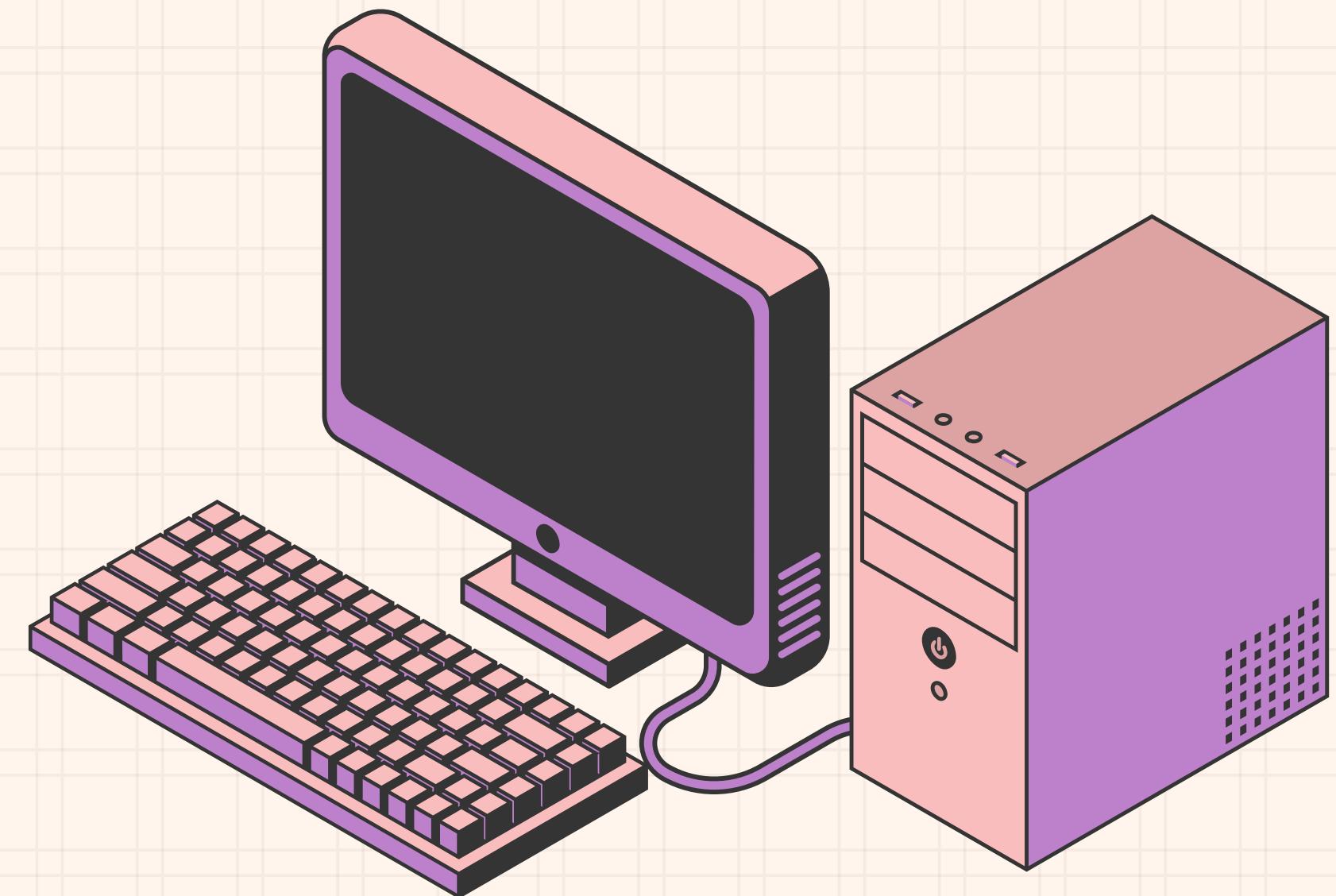


BÃI ĐÔ XE THÔNG MINH



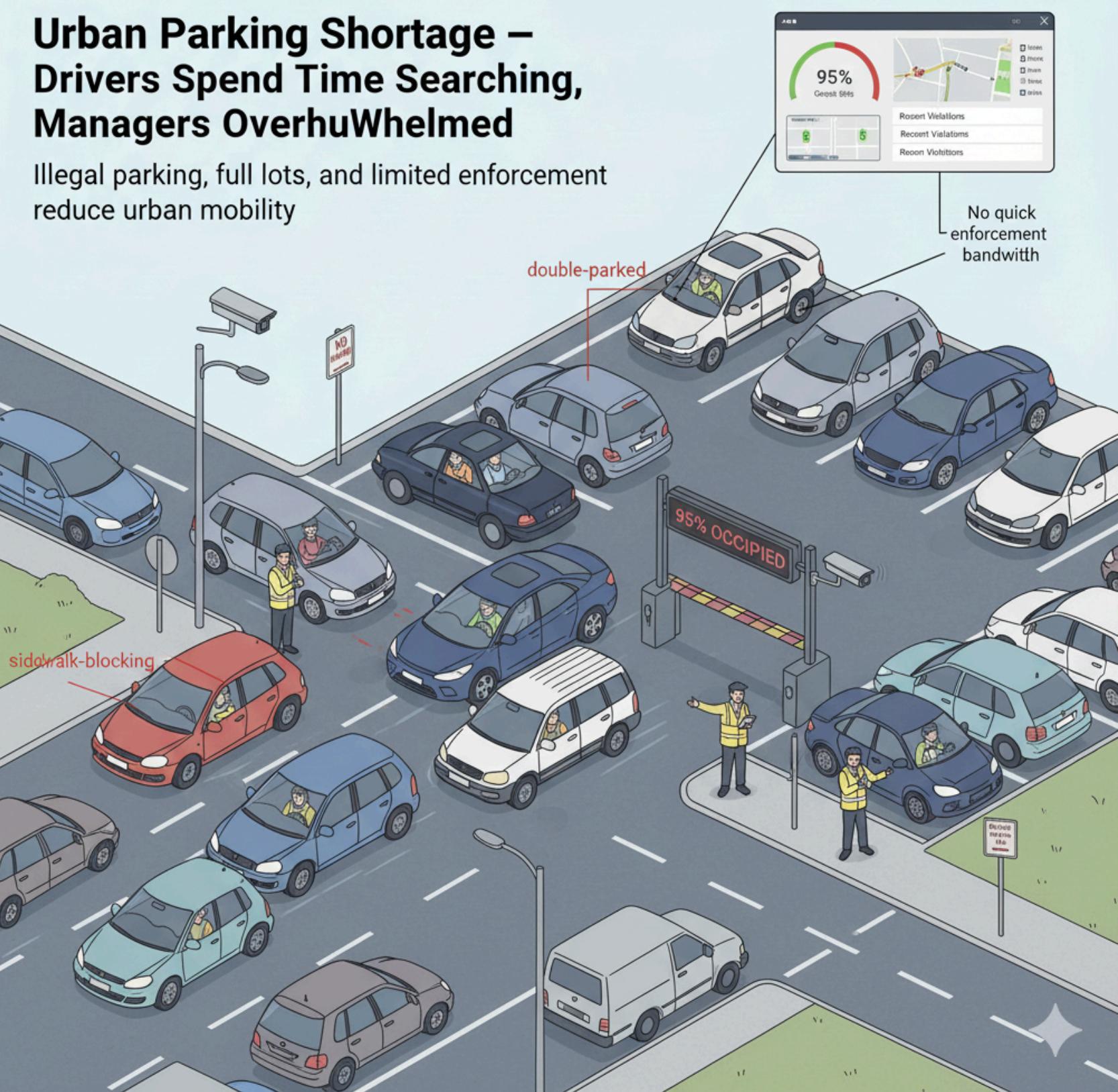
1. GIỚI THIỆU ĐỀ TÀI

Thực trạng

Ở các đô thị lớn, nhu cầu đỗ xe, đặc biệt là cho ô tô đang còn thiếu sót rất nhiều về mặt quản lý, người lái xe phải mất nhiều thời gian để đi tìm chỗ trống và việc phát hiện đỗ xe sai cách gây tốn diện tích cùng với số lượng xe nhiều khiến người quản lí gặp nhiều trở ngại trong việc kiểm soát bãi đỗ xe

