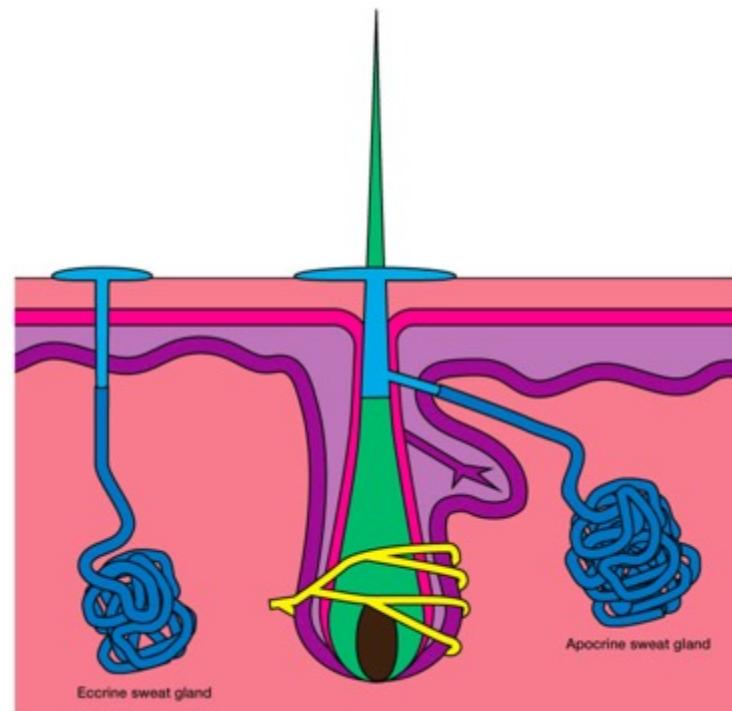
COMP.5650/4150.201
Internet of Things
Lecture 8.4: EDA and PPG

Instructor
Mohammad Arif UI Alam

Skin Anatomy

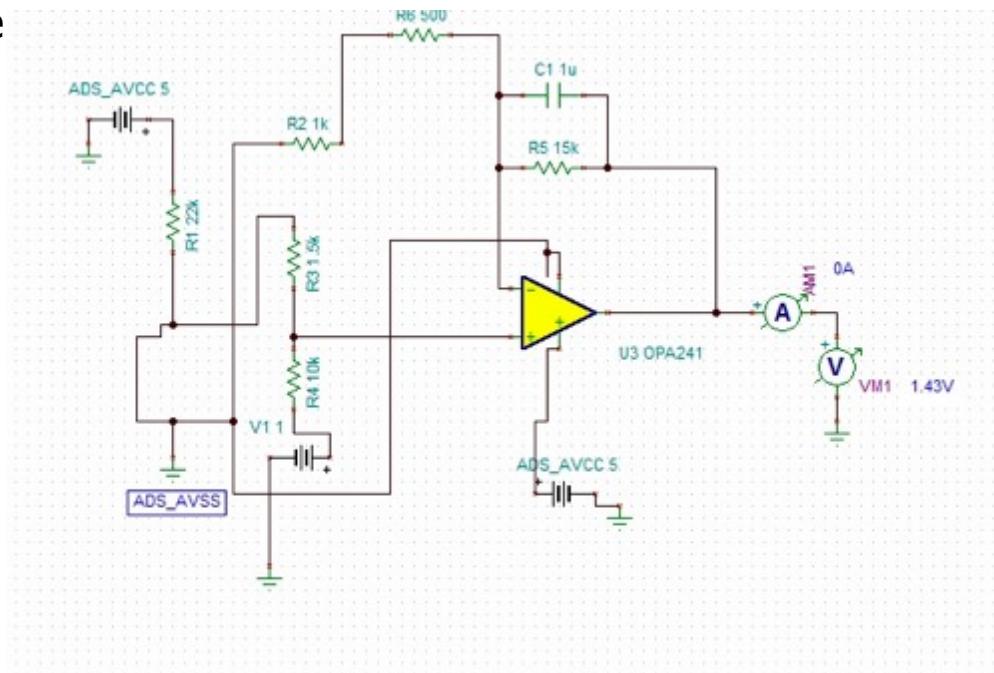


Some brain events (arousal) stress, depression, seizure



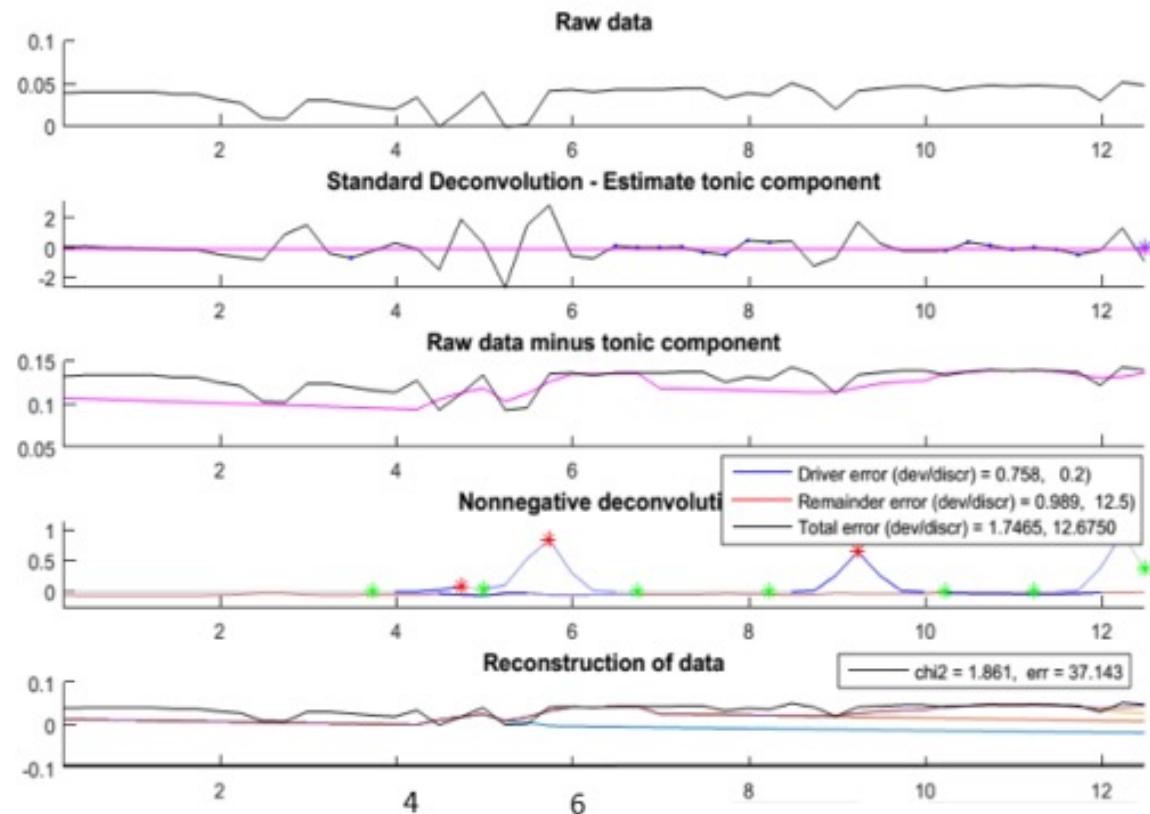
Electrodermal Activity

- Increase water level, increase conductance
- Electrodermal Response
- Voltage and Amplitude
- Quantified as Amplitude
 - Micro Siemense



