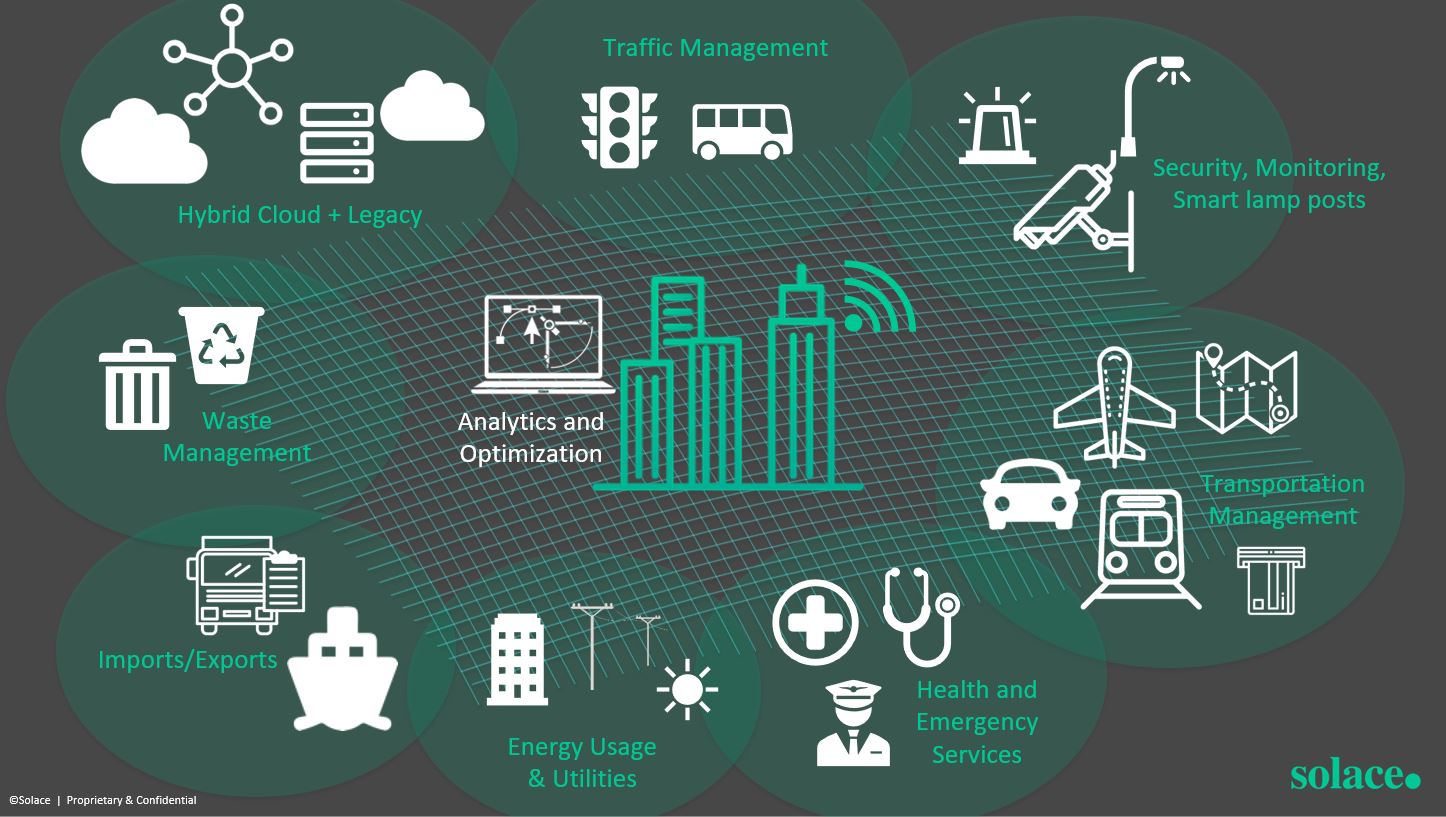
**IOT PHASE 3**

**PUBLIC TRANSPORT OPTIMIZATION**

****

Public transport optimization refers to the process of improving the efficiency, effectiveness, and overall quality of public transportation systems. This involves using various strategies and technologies to make public transportation more convenient, reliable, and environmentally sustainable.

**IOT Projects in Public Transport**

Explore a range of fascinating IOT projects that have been implemented in public transport systems, including smart ticketing, real-time passenger information systems, and automated fleet management.

**The Importance of Predicting Arrival Time**

Learn why accurately predicting arrival times is crucial for passengers, operators, and city planners. Discover how IOT-powered algorithms and data analysis techniques can help optimize scheduling and reduce delays.

**Data Collection and Analysis Methods**

* **Sensor Integration**

See how IOT sensors collect real-time data on vehicle locations, passenger counts, and traffic conditions to feed into predictive models.

* **Big Data Analytics**

Explore the use of advanced analytics techniques to process and analyze massive datasets, unlocking valuable insights for optimizing public transport operations.

