| A black and white logo  Description automatically generated with low confidence | INTERNATIONAL TELECOMMUNICATION UNION  **TELECOMMUNICATION** **STANDARDIZATION SECTOR**  STUDY PERIOD 2022-2024 | | **Focus Group on AI Native Networks** | |
| --- | --- | --- | --- | --- |
| **AINN-I-xx** | |
| **Original: English** | |
| **Question(s):** | | N/A | Virtual, TBD 2024 | |
| **INPUT DOCUMENT** | | | | |
| **Source:** | | *SRM University - Andhra Pradesh* | | |
| **Title:** | | *HeaxCore Team –* Report on ITU WTSA Hackathon 2024 *– AuRa: AuratRaksha* | | |
| **Contact:** | | Mr. Gopi Chand Janjanam | | E-mail: gopichand\_janjanam@srmap.edu.in |
| **Contact:** | | Mr. Rohith Kumar Ankam | | E-mail: rohithkumar\_ankam@srmap.edu.in |
| **Contact:** | | Mr. Eswar Panchakarla | | E-mail: eswar\_panchakarla@srmap.edu.in |

| **Abstract:** | This document contains the submission of a report for *HeaxCore* Team towards ITU WTSA Hackathon 2024 for use case “AuRa: Alert, Unify, Respond, Assist towards AuratRaksha”. |
| --- | --- |

## **Use case introduction: “AuRa: Alert, Unify, Respond, Assist - AuratRaksha”**

*Ensuring Safety and Immediate Assistance for Women in UnSafe Situation or Constrained Situations*

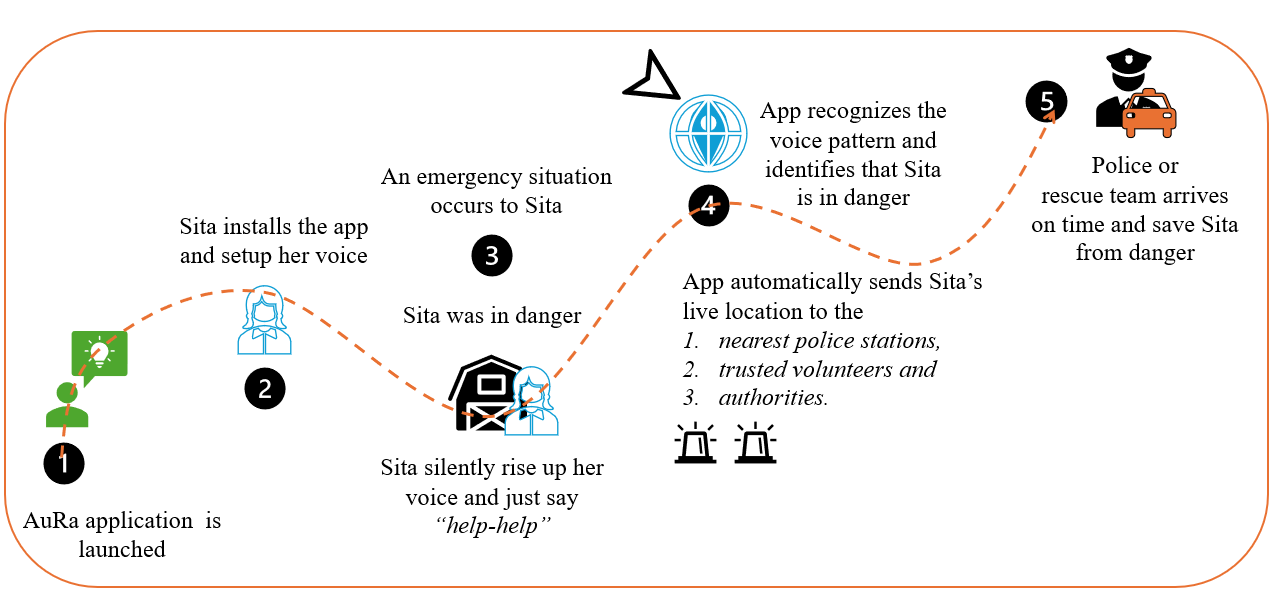
According to the [Report](https://www.indiatoday.in/law/story/crimes-against-women-children-national-crimes-record-bureau-ncrb-data-2471938-2023-12-04), India reported a total of 58,24,946 cognizable crimes in the year 2022. Millions of women worldwide face violence, harassment, or threats in supposedly safe spaces, such as homes, offices, hospitals, public transportation, or while walking alone at night. In these situations, reaching out for help via phone calls or messages can exacerbate the danger. The fear of vengeance or escalation prevents them from seeking immediate assistance. Imagine a lady Sita, or someone close to you is in a situation where the lady is not safe. It could be at home, office, or hospital. Sita might feel it was a safe place. But unfortunately what if she is not safe? In the privacy of her own space, someone she knows or someone she doesn't know tries to harm her?