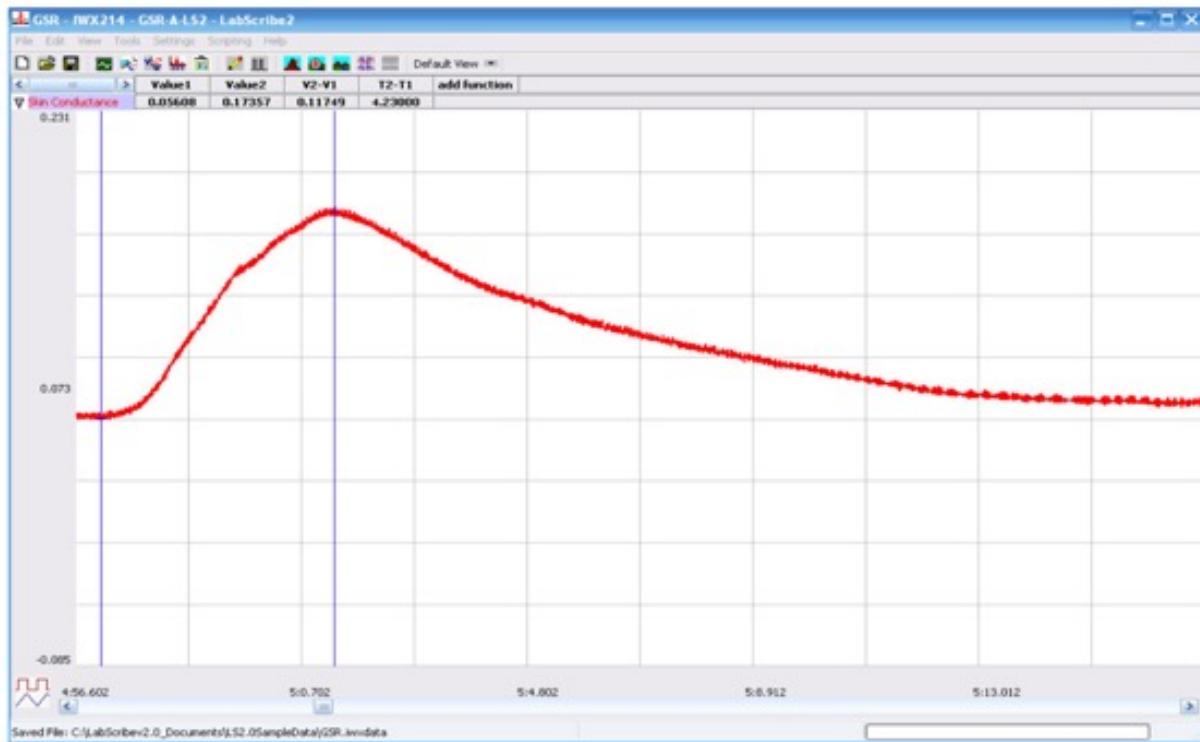
Electrodermal Activity

- Amplitude
 - Difference between tonic and Skin conductance

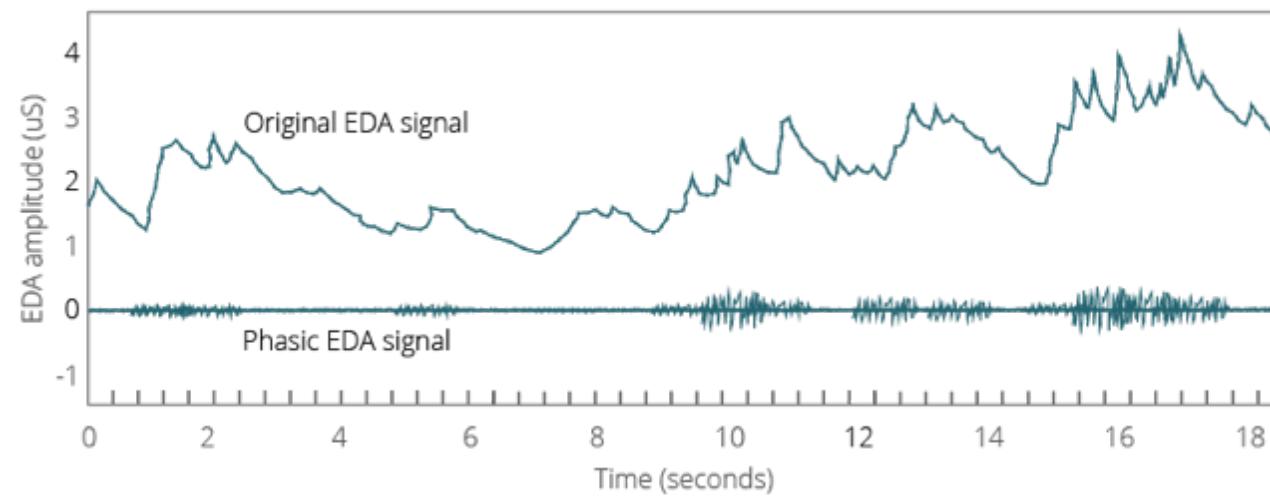


Electrodermal Activity

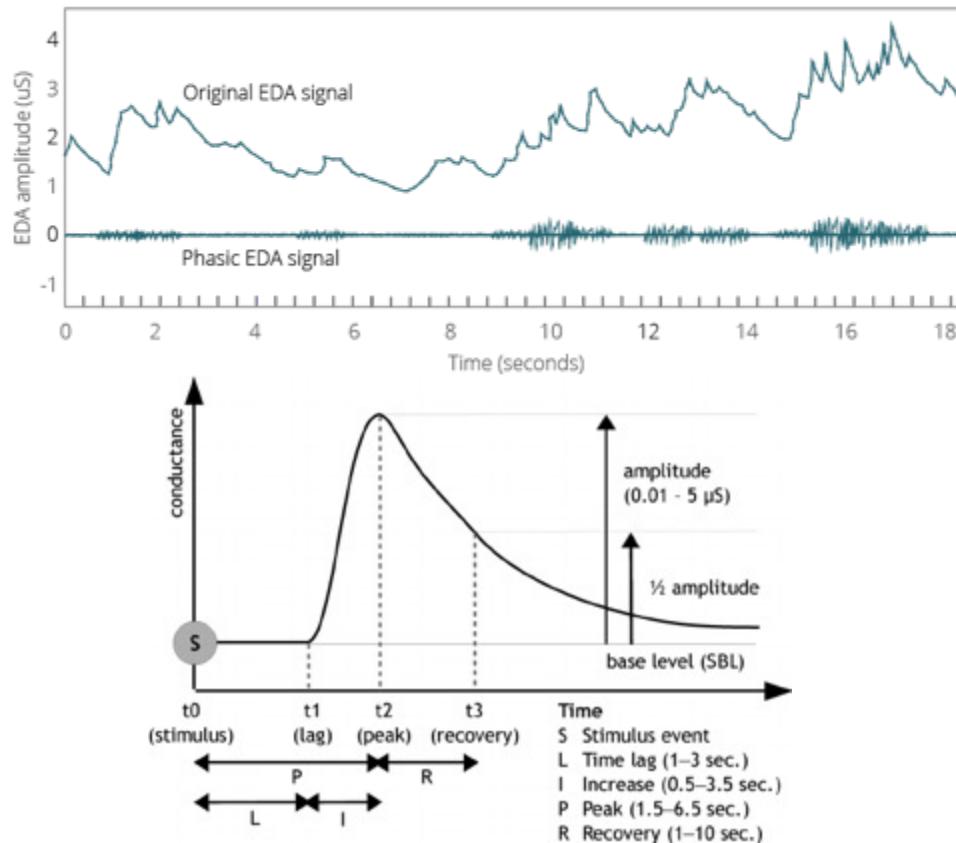
- Amplitude
 - Difference between tonic and Skin conductance
- Tonic 7.5 microS
- Amplitude 2.12 microS
- Arousal
 - Local Maxima