Urban Parking Shortage – Drivers Spend Time Searching, Managers Overwhelmed

Illegal parking, full lots, and limited enforcement reduce urban mobility



2. LÝ THUYẾT VÀ CÔNG NGHỆ LIÊN QUAN

Tổng quan về IoT

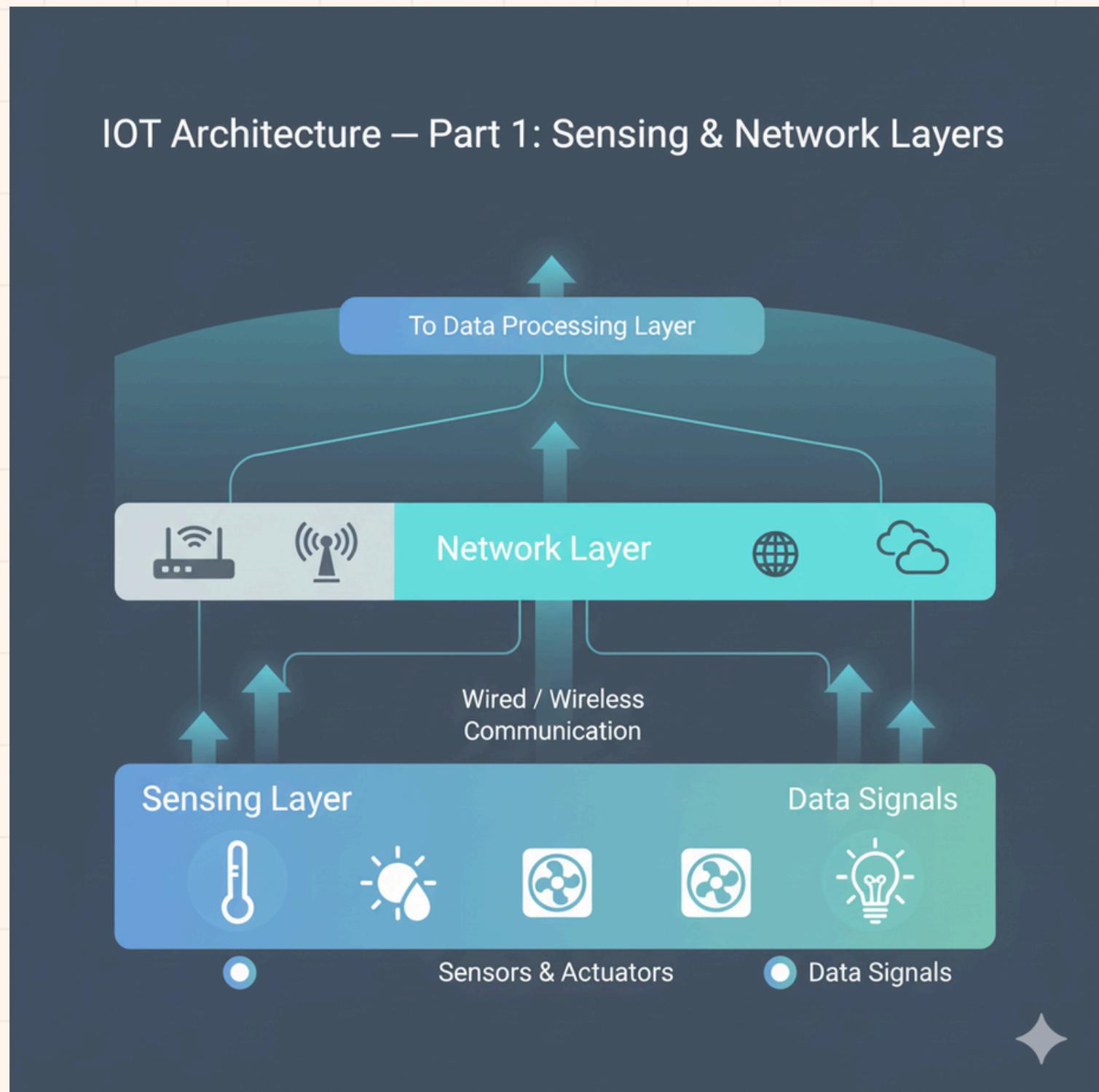
- IoT (Internet of Things) là mạng lưới các thiết bị thông minh có khả năng giao tiếp với đám mây và với nhau thông qua Internet.
- Nguồn gốc:
 - Bắt đầu từ thập niên 1990 khi cảm biến và bộ xử lý được tích hợp vào vật dụng hàng ngày.
 - Ban đầu phát triển chậm do chip còn lớn và đắt.
 - Sự xuất hiện của chip RFID và vi xử lý nhỏ, tiết kiệm năng lượng giúp IoT bùng nổ.
- Ý nghĩa:
 - IoT biến mọi vật dụng thành thiết bị thông minh.
 - Hình thành hệ sinh thái “thiết bị điện toán vô hình” trong nhà, doanh nghiệp, văn phòng.