Harmful scenarios or other dangers can persist anywhere which includes in the streets, public places, walking alone at night, traveling in train or bus, trapped in an office or home and other places having a fear and uncertainty. Sita wants to cry out for help, but the threat is too close. Reaching for her phone, making a call, or even sending a message could put her in greater danger.

*How can Sita escape from a particular harmful situation and back to a normal position? Can you help Sita?*

| Scenario Story:  A student named Gopi from Andhra Pradesh, while he was browsing on the internet, a news report caught his eye: "India Reports 58,24,946 Cognizable Crimes". He was sad to hear that millions of women worldwide face violence, harassment, or threats in supposedly safe spaces. One line haunted him: "Many victims remain silent, fearing escalation or vengeance." Gopi thought of his close ones. Could they be next?  Gopi thinks, "What if I could create an app that lets women like Sita call for help silently?". His vision took shape: *an AI-driven safety system*, harnessing 5G/6G networks, to protect women wherever they were. With a simple, silent voice command, sita can trigger an alert. Sita’s location and live audio feed would be shared with nearest police officers, trusted volunteers and authorities. Real-time tracking and communication will ensure a swift help. Gopi's team partnered with local organizations, training volunteers to respond to alerts. They collaborated with law enforcement, streamlining the process for rapid assistance. The app launched, and the response overwhelmed Gopi. Women from across India downloaded the app, sharing stories of empowerment. Sita felt safer knowing she had a secret power in her pocket. One night, a message arrived: "Gopi, your app saved my life. I was trapped in my own home, but with the app AuRa, I whispered for help. The police arrived in minutes and I am safe. Thank you for hearing my silent cry."  Gopi’s work had made a difference. The app AuRa became a symbol of hope, a reminder that technology could be a powerful tool against violence. His story inspired others to join the fight against violence. AuRa app expanded globally, safeguarding women in every corner of the world. |
| --- |

Consider the scene map below:



*Outline of the Use-case (Each Phase indicates the Timeline of the Scenario)*

Phase 1: Sita is in an unsafe situation where some are attacking her. Sita has “*AuRa, an emergency rescue app*”. Sita installs the app and sets up her voice commands successfully.

Phase 2: Unfortunately there is no one to help sita.

Phase 3: Sita “ *raises her voice seeking help*”. The app which is already having Sita’s voice inputs. It automatically recognizes the voice pattern and identifies that sita is in danger. App automatically sends Sita’s live location to the nearest police stations, trusted volunteers and authorities.

Phase 4: “*Police or rescue team arrives within minutes*” identifies the situation of Sita, and immediately the police men or rescue team saves Sita from danger.

Phase 5: Sita comes out safely with the help of police officers or the rescue team, who arrives first to secure Sita from an emergency situation.

## 

## 

## **Use case requirements**

**Requirement-1**: It is critical to have 5G/6G technology to ensure ultra-low latency for voice, audio, and data transmission. This is critical for real-time communication of audio alerts and GPS data to emergency responders, ensuring swift action during emergencies.