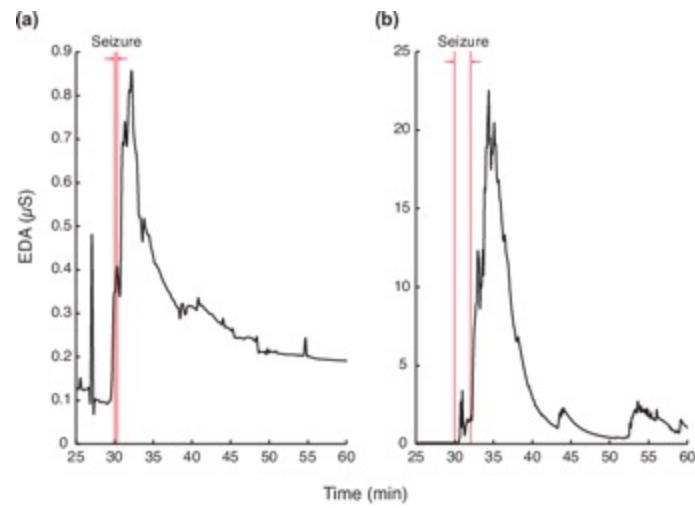
Electrodermal Activity



Electrodermal Activity

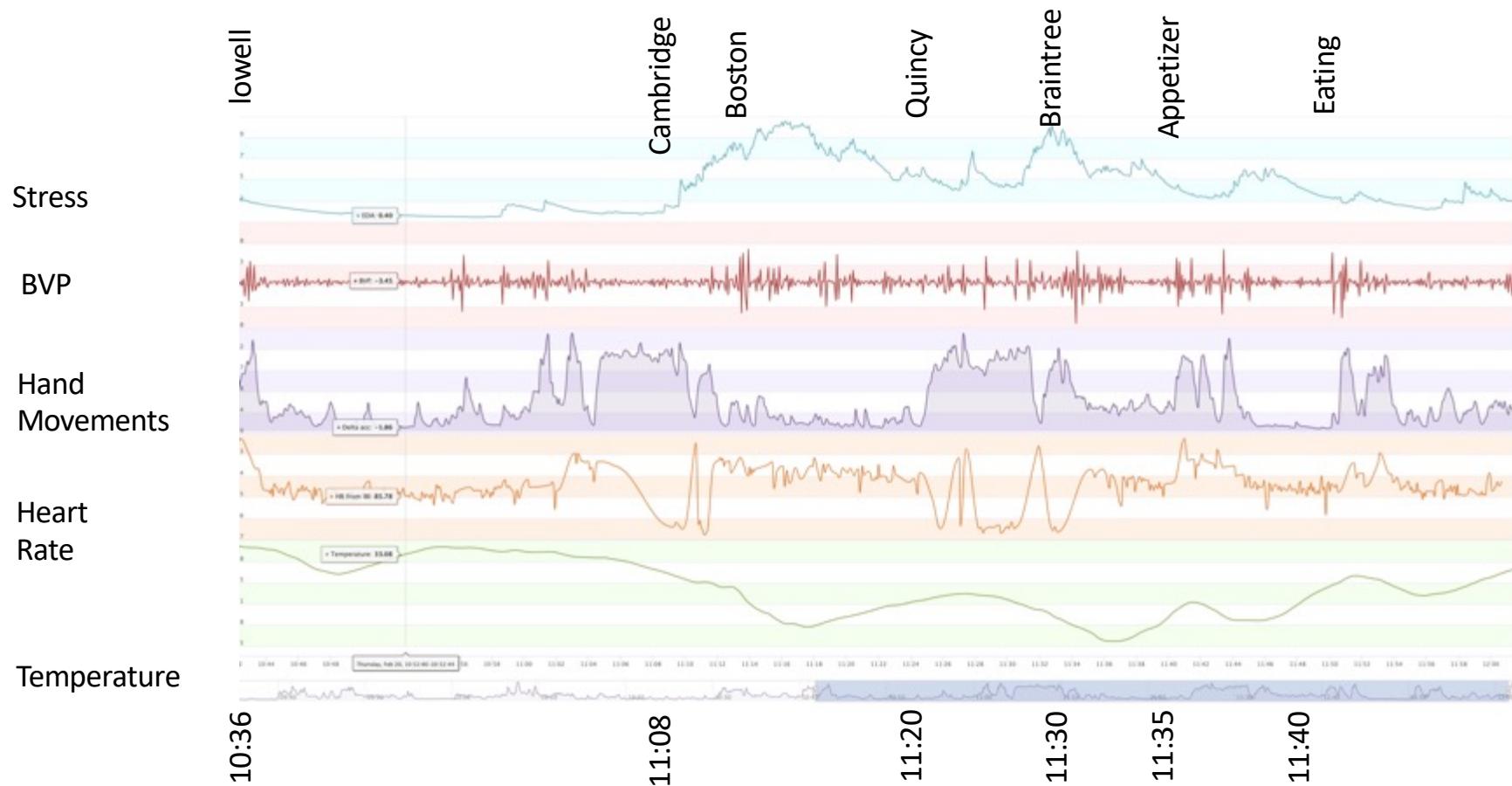


Electrodermal Activity (Usage)

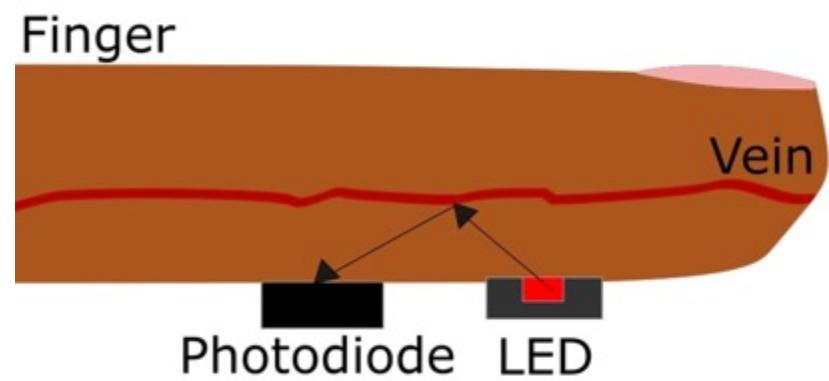
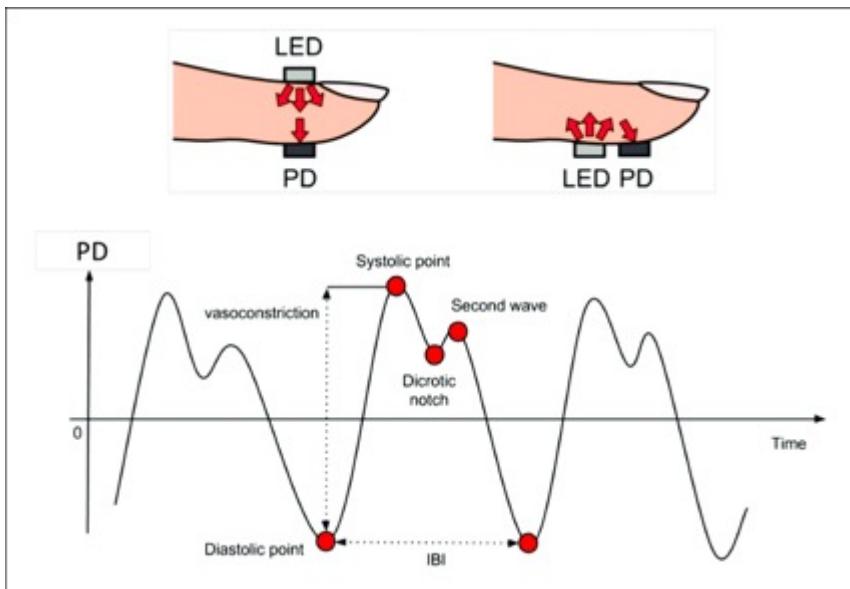