2. LÝ THUYẾT VÀ CÔNG NGHỆ LIÊN QUAN

Mô hình 4 lớp

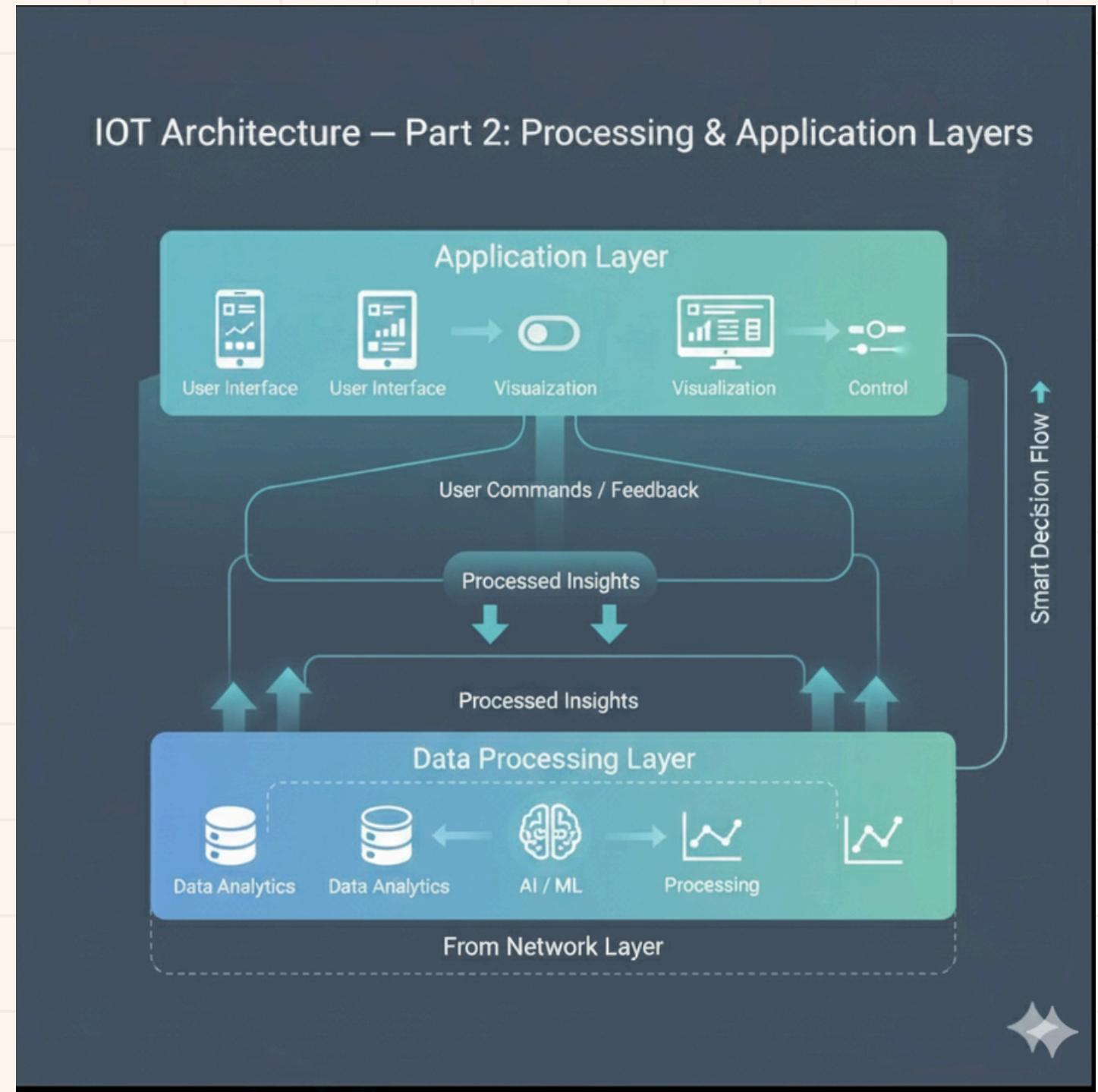
- Lớp Cảm biến (Sensing Layer):
 - Chức năng: Thu thập dữ liệu từ môi trường (nhiệt độ, độ ẩm, ánh sáng, âm thanh...).
 - Thành phần: Cảm biến và cơ cấu chấp hành (actuators).
 - Kết nối: Sử dụng các giao thức truyền thông có dây hoặc không dây để gửi dữ liệu lên lớp mạng.
- Lớp Truyền thông (Network Layer):
 - Đảm nhiệm truyền dữ liệu giữa các thiết bị IoT và Internet.
 - Giúp thiết bị kết nối, giao tiếp và trao đổi dữ liệu trong toàn hệ thống IoT.



