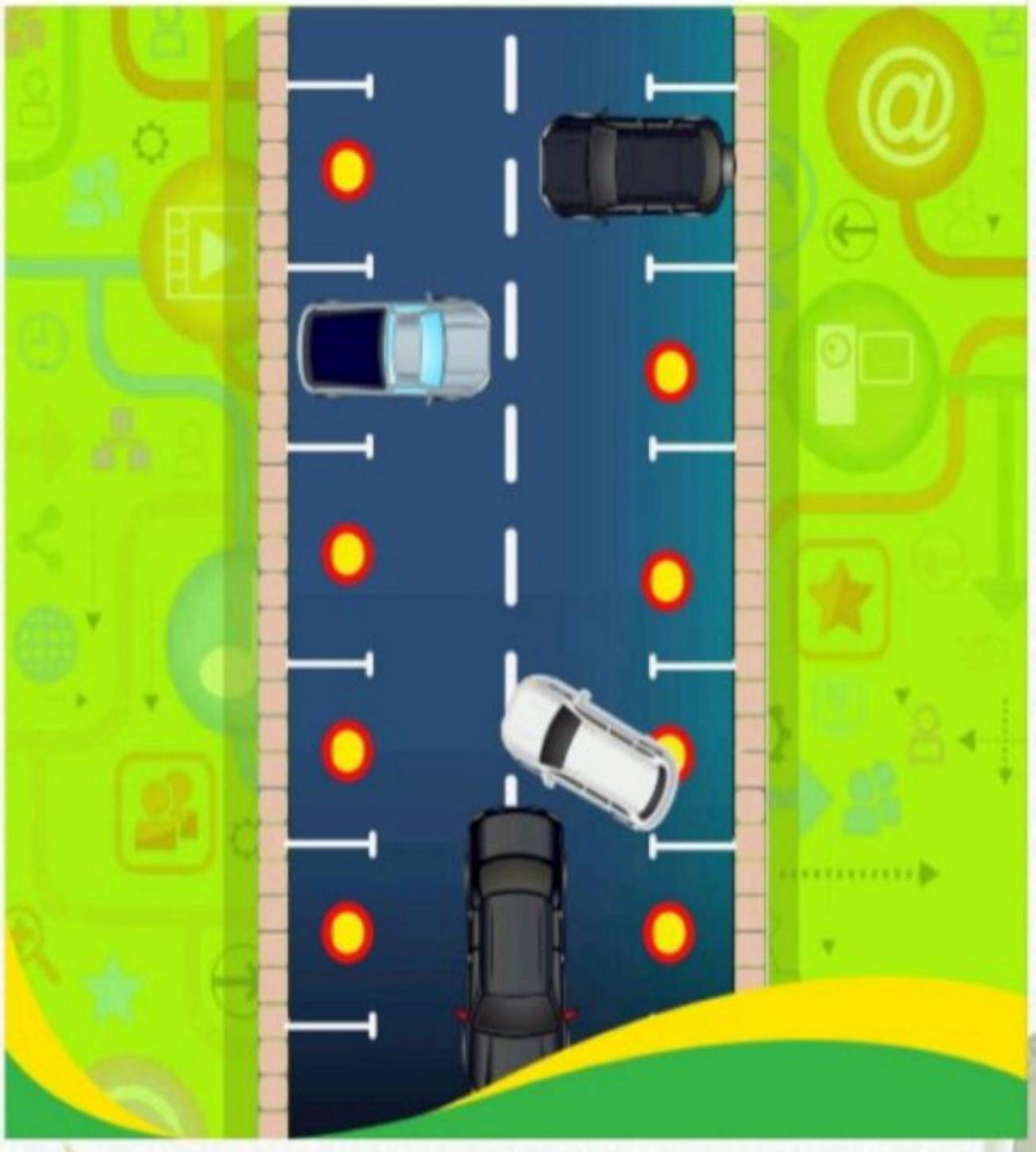
A PRESENTED BY SMART PARKING

PRESENTED BY:

DEVAKI k
ADHI LAKSHMI S
DIVYA G
SARANYA R
SURYA PRAKSH S

SMART PARKING



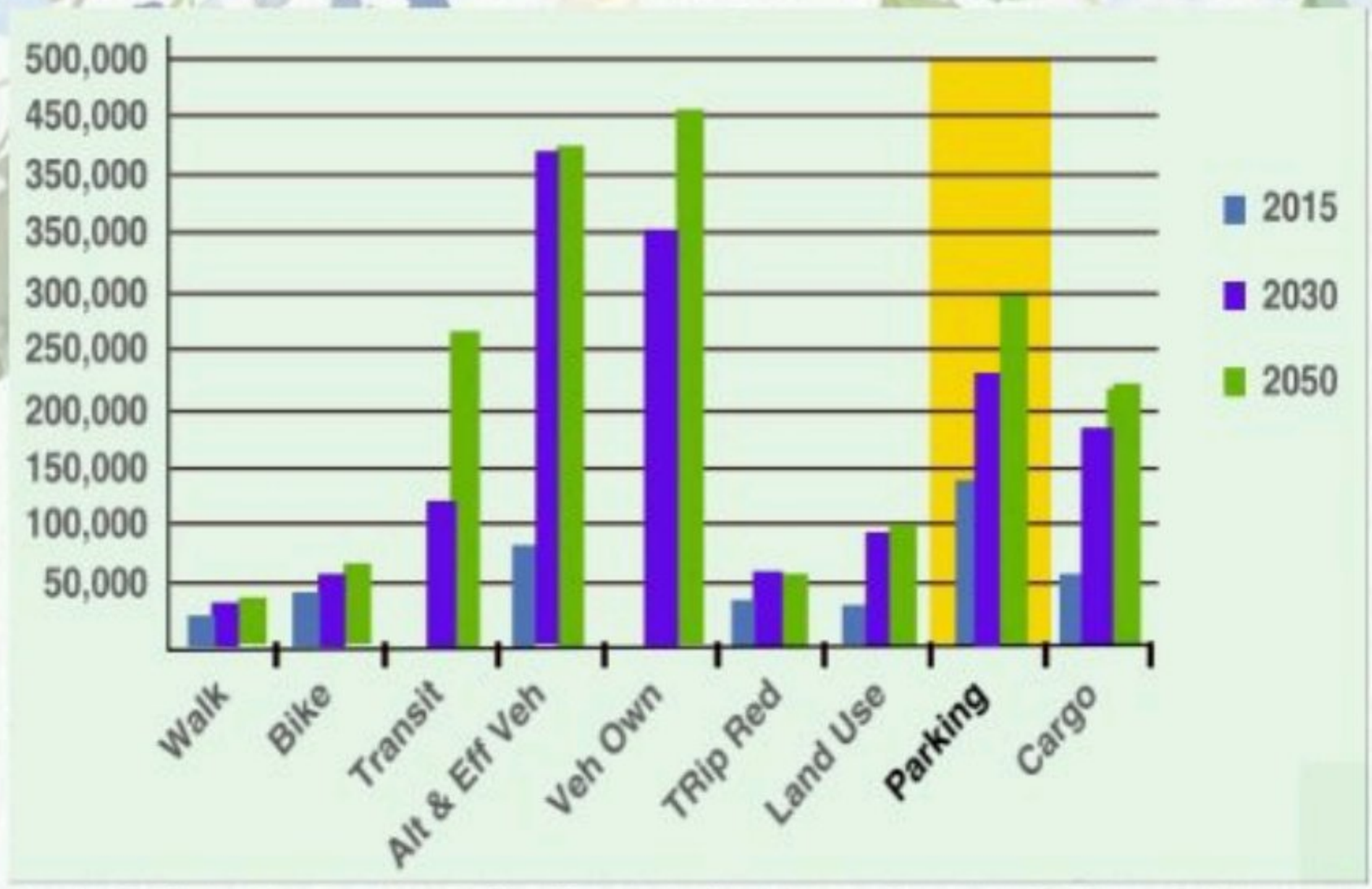
Potential market Landscape

The rapid growth in the number of vehicles worldwide is intensifying the problem of the scarcity of parking space. Again according to industry data, 30% of traffic congestion occurs due to vehicle drivers struggling to find parking space. These in turn are magnifying the necessity of smart and efficient parking systems. Today's intelligent parking management systems are capable of providing extreme level of convenience to the drivers, as well as simplifying and automating the business operation and administrative functions of the parking site owners.

Emerging Trends in Parking

Following are the trends having the greatest effect on Parking Industry

The high growth rate in the registration of new cars worldwide, with major boom from regional economies such as Asia Pacific (APAC), will open the window of opportunities for parking management business. The ongoing and upcoming smart city projects worldwide will



Smart Parking systems typically obtains information about available parking spaces in a particular geographic area and process is

real-time to place vehicles at available positions .It involves using low-cost sensors, real-time data collection, and mobile-phone-enabled

automated payment systems that allow people to reserve parking in advance or very accurately predict where they will likely find a spot.

When deployed as a system, smart parking thus reduces car emissions in urban centers by reducing the need for people to needlessly

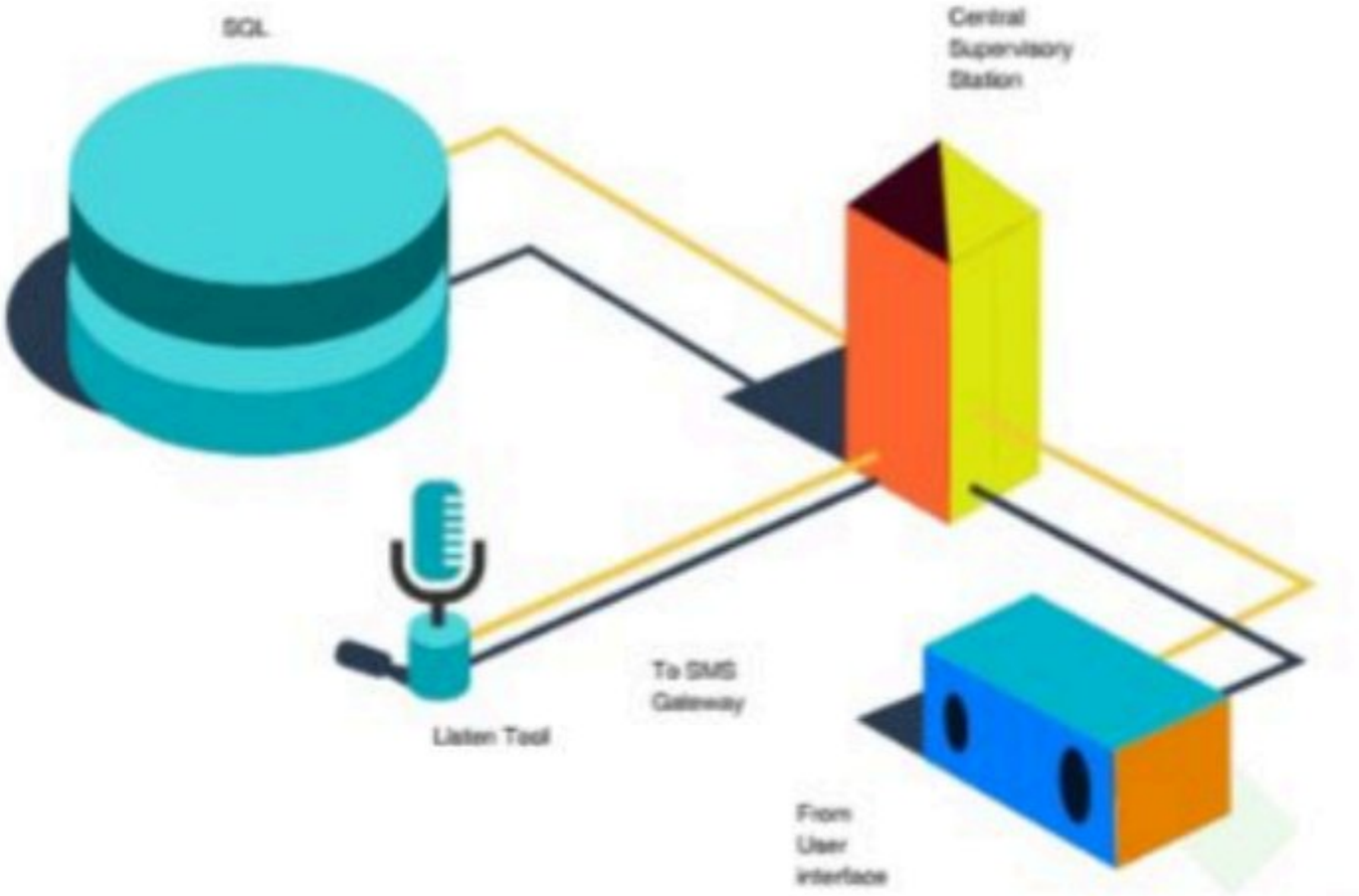
circle city blocks searching for parking. It also permits cities to carefully manage their parking supply