Epileptic Seizure Detection Real-time

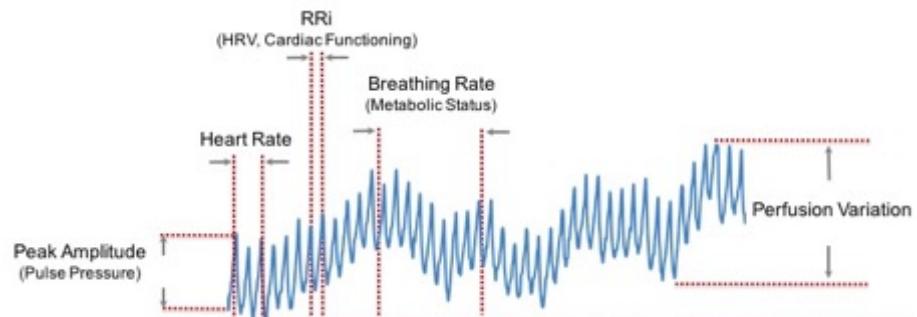
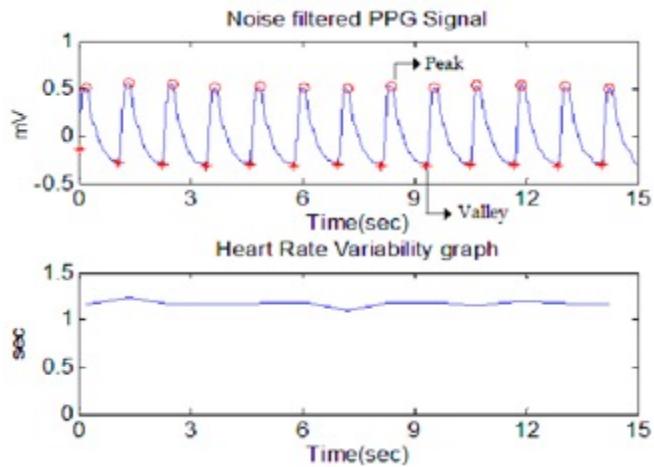
My Stress Assessment to join a Party in BrainTree



PPG : Photoplethysmography



PPG : Photoplethysmography



Empatica E4

E4 Sensors



PPG Sensor

Photoplethysmography Sensor - Measures Blood Volume Pulse (BVP), from which heart rate, heart rate variability (HRV), and other cardiovascular features may be derived



3-axis Accelerometer

Captures motion-based activity



Event Mark Button

Tags events and correlate them with physiological signals



EDA Sensor (GSR Sensor)

Electrodermal Activity Sensor
- Used to measure sympathetic nervous system arousal and to derive features related to stress, engagement, and excitement.



Infrared Thermopile

Reads peripheral skin temperature



Internal Real-Time Clock

Temporal resolution up to 0.2 seconds in streaming mode

Empatica E4



Empatica E4 Data

- ACC.csv - Data from 3-axis accelerometer sensor in the range [-2g, 2g]. (sampled at 32 Hz)
- BVP.csv - Data from photoplethysmograph (PPG). (sampled at 64 Hz)
- EDA.csv - Data from the electrodermal activity sensor in μS . (sampled at 4 Hz)
- IBI.csv - Inter beat intervals. (intermittent output with 1/64 second resolution)
- TEMP.csv - Data from temperature sensor expressed degrees on the Celsius ($^{\circ}\text{C}$) scale (sampled at 4 Hz)
- HR.csv - This file contains the average heart rate values, computed in spans of 10 seconds
More details here: [E4 data - HR.csv explanation](#)
- info.txt - Descriptions of the files

Empatica E4 Usage

- Sleep monitoring
- Stress monitoring
- Emotional arousal
- Stress level of dementia
- Memory loss detection
- Autonomic Dysfunction and Autism
- Epileptic seizure monitoring

COMP.5650/4150.201
Internet of Things
Lecture 8.5:
Electroencephalogram (EEG)

Instructor
Mohammad Arif UI Alam

Why Measure the EEG ?

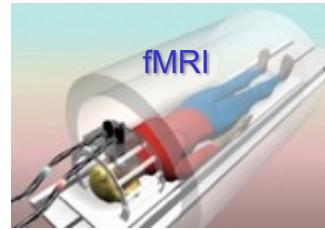
The greatest advantage of EEG is its **temporal resolution**. EEG can determine the relative strengths and positions of electrical activity in different brain regions.

According to R. Bickford (1987) research and clinical applications of the EEG in humans and animals are used to:

- (1)monitor alertness, coma and brain death;
- (2)locate areas of damage following head injury, stroke, tumor, etc.;
- (3)test afferent pathways (by evoked potentials);
- (4)monitor cognitive engagement (alpha rhythm);
- (5)produce biofeedback situations, alpha, etc.;
- (6)control anesthesia depth (“servo anesthesia”);
- (7)investigate epilepsy and locate seizure origin;
- (8)test epilepsy drug effects;
- (9)assist in experimental cortical excision of epileptic focus;
- (10)monitor human and animal brain development;
- (11)test drugs for convulsive effects;
R.D. Bickford. Electroencephalography. In: Adelman G. ed. *Encyclopedia of Neuroscience*, 371-3, 1987.
M. Teplan, Fundamental of EEG Measurement, In: *Measurement Science Review*, 2, 2002.
- (12)investigate sleep disorder and physiology.

R.D. Bickford. Electroencephalography. In: Adelman G. ed. *Encyclopedia of Neuroscience*, 371-3, 1987.
M. Teplan, Fundamental of EEG Measurement, In: *Measurement Science Review*, 2, 2002.