**Requirement-2:** *High Precision AI Detection(AI/ML models)—*The application must integrate high precision AI algorithms capable of accurately detecting the victim's voice and assessing the situation(emotion/emergency detection)

**Requirement-3:** *Prioritized Resource Allocation Based on Location*—It is critical to prioritize resource allocation based on the user's location during an emergency. This involves deploying and managing RAN slices specifically optimized for emergencies to ensure that resources are available where they are most needed.

**Requirement-4:** *RAN Slice Management for Uninterrupted Communication—*It is critical to have and must support RAN slice management, specifically selecting URLLC (Ultra-Reliable Low Latency Communication) slices to facilitate uninterrupted communication during emergencies. This ensures that voice commands and data transmissions are processed with minimal delay and maximum reliability.

**Requirement-5:** *Fake Emergency Detection Mechanism*—To prevent misuse of the system, a mechanism must be implemented to detect potential fake emergencies. This could involve analyzing patterns of voice commands over time, assessing user behavior, and employing machine learning models to distinguish between genuine distress signals and false alarms.

*Our use-case has the potential to address the below United Nations Sustainable development Goals (SDG’s)*

UN Goals

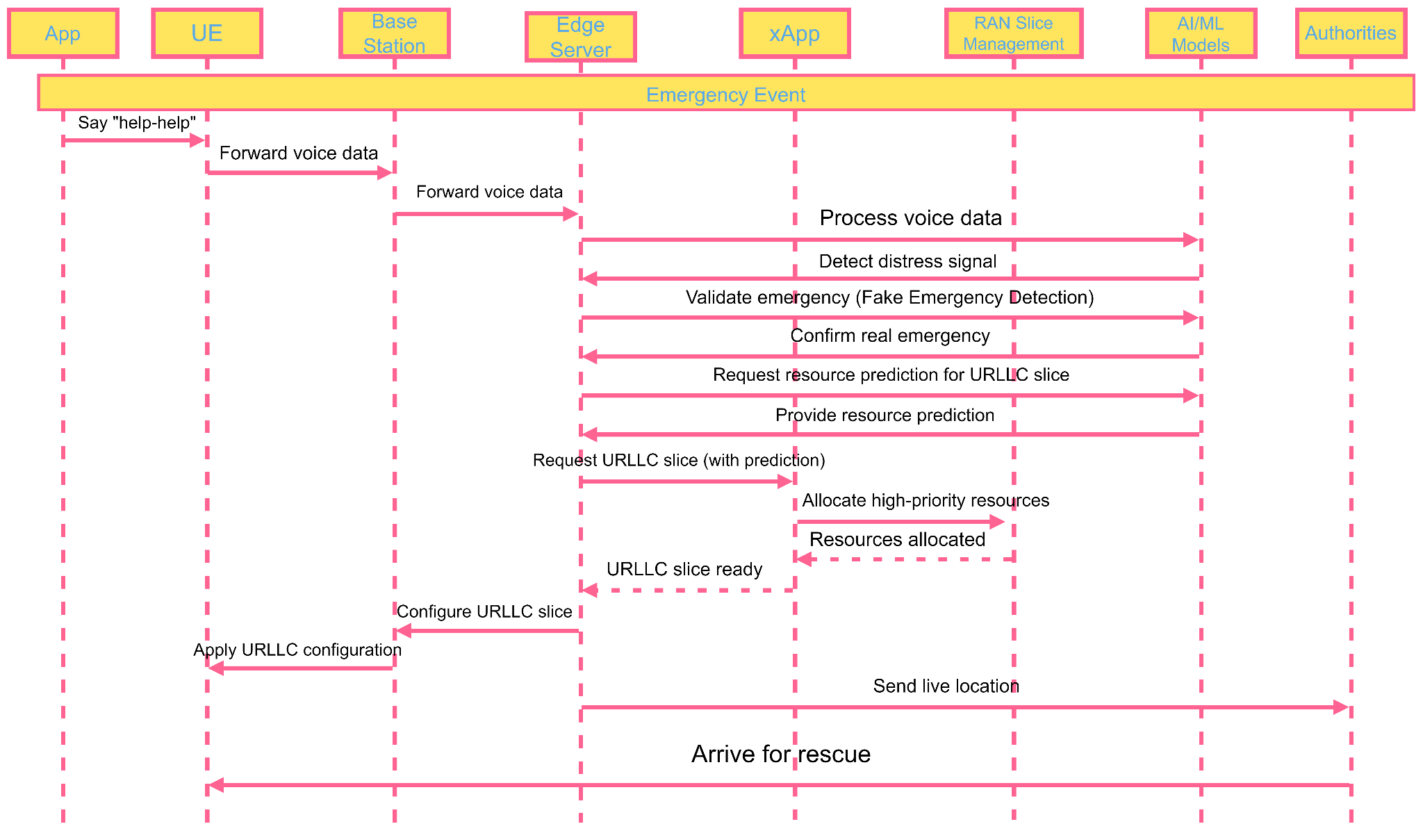
1. SDG-5: Gender Equality
2. SDG-3: Good Health and well being
3. SDG-16: Peace, Justice, and Strong Institutions