Smart parking helps one of the biggest problems on driving in urban areas; finding empty parking spaces and controlling illegal parking.

This implies M2M technologies aims rightness/safety as well as conveni

challenge in reducing the cost and complexity of smart parking. The variety of infrastructure hardware and software systems that need to be integrated is enormous and add to it the conventional older hardware making investment in Smart Parking solution highly risky and fragmented.

Another maior pain point comes from the electronic payment vendors.



create room for the intelligent parking management systems. The global parking management industry is expected to grow at a Compound Annual Growth Rate (CAGR) of 11.4% from 2014 to 2019.

The parking management market is estimated to be at $5,025.9 million in 2014. The market is expected to grow in tandem with the growth in vehicle ownerships and parking facilities development. Need for smooth traffic flow, business benefits to the parking site operators, and decreasing hardware and connectivity costs are the key drivers for the parking management industry.

Traffic congestion and gasoline prices leads the list for the major societal changes having significant influence on parking.

25• The Parking Assistance System include three modules-Monitoring module, Control module and a displaying unit. Along with above

three module it will also have centralized supervisory system to maintain a data base of parking space and will have a SMS gateway.

• The monitoring module includes ultrasonic sensors/ ambient light sensor which identifies the free parking spaces and transmits the

Information to control unit through Zigbee.

• Apart from detecting the car the sensor also provides additional information like the stretch of time the car has been parked and also

its health status.

• The control units processes the information and sends the information to Centralized supervisory system.

• Centralized supervisory system receives information of parking space from the controller through UDP. It then sends the information

possess a great threat or concern to the system scalability.

The technology platform supporting P&E , PARC and PUCRS systems comprises of a myriad of hardware sensors, dynamic messaging systems and traffic control devices, wireless and wireline telecommunications systems, computer clients and servers and hardware drivers and application interfaces.

Enabling all these devices from thousands of different vendors to communicate and tying them together into one platform is the greatest

• The Parking Assistance System include three modules-Monitoring module, Control module and a displaying unit. Along with above

three module it will also have centralized supervisory system to maintain a data base of parking space and will have a SMS gateway.

• The monitoring module includes ultrasonic sensors/ ambient light sensor which identifies the free parking spaces and transmits the

Information to control unit through Zigbee.

• Apart from detecting the car the sensor also provides additional information like the stretch of time the car has been parked and also

its health status.

• The control units processes the information and sends the information to Centralized supervisory system.

• Centralized supervisory system receives information of parking space from the controller through UDP. It then sends the information

such as slot allotted, time parked, billing information

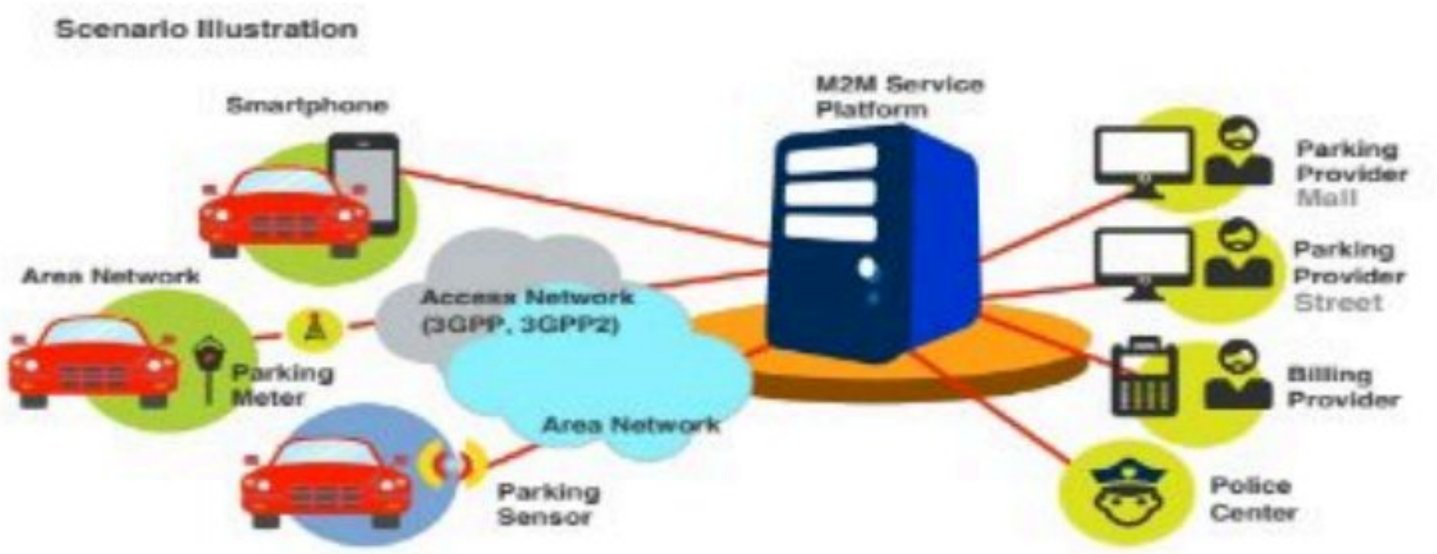
than cars.