Current Neuroimaging Modalities



Functional magnetic resonance imaging

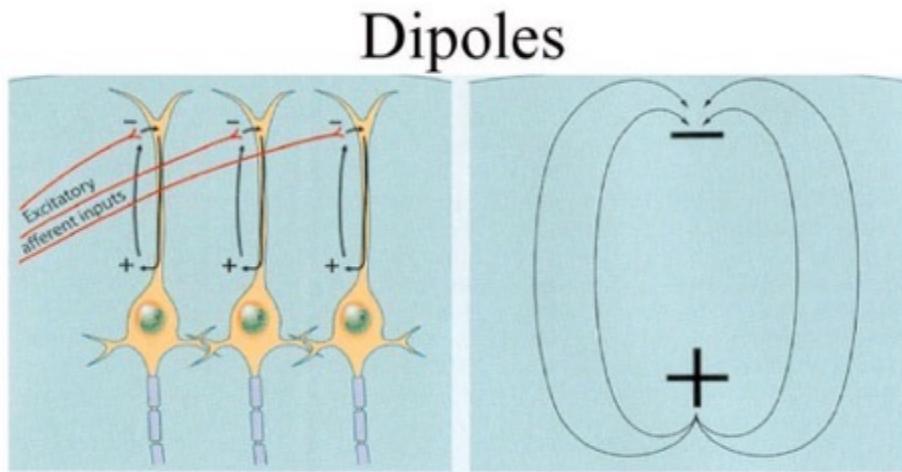


Positron emission tomography



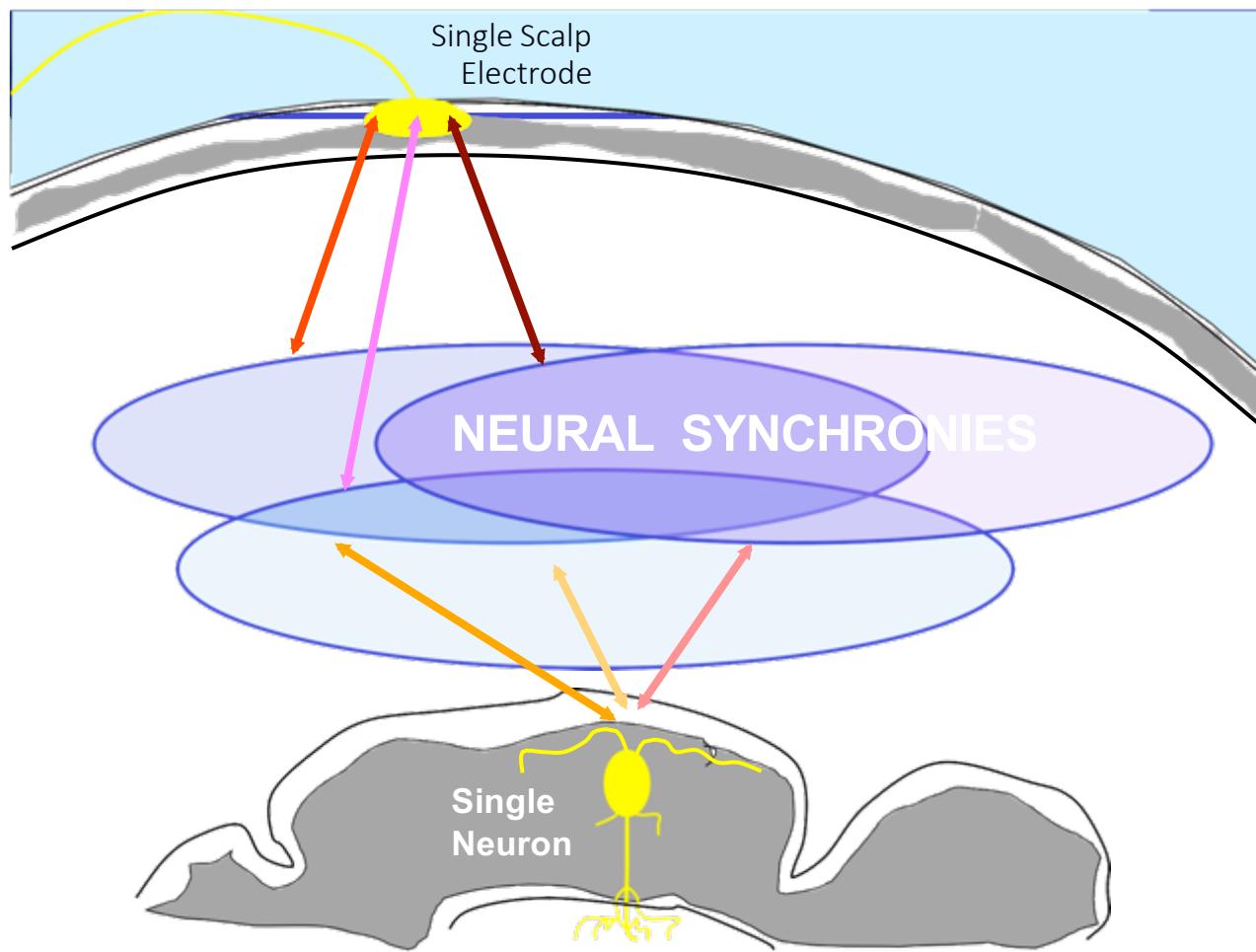
Magnetoencephalography

- In all modalities but EEG, the sensors are heavy.
- EEG is the only modality that does not require the head/body to be fixed.
- EEG might enable the monitoring of the brain functions of unconstrained participants performing normal tasks in the workplace and home.



- When neurons are activated, local currents are produced.
- EEG measures the current that flow during the excitations of the dendrites of many pyramidal neurons in the cerebral cortex.
- Potential differences are caused by summed postsynaptic potentials from pyramidal cells that create dipoles between soma and apical dendrites.
- **Necessary conditions: Aligned neurons and synchronous activity.**

Figure is from Gazzaniga et al., *Cognitive Neuroscience: The biology of the mind* Norton and Company, 2009.



EEG Acquisition

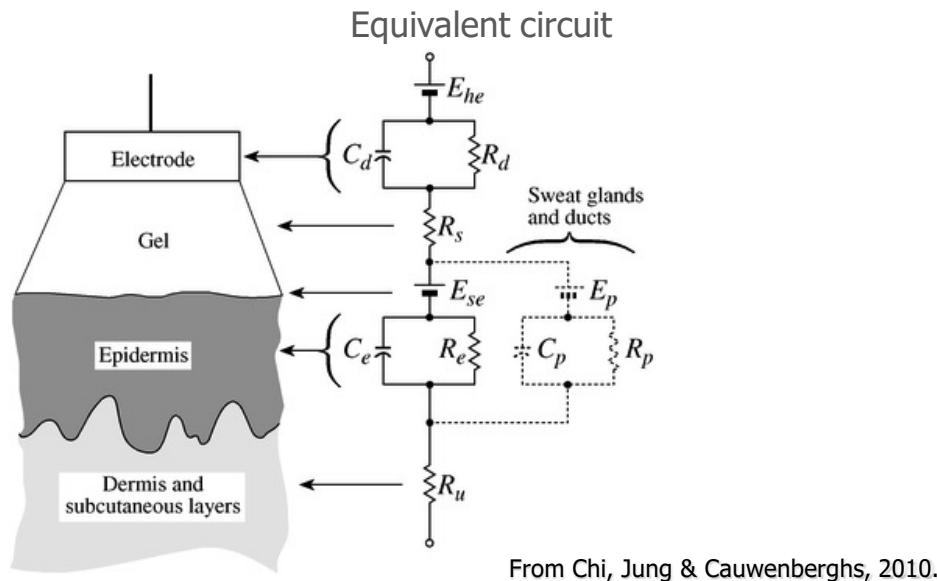
- Electrode caps, conductive jelly, ruler, injection and aid for disinfection.
- EEG amplifier unit, PC/laptop



EEG Electrodes

In common applications, EEG signals are measured by an electrode with electrolyte gel placed directly on the skin.

The coupling between skin and electrode can be described as a layered conductive and capacitive structure, with series combinations of parallel RC elements.



From Chi, Jung & Cauwenberghs, 2010.