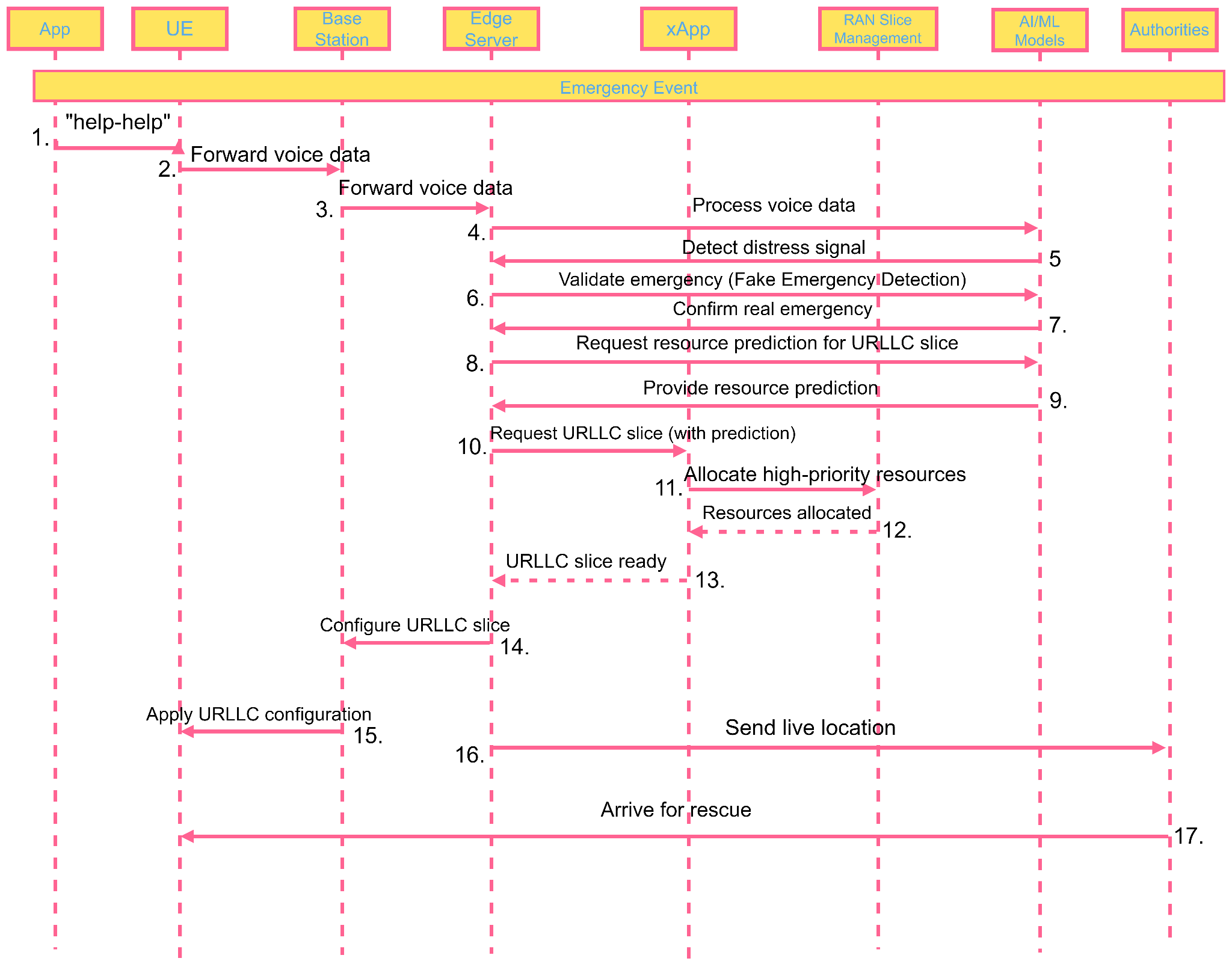
Justify UN Goals Selection:

* SDG - 5: Eliminate all forms of violence against all women and girls in the public and private sphere and other types of exploitation.
* SDG-3: Our use case helps to reduce the stress and anxiety for women through enhanced safety.
* Goal 16: Peace, Justice, and Strong Institutions – Promoting a safe environment is directly linked to reducing violence and ensuring justice. By building systems that can respond to emergencies.

## **SPS1: pipeline design**

* AI /ML Concept used is emotion/emergency detection, real-time data processing, anomaly detection and live racking detection.
* In Relation with ITU Y.3172 – for submission.  
  xml link for the above sequence is here: Aura: [sequence xml file](https://viewer.diagrams.net/?tags=%7B%7D&lightbox=1&highlight=0000ff&edit=_blank&layers=1&nav=1&title=Sequence%20Diagram.drawio#Uhttps%3A%2F%2Fdrive.google.com%2Fuc%3Fid%3D1z_EJdCcZEGCR6vu0i--zqC5VxdPNCPM0%26export%3Ddownload)





Yaml Code  
tosca\_definitions\_version: tosca\_simple\_yaml\_1\_3

description: Women's Safety Emergency App Service Template with AI-based threat detection, and enhanced RAN slice management.

node\_types:

App:

derived\_from: tosca.nodes.SoftwareComponent

properties:

app\_version:

type: string

safety\_features:

type: list

entry\_schema:

type: string

UserDevice:

derived\_from: tosca.nodes.MobileDevice

properties:

device\_type:

type: string

os\_version:

type: string

BaseStation:

derived\_from: tosca.nodes.Network.BaseStation

properties:

coverage\_radius:

type: float

unit: km

EdgeServer:

derived\_from: tosca.nodes.Compute

properties:

processing\_power:

type: float

unit: GFLOPS

SafetyXApp:

derived\_from: tosca.nodes.SoftwareComponent

properties:

api\_version:

type: string

supported\_emergency\_types:

type: list

entry\_schema:

type: string

enable\_silent\_mode:

type: boolean

predictive\_threat\_detection:

type: boolean

RANSliceManagement:

derived\_from: tosca.nodes.SoftwareComponent

properties:

max\_priority\_slices:

type: integer

dynamic\_scaling:

type: boolean

SafetyAIModels:

derived\_from: tosca.nodes.SoftwareComponent

properties:

model\_version:

type: string

supported\_languages:

type: list

entry\_schema:

type: string

geofencing\_enabled:

type: boolean

EmergencyAuthorities:

derived\_from: tosca.nodes.SoftwareComponent

properties:

jurisdiction:

type: string

emergency\_services:

type: list

entry\_schema:

type: string

alert\_confirmation:

type: string

topology\_template:

node\_templates:

app:

type: App

properties:

safety\_features:

- voice\_activated

- location\_tracking

- fake\_call

requirements:

- host: user\_device

user\_device:

type: UserDevice

properties:

device\_type: "smartphone"

os\_version: "latest"

requirements:

- connection: base\_station

base\_station:

type: BaseStation

properties:

coverage\_radius: 5.0

requirements:

- connection: edge\_server

edge\_server:

type: EdgeServer

properties:

processing\_power: 1000.0

requirements:

- connection: safety\_xapp

safety\_xapp:

type: SafetyXApp

properties:

supported\_emergency\_types:

- physical\_threat

- medical\_emergency

- harassment

- predictive\_threat

enable\_silent\_mode: true

predictive\_threat\_detection: true

requirements:

- host: edge\_server

- dependency: safety\_ai\_models

ran\_slice\_management:

type: RANSliceManagement

properties:

max\_priority\_slices: 10

dynamic\_scaling: true

requirements:

- host: edge\_server

safety\_ai\_models:

type: SafetyAIModels

properties:

model\_version: "3.3.0"

supported\_languages:

- English

-

geofencing\_enabled: true

requirements:

- host: edge\_server

emergency\_authorities:

type: EmergencyAuthorities

properties:

jurisdiction: "local"

emergency\_services:

- police

- ambulance

alert\_confirmation: true

relationships:

app\_to\_device:

type: tosca.relationships.ConnectsTo

source: app

target: user\_device

device\_to\_base\_station:

type: tosca.relationships.ConnectsTo

source: user\_device

target: base\_station

base\_station\_to\_edge\_server:

type: tosca.relationships.ConnectsTo

source: base\_station

target: edge\_server

edge\_server\_to\_safety\_xapp:

type: tosca.relationships.ConnectsTo

source: edge\_server

target: safety\_xapp

safety\_xapp\_to\_ai\_models:

type: tosca.relationships.DependsOn

source: safety\_xapp

target: safety\_ai\_models

safety\_xapp\_to\_ran\_slice:

type: tosca.relationships.ConnectsTo

source: safety\_xapp

target: ran\_slice\_management

safety\_xapp\_to\_authorities:

type: tosca.relationships.ConnectsTo

source: safety\_xapp

target: emergency\_authorities

interfaces:

Standard:

operations:

create:

description: Set up the women's safety emergency system

delete:

description: Tear down the system

start:

description: Start the emergency response service

stop:

description: Stop the emergency response service

configure\_priority\_slice:

description: Configure high-priority network slice for emergency response

validate\_emergency:

description: Detect and validate emergency situations using AI models and geofencing

allocate\_emergency\_resources:

description: Allocate high-priority resources dynamically for emergency response

send\_location\_to\_authorities:

description: Send real-time location data to relevant emergency authorities

confirm\_alert\_receipt:

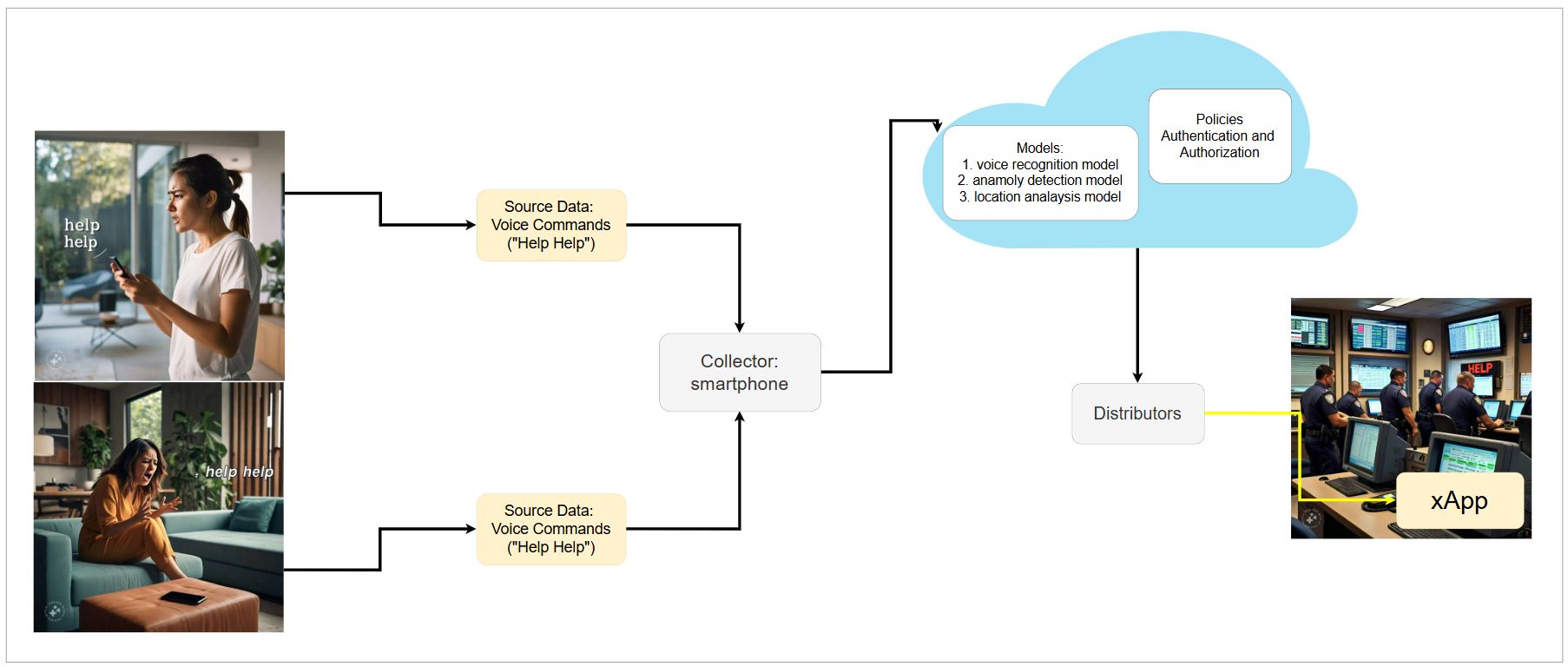
description: Confirm that emergency authorities have received the alert and responded

predictive\_threat\_alert:

description: Detect potential threats using predictive AI algorithms

**Requirements for this type of application?**

* SRC of data: UE
* Collector: edge server
* Models: emotion, emergency detection
* Policies: fake emergency detection
* Distributors: edge server
* Model inference Application (SINK): xApp



## **PS2: xApp design**

* The role of xApp?

In this xApp design main functionalities required for the woman safety are:

1. This xApp is designed for RAN slice management and slice selection.
2. Verifying the emergency to prevent false alarms.
3. Allocating a URLLC (Ultra-Reliable Low-Latency Communication) slice for emergency situations.
4. Prioritizing network resources for the emergency UE

*xApp code here:*

import json

import time

from datetime import datetime

class WomanSafetyXapp:

def \_\_init\_\_(self):

self.emergency\_detected = False

self.current\_slice = "default"

def process\_e2\_input(self, e2\_message):

message = json.loads(e2\_message)

if message['type'] == 'voice\_data':

self.process\_voice\_data(message['data'])

elif message['type'] == 'ue\_measurement':

self.process\_ue\_measurement(message['data'])

def process\_voice\_data(self, voice\_data):

emotion = self.detect\_emotion(voice\_data)

if self.detect\_emergency(emotion, voice\_data):

self.handle\_emergency()

def detect\_emotion(self, voice\_data):