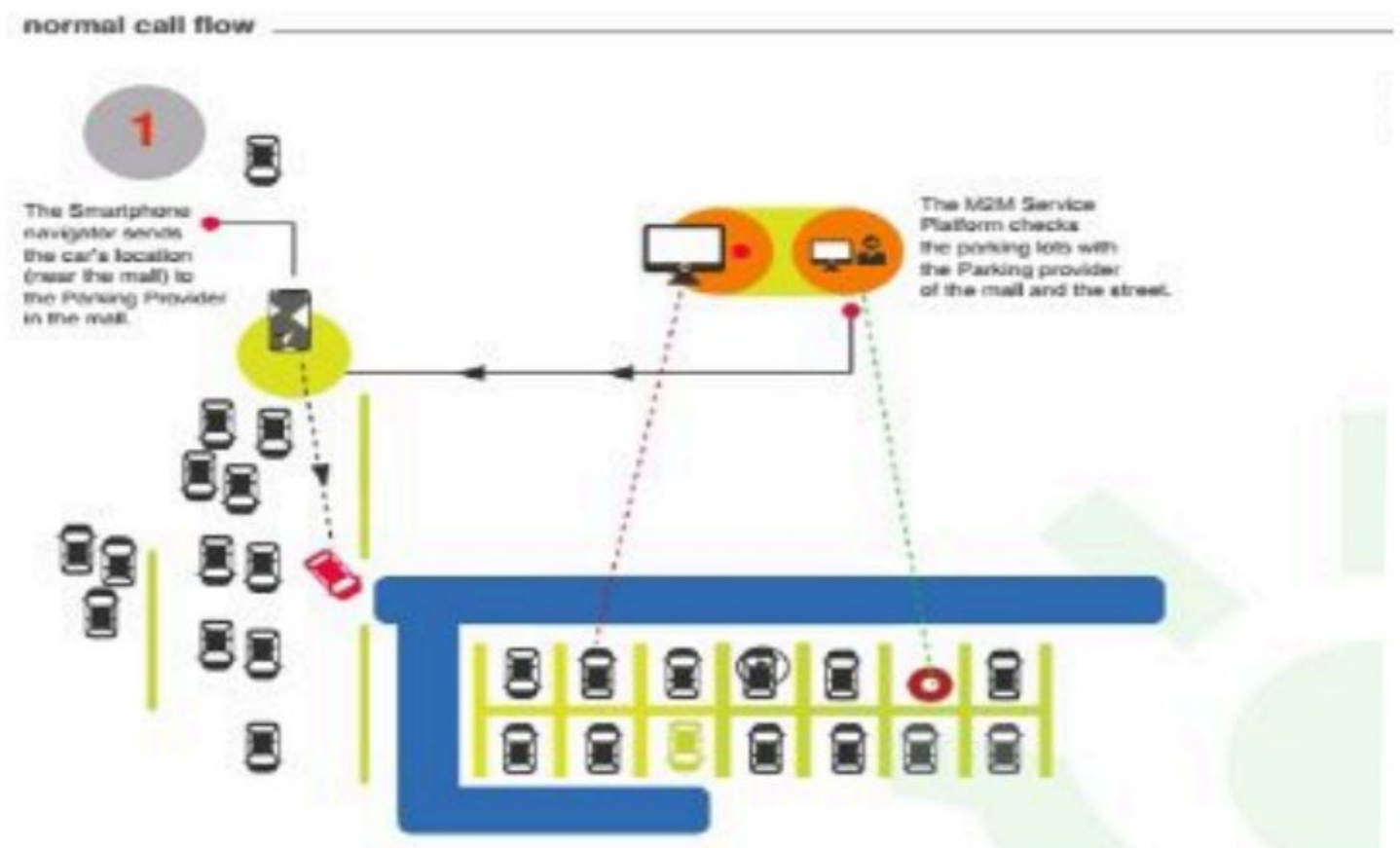
Street to Vehicle communication would be pivotal and crucial along with the Vehicle to