Wearable EEG Devices



(a)

Emotiv



(b)

NeuroSky



(c)

Zeo



(d)

Starlab



(e)

EmSense



(f)

nia Game Controller



Mindo 4

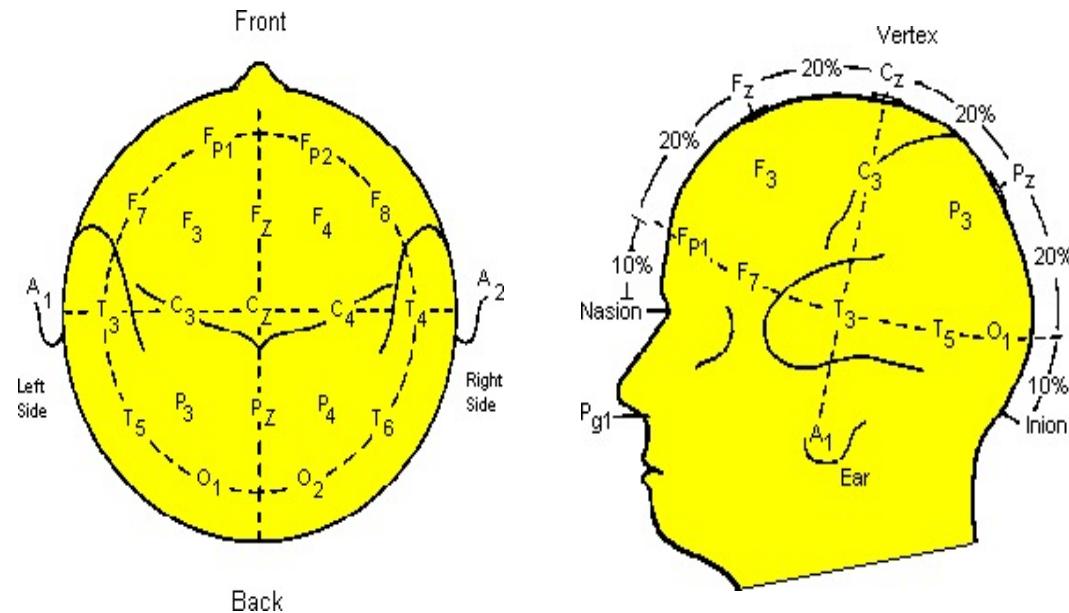


(h)