# we will Simulate emotion detection

# In a real implementation, this use a machine learning model

print("Detecting emotion from voice data...")

return "distress" # Simulated result

def detect\_emergency(self, emotion, voice\_data):

# we will Simulate emergency detection

# In a real implementation, this would use more sophisticated logic

emergency\_keywords = ["help", "emergency", "danger"]

return emotion == "distress" and any(keyword in voice\_data.lower() for keyword in emergency\_keywords)

def handle\_emergency(self):

self.emergency\_detected = True

self.allocate\_urllc\_slice()

self.prioritize\_resources()

self.alert\_emergency\_services()

def allocate\_urllc\_slice(self):

print(f"Allocating URLLC slice at {datetime.now().strftime('%H:%M:%S')}")

self.current\_slice = "URLLC"

return self.generate\_e2\_output('slice\_allocation', {

'slice\_type': 'URLLC',

'action': 'allocate'

})

def prioritize\_resources(self):

print(f"Prioritizing resources for emergency UE at {datetime.now().strftime('%H:%M:%S')}")

return self.generate\_e2\_output('resource\_prioritization', {

'ue\_id': 'emergency\_ue',

'priority': 'high'

})

def alert\_emergency\_services(self):

print(f"Alerting emergency services with location data at {datetime.now().strftime('%H:%M:%S')}")

# In a real implementation, this would interface with emergency services systems

def process\_ue\_measurement(self, measurement\_data):

print(f"Processing UE measurement: {measurement\_data}")

if self.emergency\_detected and measurement\_data['rsrp'] < -100:

self.prioritize\_resources() # Re-prioritize if signal strength is low

def generate\_e2\_output(self, message\_type, data):

return json.dumps({

'type': message\_type,

'data': data,

'timestamp': datetime.now().isoformat()

})

# Example usage

xapp = WomanSafetyXapp()

# Simulate E2 input (voice data)

e2\_input\_voice = json.dumps({

'type': 'voice\_data',

'data': 'Help! I need emergency assistance!',

'timestamp': datetime.now().isoformat()

})

xapp.process\_e2\_input(e2\_input\_voice)

# Simulate E2 input (UE measurement)

e2\_input\_measurement = json.dumps({

'type': 'ue\_measurement',

'data': {

'ue\_id': 'emergency\_ue',

'rsrp': -95,

'rsrq': -12

},

'timestamp': datetime.now().isoformat()

})

xapp.process\_e2\_input(e2\_input\_measurement)

## 

## 

## 

## **Relation to Standards**

1. ITU-T Y.3173: Integrating real-time machine learning into the network for tasks like emergency detection, resource allocation, and response optimization

Example:

1. Automation Level: Assessing if the network can autonomously allocate resources in response to emergencies.
2. Learning Efficiency: How quickly an AI system learns from emergency patterns and improves its response.

2.ITU-T Y.2091,refers to a structured set of capabilities designed to offer value-added functionality, particularly in this use case emergency response and women's safety. Let's break down how this

Application in use case:

* Structured Capabilities:
  + The safety system integrates multiple components like voice command recognition, real-time location tracking, and emergency alert dispatch. Each of these components contributes to the structured functionality of the application.
* Value-Added Functionality:
  + The core value lies in the system’s ability to provide timely and effective help during an emergency. By triggering alerts through voice commands or other data sources and seamlessly communicating with authorities and volunteers, the system offers crucial functionality that enhances personal safety.
* Supported by Services:
  + The application would leverage several network services, such as:
    - Location-based services (LBS) for tracking the user.
    - AI-driven voice recognition for detecting distress.
    - Emergency communication protocols for prioritizing critical data transmission over the network (via 5G/6G).

## **Code submission details**

1. I created the GitHub account
2. I created the repository, made it public
3. I commit my file, added the document
4. The repository link is here: Aura: GitHub Submission Link

## 