Vehicle communication as the success

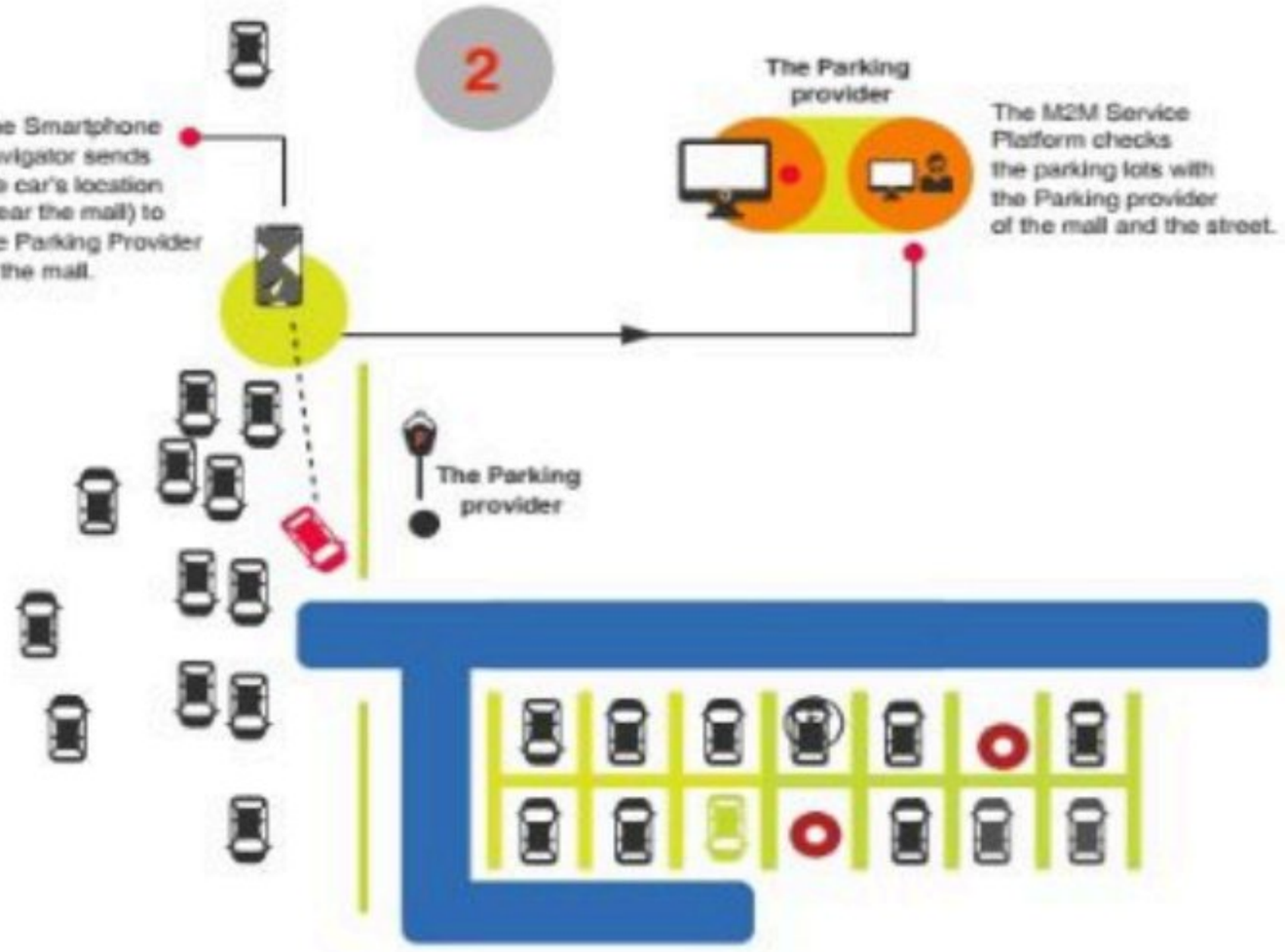
and market readiness of Autonomous vehicle ecosystem lies in collecting and

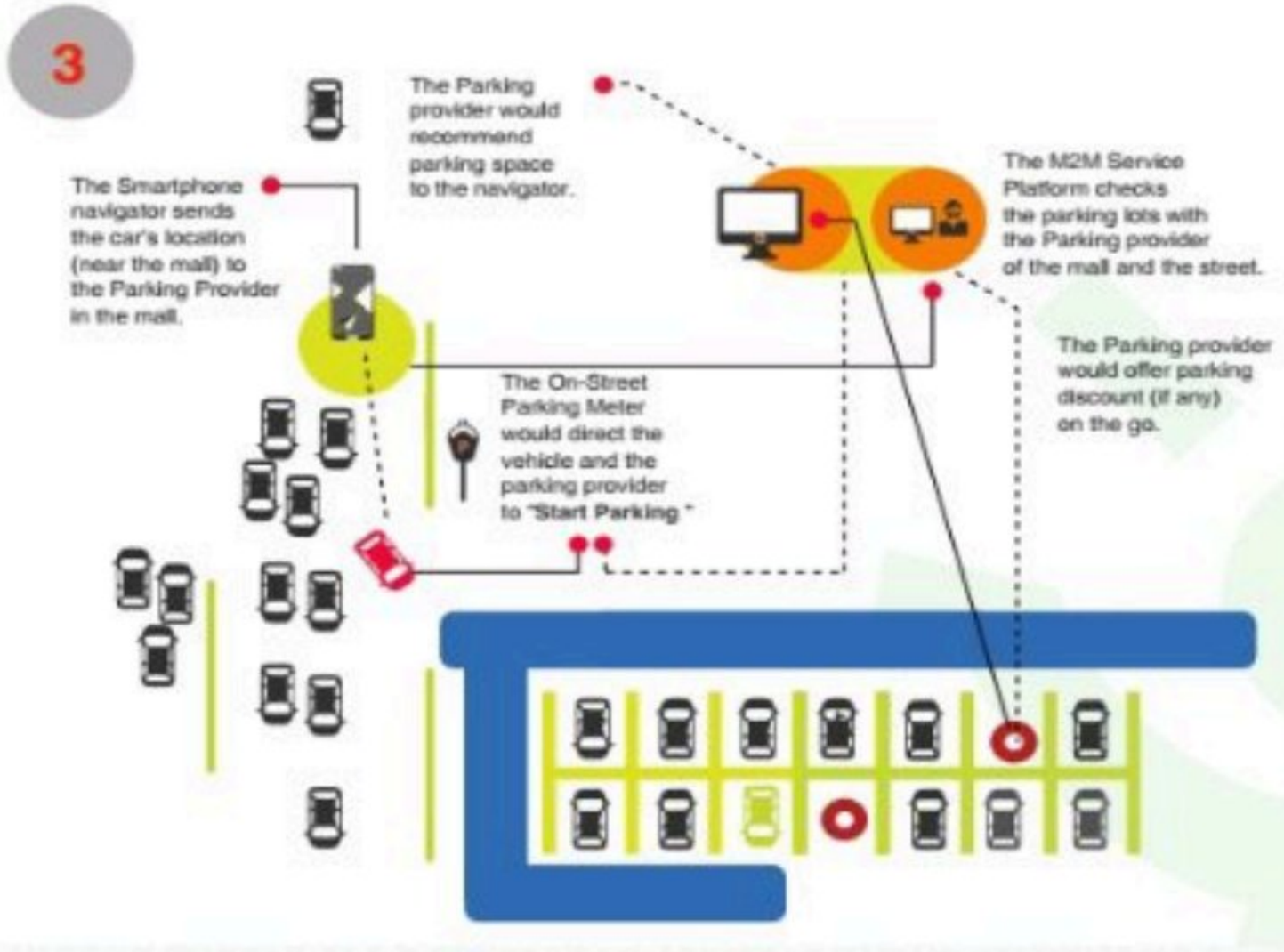
interpreting the data at the Street Level.