Mindo 16

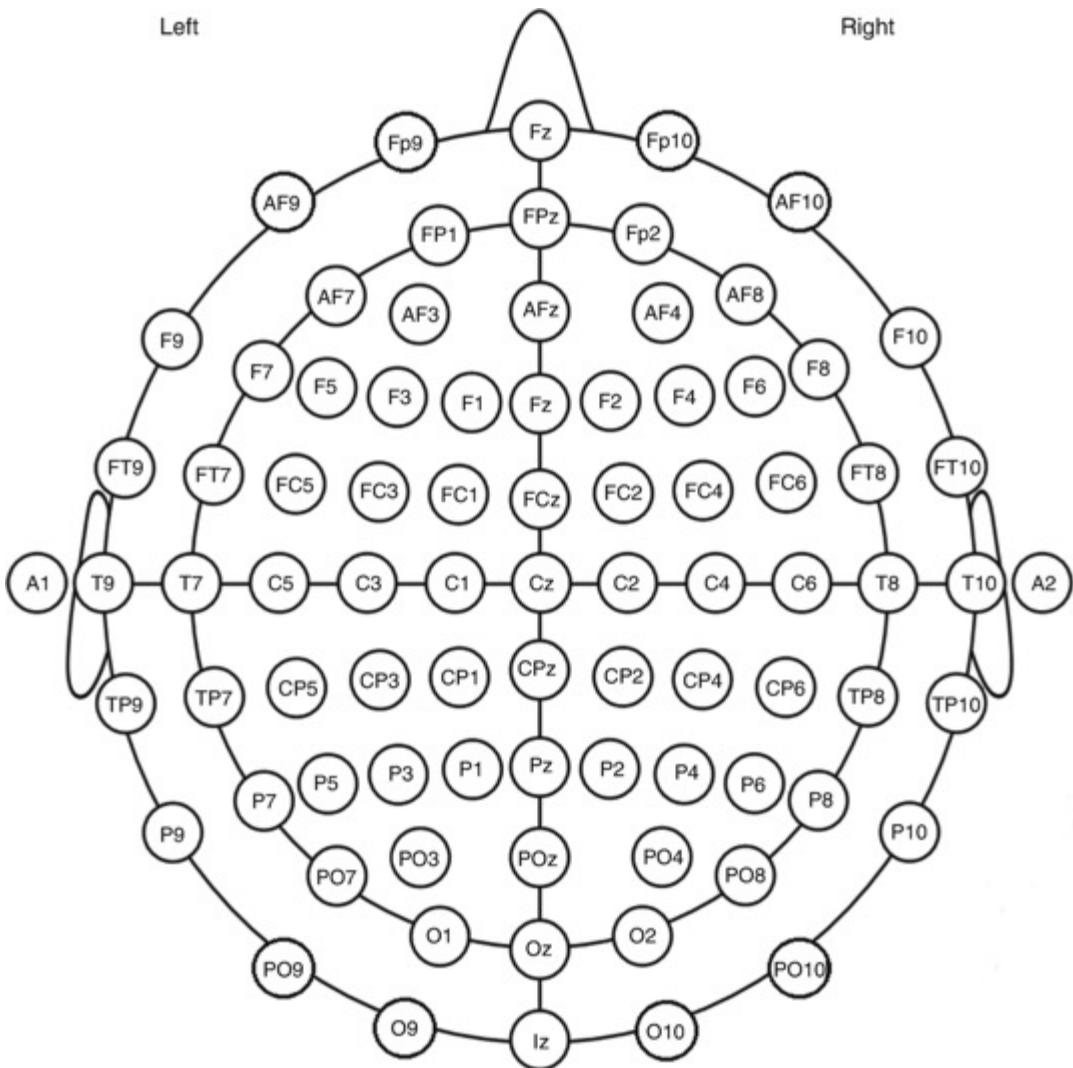
EEG Montage

International 10-20 system



Left

Right



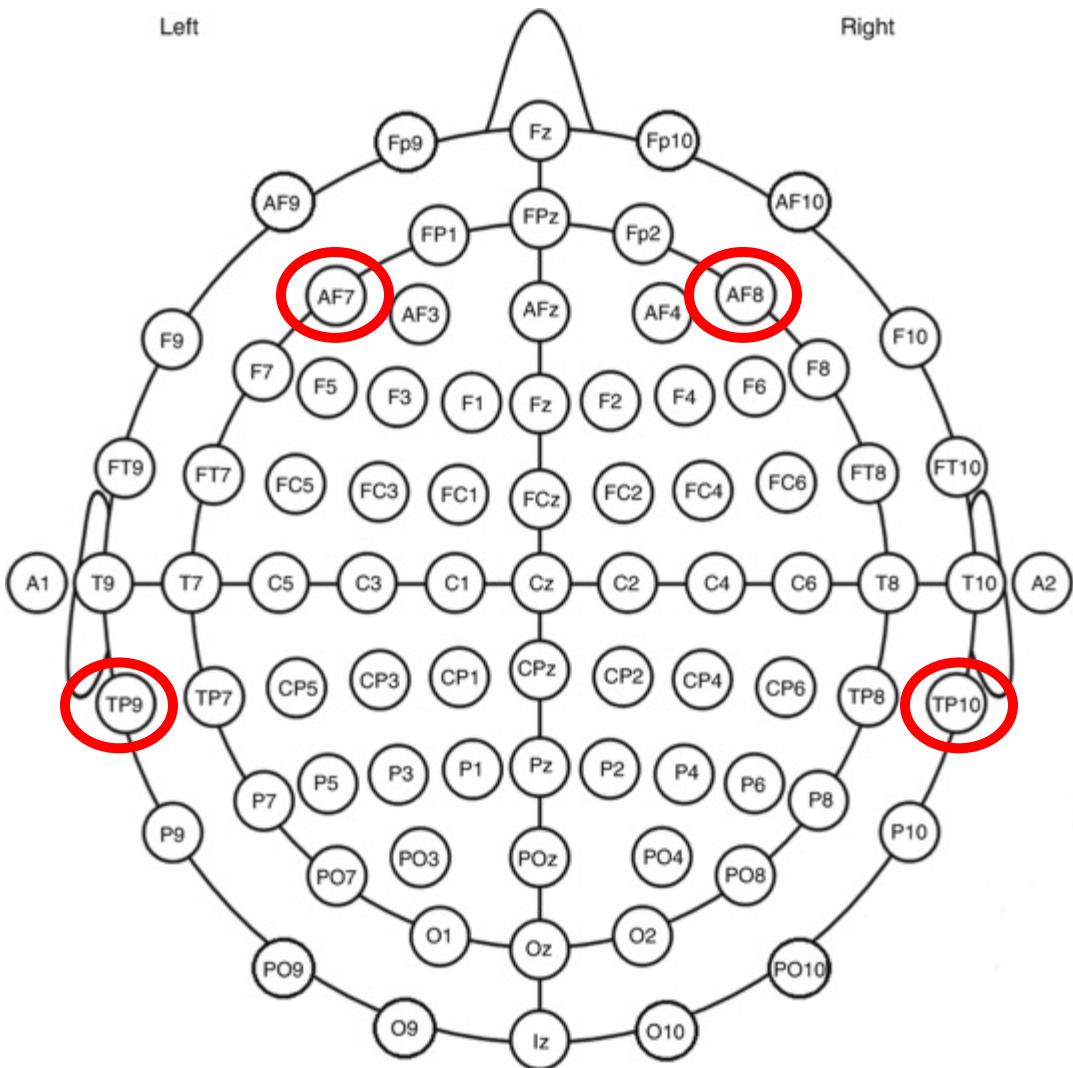
MUSE EEG



TP9, AF7, AF8, TP10

Left

Right



MUSE EEG



TP9, AF7, AF8, TP10

What EEG/ERPs Can and Cannot Tell Us About Brain Functions

CAN

- Precise timing of neural activity
- Sequence of mental operations

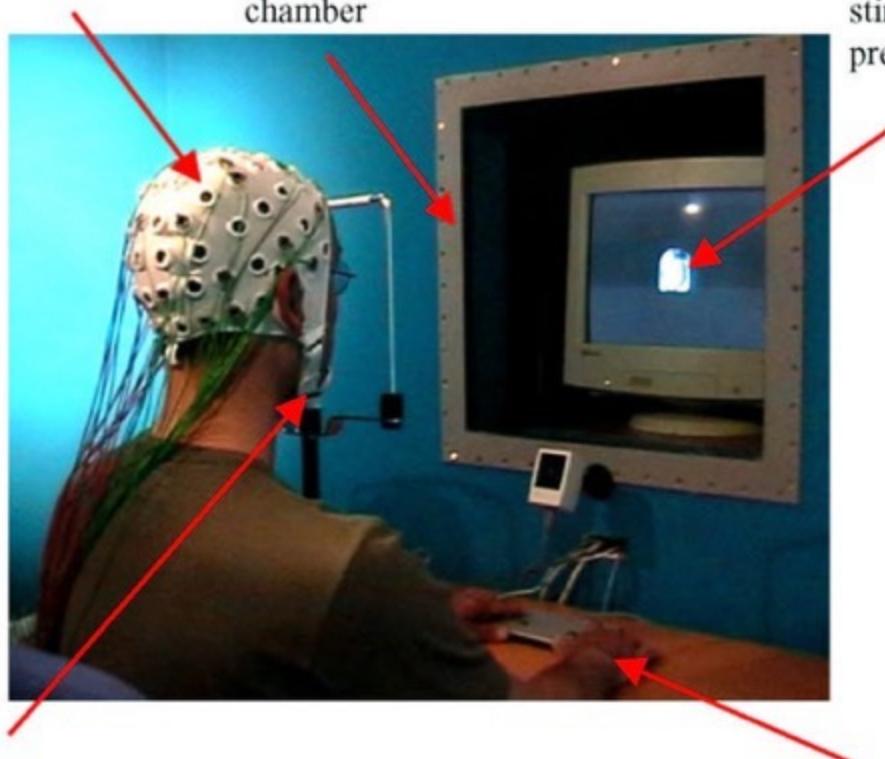
CANNOT

- Precise brain location of neural activity

Head cap with inserted
Ag/AgCl electrodes

Electromagnetically
shielded recording
chamber

Visual
stimulus
presentation

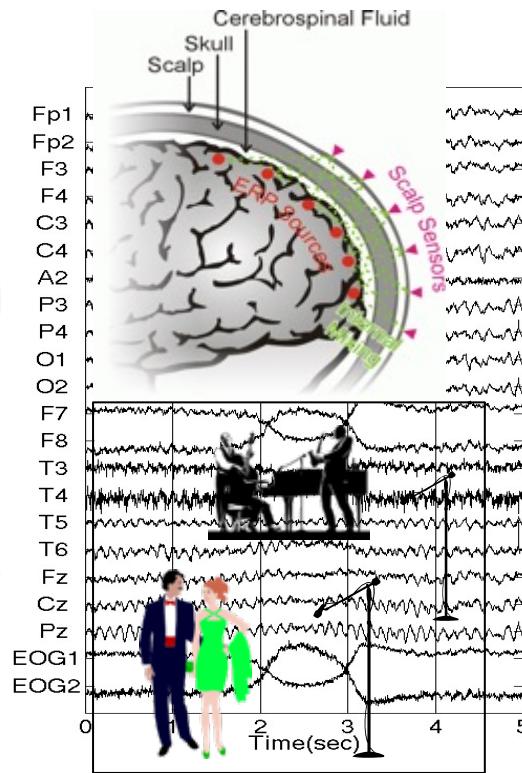


Fixed chin rest

Keypress response pad

Challenges of EEG Analysis

- Pervasive artifacts
- EEG recordings are mixtures of all brain activities arising from different networks
- Response variability
- Inverse problem
- others



Sensor Values

- Measurement Voltage fluctuations of brain activity



- Frequency of EEG signals range from 0.01 Hz to around 100 Hz
- Four Channels
- Four basic types of fluctuations/frequency bands

Table 1. Basic brain wave with their frequency.

Band Name	Frequency (Hz)	Interpretation
Delta	<4	Deep sleep
Theta	4–8	Relaxed state and meditation
Alpha	8–13	Relaxed state of consciousness
Beta	13–30	active thinking

Artifacts Removal

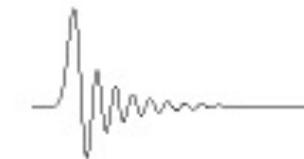
- Ocular Artifacts
 - Blinking Eyes
- Muscle Artifacts
 - Muscle movements
 - Any muscle of body
- Cardiac Artifacts
 - Electrodes placed near blood vessels
- Extrinsic Artifacts
 - Any external artifacts such as sweating, wet, hair gel etc

Artifacts Removal (Filters)

- Infinite Impulse Response (IIR)
 - Cardiac Artifacts
- Moving Average
 - Smoothen any artifacts
- Band Pass filter
 - High band pass filter
 - Ocular
 - Extrinsic
 - Low band pass filter
 - Muscle movements



Infinite impulse response filter can have an impulse response that is infinite



But for most practical applications, the impulse response will die away after some time

© BORES Signal Processing

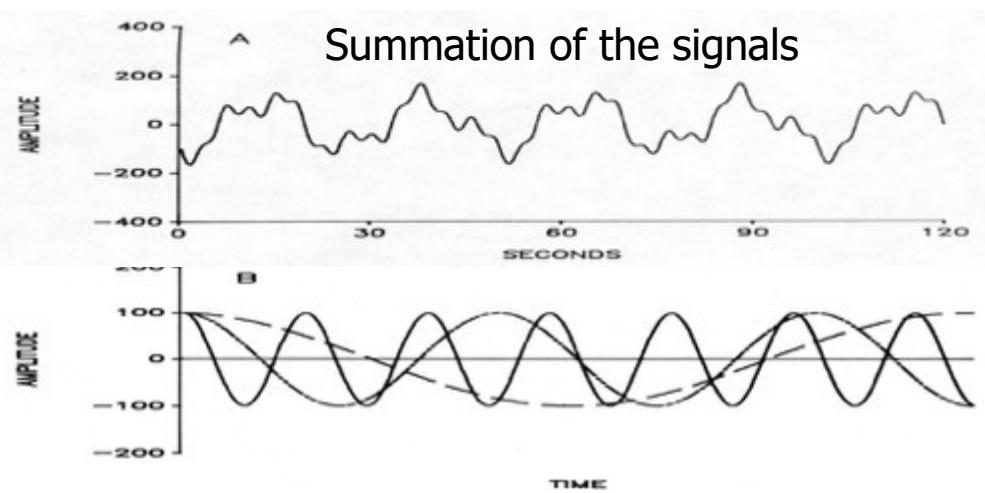
Feature Extraction

- Time Domain and Frequency Domain
- Time domain
 - Average, mean, median, maximum etc.