* **Machine Learning**

Discover how machine learning algorithms can be trained on historical data to recognize patterns and predict arrival times with greater accuracy.

**Development of Prediction Models**

* **Supervised Learning**

Understand the process of training prediction models using labeled data, enabling them to make accurate forecasts based on past patterns.

* **Neural Networks**

Explore how artificial neural networks can leverage the power of parallel processing and hidden layers to create more sophisticated prediction models.

* **Time Series Analysis**

Learn how statistical techniques can analyze historical data to identify temporal patterns and seasonality, facilitating precise arrival time predictions.

**Integration of Prediction Models**

* **Real-Time Data Sync**

Discover how prediction models integrate with real-time data feeds from sensors and other sources, enabling dynamic adjustments to arrival time estimates.

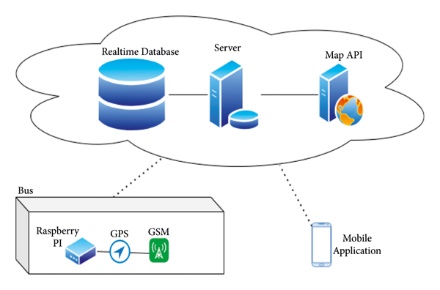
* **Passenger Information Systems**

See how arrival time predictions are seamlessly communicated to passengers through digital displays, mobile apps, and public announcements.

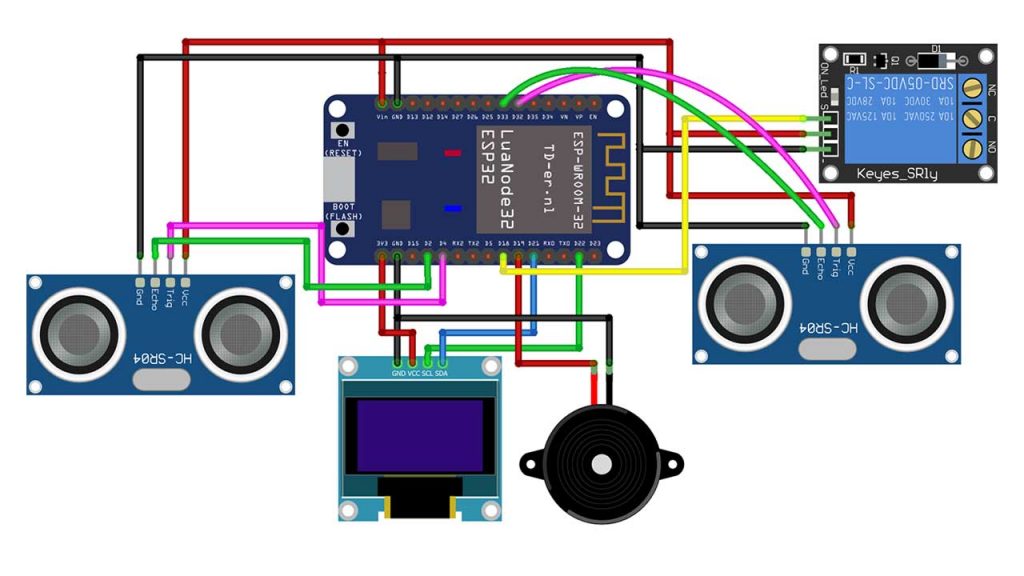
* **Operational Decision Support**

Explore how transport operators use prediction models to optimize driver schedules, minimize waiting times, and improve the overall efficiency of their services.

**Location Provider Module**

****

**Passenger Count**



Program for Passenger Count

import time

import random

class Bus:

def \_\_init\_\_(self, bus\_id):

self.bus\_id = bus\_id

self.location = 0

def move(self):

self.location += random.randint(1, 5)

def get\_location(self):

return self.location

class Passenger:

def \_\_init\_\_(self, passenger\_id, destination):

self.passenger\_id = passenger\_id

self.destination = destination

def request\_bus\_info(self):

# Simulate requesting bus information

print(f"Passenger {self.passenger\_id}

is looking for a bus to {self.destination}.")

class BusStop:

def \_\_init\_\_(self, name, location):

self.name = name

self.location = location

def is\_near(self, bus):

return abs(bus.get\_location() - self.location) <= 2

def main():

# Create buses and bus stops

buses = [Bus(1), Bus(2)]

bus\_stops = [BusStop("Stop A", 10), BusStop("Stop

B", 20)]

passengers = [Passenger(1, "Stop B"), Passenger(2,

"Stop A")]

while True:

for bus in buses:

bus.move() for passenger in passengers:

if passenger.request\_bus\_info():

for stop in bus\_stops:

if stop.name == passenger.destination:

if stop.is\_near(bus):

print(f"Bus {bus.bus\_id} is approaching

{stop.name} for Passenger

{passenger.passenger\_id}.")

time.sleep(1)

# Simulate time passing

if \_\_name\_\_ == "\_\_main":

main()