such as slot allotted, time parked, billing information



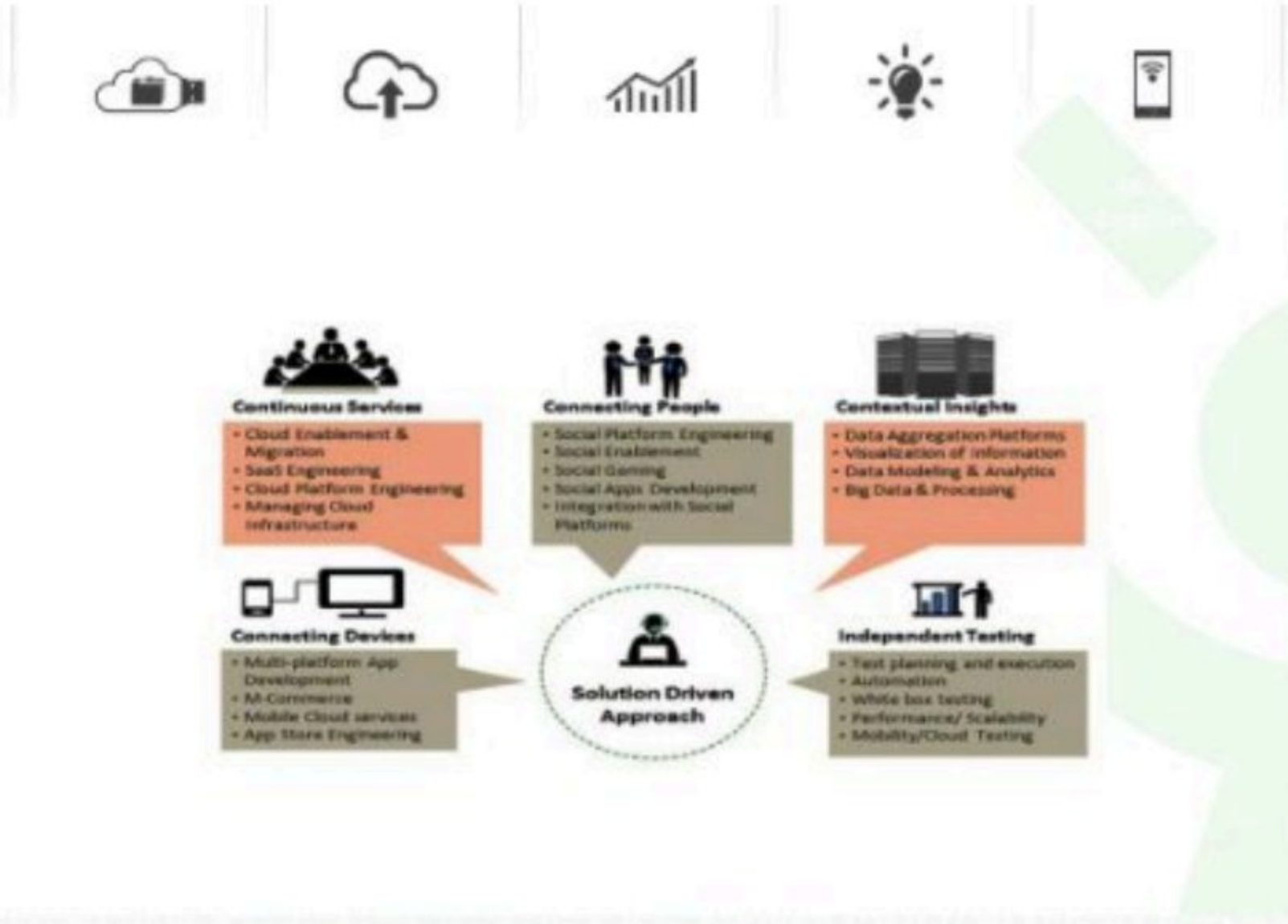


fragmented.

Another major pain point comes from the electronic payment vendors. These payment processors provide permit based electronic

payment, typically for a convenience fee. The key to many of these hosted solutions is scalability, the ability of the transaction processor

to support over wide geographical, market and service areas, with minim



I

The major challenge in Parking Systems is of system integration due to wide variety of hardware and software platforms involved and hence possess a great threat or concern to the system scalability.