Frequency-domain Analysis of the EEG



- Joseph Fourier (1768-1830)
- Any complex time series can be broken down into a series of superimposed sinusoids with different frequencies.



Fourier Transform

- **Advantage:**

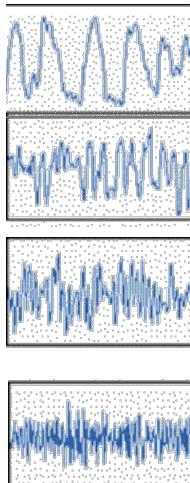
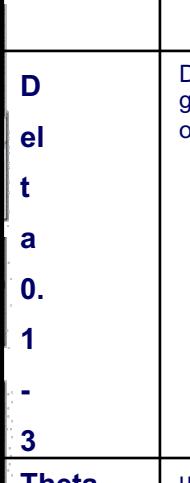
For many signals, Fourier analysis is extremely useful because the signal's frequency content is of great importance.

- **Disadvantage:**

Fourier analysis has a serious drawback.

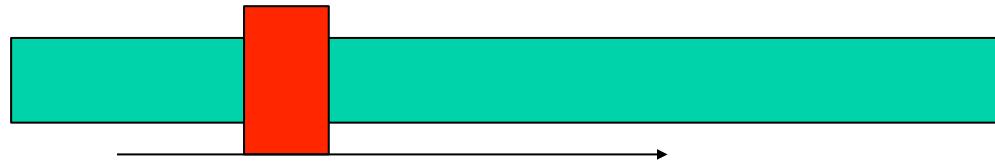
In transforming to the frequency domain, time information is lost.

Frequency-domain Analysis of the EEG

EEG Bands (Hz)	Distribution	Subjective feeling	Associated tasks & behaviors	Physiological correlates
 Delta 0.1 - 3	Distribution: generally broad or diffused	deep, dreamless sleep, non-REM sleep, unconscious	lethargic, not moving, not attentive	not moving, low-level of arousal
 Theta 4-8	usually regional, may involve many lobes	intuitive, creative, recall, fantasy, imagery, creative, dreamlike, drowsy	creative, intuitive; distracted, unfocused	healing, integration of mind/body
 Alpha 8-12	regional, usually involves entire lobe	relaxed, not agitated, but not drowsy	meditation, no action	relaxed, he

Time-Frequency Analysis of the EEG

- We often apply a 'window' to the data.
- This simply means taking the amount we want from the data stream
- ie



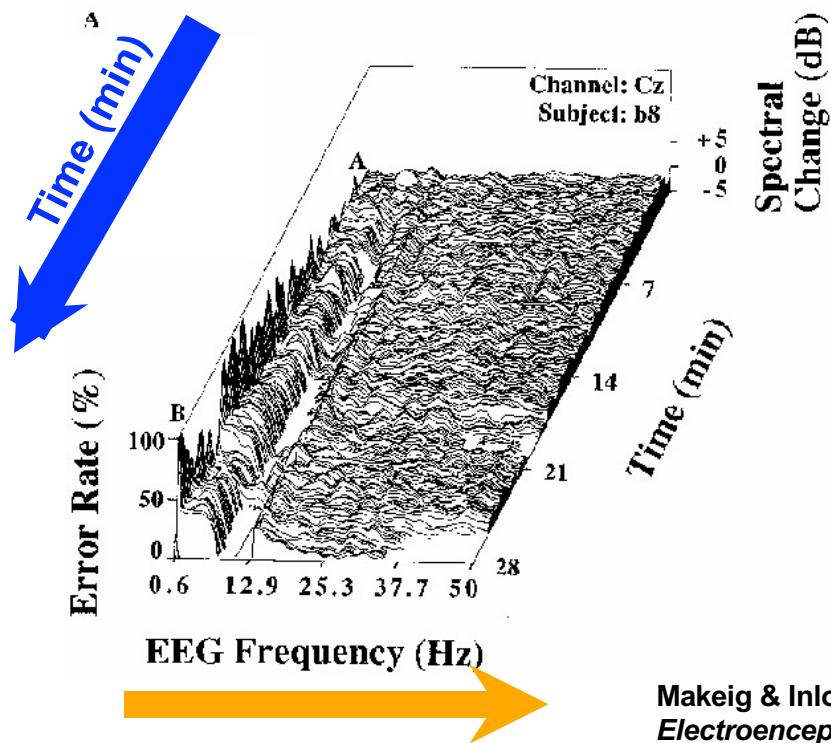
The window is moved along the data; we perform the FFT on this windowed data

Time-Frequency Analysis of the EEG

- Four channels of EEG signal
 - double[][] rawEEG
 - rawEEG[][][0-3]
 - TP9, AF7, AF8, TP10
 - Extract Bands
 - Suppose you want to extract 6 bands features for each channel
 - BAND = {1, 4, 8, 12, 18, 30, 45};
 - BandPassFilter(rawEEG, SampleRate, WindowSize, Num_Band)
 - returns 24 (6 x 4) features
- https://github.com/zhijiee/MuseEEGClassifier/blob/master/app/src/main/java/controllers/SVMController/SVM_Helper.java

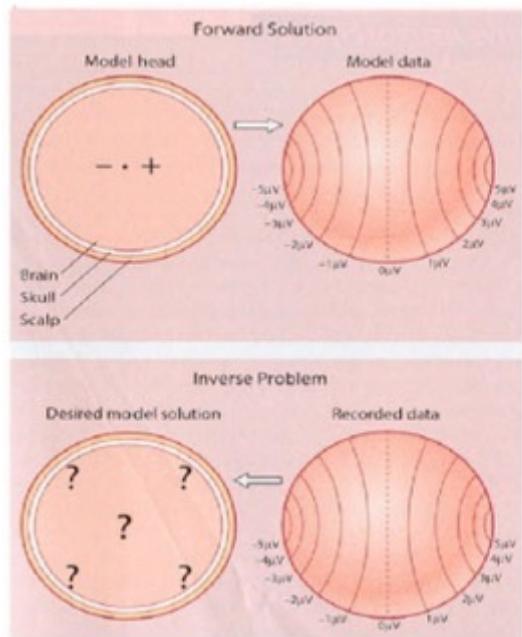
<https://imotions.com/blog/eeg/>

Time-Frequency Analysis of the EEG



Makeig & Inlow (1993)
Electroenceph Clin Neuro

Forward Solution and Inverse Problem



A single pattern of neural activity will produce a unique scalp map

BUT ... A single scalp map could have been produced by an infinite number of patterns of neural activity

Figure is copied from Gazzaniga et al., *Cognitive Neuroscience: The biology of the mind* Norton and Company, 2009.

Thank You