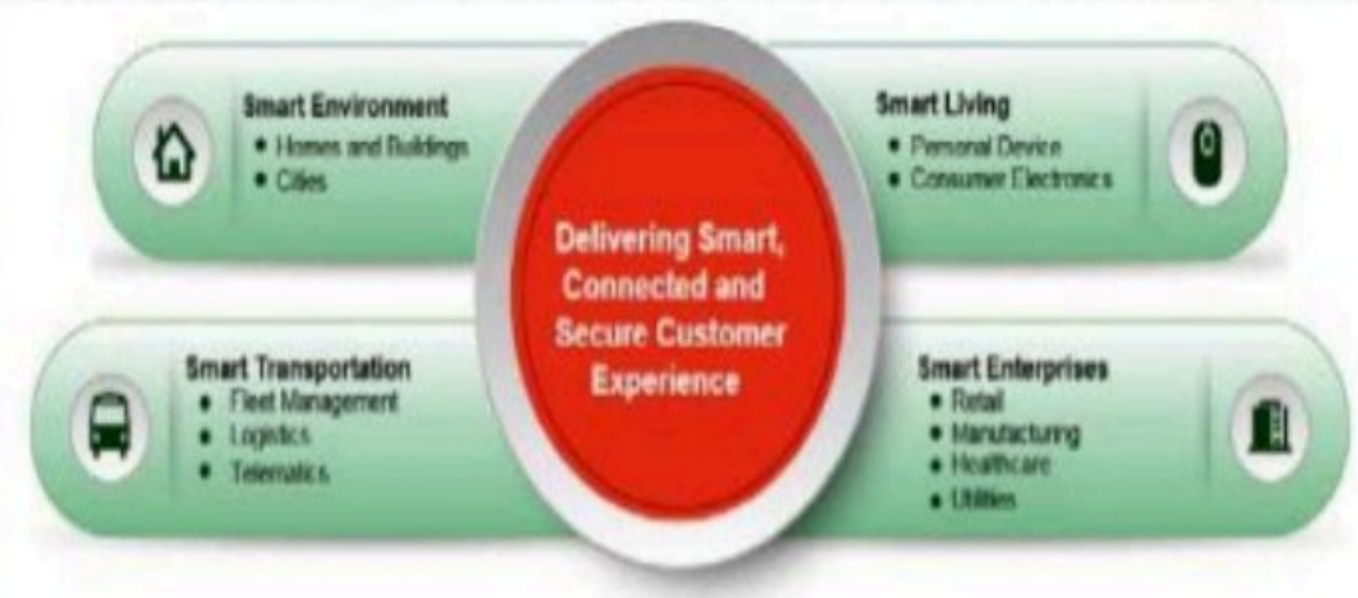
The technology platform supporting P&E , PARC and PUCRS systems comprises of a myriad of hardware sensors, dynamic messaging systems and traffic control devices, wireless and wireline telecommunications systems, computer clients and servers and hardware drivers and application interfaces.

Enabling all these devices from thousands of different vendors to communicate and tying them together into one platform is the greatest challenge in reducing the cost and complexity of smart parking. The variety of infrastructure hardware and software systems that need to be integrated is enormous and add to it the conventional older hardware making investment highly risky and

growth in vehicle ownerships and parking facilities development. Need for smooth traffic flow, business benefits to the parking

site operators, and decreasing hardware and connectivity costs are the key drivers for the parking management industry.

The major challenge in Parking Systems is of system integration due to wide variety of hardware and software platforms involved and hence



challenge in reducing the cost and complexity of smart parking. The variety of infrastructure hardware and software systems that need to be

integrated is enormous and add to it the conventional older hardware making investment

fragmented.

Another major pain point comes from the electronic payment vendors. These payment processors provide permit based electronic payment, typically for a convenience fee. The key to many of these hosted solutions is scalability, the ability of the transaction processor to support over wide geographical, market and service areas, with minim

A brief blog for smart parking using ESP32 with camera module and some recommended

a devices that makes easy to make the smart parking system

Components

Query

Process

Components

Connection

Drive API Setup

Code

Final Instruction

Search Quotes

Process

To set up a smart parking system using ESP32 with a camera module, proximity sensor,

servo sensor, and display, which takes a picture of the vehicle and sends it to Google

Drive and displays the available slots on the LCD, you will need to follow these steps:

Components

Gather the required components:

ESP32 board

Camera module (such as the OV7670 camera module)

Proximity sensor (such as the HC-SR04 ultrasonic sensor)

Servo motor

Breadboard

Jumper wires

Power supply (such as a USB cable and power bank)

LCD display module

Google Drive API credentials

Connection

Connect the components:

Connect the VCC and GND pins of the camera module to the 3.3V and GND pins on the ESP32 board, respectively.

Connect the SCL and SDA pins of the camera module to the corresponding pins on the ESP32 board (such as pins 22 and 21, respectively).

Connect the VCC and GND pins of the proximity sensor to the 5V and GND pins on the ESP32 board, respectively.

Connect the trig and echo pins of the proximity sensor to any two digital pins on the ESP32 board (such as pins 12 and 14).

Connect the signal pin of the servo motor to any digital pin on the ESP32 board (such as pin 27), and the VCC and GND pins to the 5V and GND pins on the ESP32 board, respectively.

Connect the SDA and SCL pins of the LCD display module to the corresponding pins on the ESP32 board (such as pins 4 and 5, respectively), and the VCC and GND pins to the 5V and GND pins on the ESP32 board, respectively.

Drive API Setup

Set up the Google Drive API:

Go to the Google API Console and create a new project.

Enable the Google Drive API and create a new OAuth 2.0 client ID.

Download the JSON file with the API credentials.

Write the code: Here's some sample code that you can use as a starting point for your

Code

Copy

COPIED

```
#include <Servo.h>

#include <Wire.h>

#include "esp_camera.h"

#include <WiFi.h>

#include <HTTPClient.h>

#include <ArduinoJson.h>

#include <WiFiClientSecure.h>

#include <LiquidCrystal_I2C.h>

#define TRIG_PIN 12

#define ECHO_PIN 14

#define SERVO_PIN 27

#define LCD_ADDR 0x27

#define LCD_COLS 16
```

```
#define LCD_ROWS 2

Servo myservo;

int angle = 0;

const char* ssid = "your_SSID";

const char* password = "your_PASSWORD";

const char* gdrive_token = "your_GDRIVE_TOKEN";

WiFiClientSecure client;

HTTPClient http;

LiquidCrystal_I2C lcd(LCD_ADDR, LCD_COLS, LCD_ROWS);

String get_gdrive_folder_id() {

  client.setAuthorization("Bearer " + String(gdrive_token));

  http.begin(client,

"https://www.googleapis.com/drive/v3/files?q=name%3D%27SmartParking%27+and+mi

meType%3D%27application%2Fvnd.google-apps.folder%27&fields=files(id)&key=YOUR_A

PI_KEY");

  http.addHeader("Content-Type", "application/json");

  int httpCode = http.GET();

  String folder_id = "";

if (httpCode == HTTP_CODE_OK) {

  String payload = http.getString();

  StaticJsonDocument<512> doc;

  DeserializationError error = deserializeJson(doc, payload);
```

```
if (!error) {

    http.beginRequest();

    http.write(request_body.c_str(), request_body.length());

    int bytes_sent = 0;

  while (bytes_sent < file_size) {

    int bytes_to_read = file_size - bytes_sent;

    if (bytes_to_read > 1024) { bytes_to_read = 1024; }

    uint8_t buffer[bytes_to_read];

    int bytes_read = file.read(buffer, bytes_to_read);

    http.write(buffer, bytes_read);

      bytes_sent += bytes_read;

    }

    http.endRequest();

    String response = http.getString();

    http.end();

  }

}

void setup() {

  Serial.begin(115200);
```

```
  pinMode(TRIG_PIN, OUTPUT);

  pinMode(ECHO_PIN, INPUT);

  myservo.attach(SERVO_PIN);

  lcd.init();

  lcd.backlight();

  lcd.clear();

  lcd.setCursor(0, 0);

  lcd.print("Smart Parking");

  lcd.setCursor(0, 1);

  lcd.print("Initializing...");

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {

    delay(1000);

    Serial.println("Connecting to WiFi...");

  }

  Serial.println("Connected to WiFi");

  camera_config_t config;

  config.ledc_channel = LEDC_CHANNEL_0;

  config.ledc_timer = LEDC_TIMER_0;

  config.pin_d0 = 5;

  config.pin_d1 = 18;

  config.pin_d2 = 19;
```

```
config.pin_d3 = 21;
config.pin_d4 = 36;
config.pin_d5 = 39;
config.pin_d6 = 34;
config.pin_d7 = 35;
config.pin_xclk = 0;
config.pin_pclk = 22;
config.pin_vsync = 25;
config.pin_href = 23;
config.pin_sscb_sda = 26;
config.pin_sscb_scl = 27;
config.pin_pwdn = -1;
config.pin_reset = -1;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;
if (psramFound()) {
  config.frame_size = FRAMESIZE_UXGA;
  config.jpeg_quality = 10;
  config.fb_count = 2;
} else {
  config.frame_size = FRAMESIZE_SVGA;
  config.jpeg_quality = 12;
```

```
    config.fb_count = 1;
  }

  esp_err_t err = esp_camera_init(&config);

  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);

    return;

  }

  lcd.clear();

  lcd.setCursor(0, 0);

  lcd.print("Smart Parking");

  lcd.setCursor(0, 1);

    lcd.print("Ready");

    delay(2000);

  }

  void loop() {

    float duration, distance;

    digitalWrite(TRIG_PIN, LOW);

    delayMicroseconds(2);

    digitalWrite(TRIG_PIN, HIGH);

    delayMicroseconds(10);

    digitalWrite(TRIG_PIN, LOW);

    duration = pulseIn(ECHO_PIN, HIGH);
```

```
distance = duration * 0.034 / 2;

int available_slots = distance / 50;

if (available_slots < 0) { available_slots = 0; }

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Available slots:");

lcd.setCursor(0, 1);

lcd.print(String(available_slots));

if (available_slots > 0) {

  float angle = map(available_slots, 0, MAX_SLOTS, 0, 180);

  myservo.write(angle);

delay(500);

camera_fb_t* fb = NULL;

fb = esp_camera_fb_get();

if (!fb) {

  Serial.println("Failed to capture image");

  return;

}

Serial.printf("Captured image with size %u\n", fb->len);

uploadToGoogleDrive(fb);

esp_camera_fb_return(fb);

delay(5000);
```

}
}

Final Instruction

This code calculates the available parking slots based on the distance measured by the proximity sensor, and then maps this value to an angle to control the servo motor. It then captures an image using the camera module and uploads it to Google Drive using the uploadToGoogleDrive() function. Finally, it waits for 5 seconds before repeating the process.

Note that you will need to replace YOUR_ACCESS_TOKEN and YOUR_FOLDER_ID with your own values in the uploadToGoogleDrive() function, as well as ssid and password with your WiFi credentials in the setup() function. You will also need to adjust the pin numbers and camera configuration to match your own hardware setup.

Search Quotes

"Discover the Future of Parking: How Smart Parking Systems Are Revolutionizing Urban Mobility. Learn about the benefits of IoT-enabled solutions for drivers, cities, and businesses."

"Say Goodbye to Parking Hassles: How Smart Parking Systems Are Streamlining Parking. Explore the latest innovations in parking technology, including sensors, data analytics, and mobile apps."

"Maximize Your Parking Efficiency: How Smart Parking Systems Can Optimize Your Parking Operations. Find out how automated parking systems can reduce congestion, increase revenue, and improve customer satisfaction."

"The Ultimate Guide to Smart Parking Systems: How to Choose the Right Solution for Your Needs. Get expert advice on selecting the best parking technology for your business or municipality."

"From Parking Lots to Smart Cities: How Smart Parking Systems are Driving Sustainable Urban Development. Discover the role of smart parking in creating more livable, eco-friendly cities for the future."

THANK YOU