2. LÝ THUYẾT VÀ CÔNG NGHỆ LIÊN QUAN

Mô hình 4 lớp

- Lớp Xử lý dữ liệu (Data Processing Layer):
 - Thu thập, phân tích và diễn giải dữ liệu từ lớp mạng.
 - Biến dữ liệu thô thành thông tin có ý nghĩa.
 - Nền tảng phân tích, thuật toán học máy (AI/ML)
 - Hỗ trợ ra quyết định thông minh dựa trên dữ liệu.
- Lớp Ứng dụng (Application Layer):
 - Tương tác trực tiếp với người dùng cuối.
 - Cho phép điều khiển và giám sát thiết bị IoT.
 - Tích hợp middleware và phân tích nâng cao (trực quan hóa dữ liệu, học máy...).



2. LÝ THUYẾT VÀ CÔNG NGHỆ LIÊN QUAN

Tổng quan về phần cứng

ESP32(CH340): Dòng vi điều khiển (microcontroller) System-on-Chip (SoC) chi phí thấp và tiết kiệm năng lượng được phát triển bởi Espressif Systems. Điểm nổi bật chính của nó là việc tích hợp sẵn các module Wi-Fi và Bluetooth trên cùng một con chip, giúp nó trở thành một lựa chọn lý tưởng cho các dự án Internet of Things (IoT)



2. LÝ THUYẾT VÀ CÔNG NGHỆ LIÊN QUAN

Tổng quan về phần cứng

Cảm biến siêu âm (SRF05) được sử dụng để đo khoảng cách đến một vật thể mà không cần tiếp xúc. Nguyên lý hoạt động của nó dựa trên việc phát và thu sóng siêu âm (sóng âm có tần số cao hơn ngưỡng tai người nghe được).



Servo motor (SG90) động cơ điện cho phép điều khiển chính xác vị trí góc quay của trục động cơ. Servo SG90 là một loại servo mini rất phổ biến, giá rẻ và thường được sử dụng trong các dự án nhỏ.



3. PHÂN TÍCH YÊU CẦU

Mục tiêu hệ thống

- Giám sát tự động: nhận diện biển số, cấp phép xe ra vào
- Tự động hóa: barie đóng/mở bằng servo + cảm biến siêu âm
- Phát hiện vị trí đỗ trống/có xe và hiển thị cho tài xế
- Quản lý, thống kê số lượng xe, thời gian đỗ, tình trạng cảm biến



3. PHÂN TÍCH YÊU CẦU

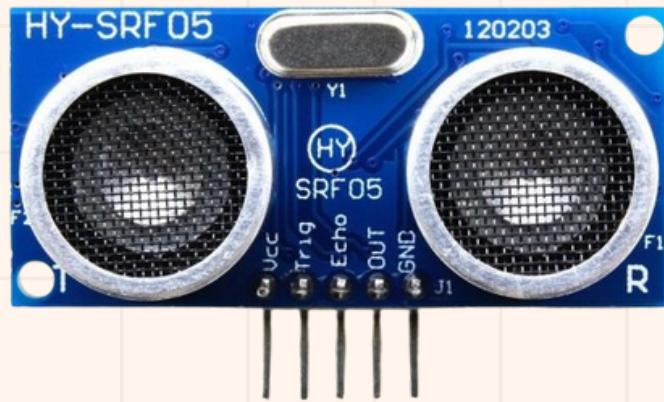
Phạm vi triển khai

Số lượng thiết bị:

- 5 cảm biến siêu âm SRF05
- 1 servo đóng mở
- Bộ kít arduino (board test, dây cắm)
- Vi mạch ESP32
- 1 Camera
- 1 Thiết bị hiển thị thông tin

Mô trường hoạt động:

- Ngoài trời: Chịu mưa, nắng điều kiện thiếu sáng
- Trong nhà: yêu cầu độ chính xác cao và chống nhiễu tốt



3. PHÂN TÍCH YÊU CẦU

Tiêu chí thành công(KPIs)

| Tiêu chí | Mục tiêu mong muốn |
|--------------------|----------------------------------------------------------------------------------------------|
| Độ chính xác | Tỷ lệ phát hiện trạng thái có xe/ trống $\geq 95\%$ |
| Độ trễ | $\leq 0,2s$ - Thời gian hệ thống hiển thị từ khi phát hiện $\leq 5s$ - Đóng mở thanh chắn |
| Độ tin cậy | Uptime $>98\%$ hệ thống hoạt động ổn định, không mất kết nối |
| Chi phí triển khai | Tổng chi phí dự kiến $\leq 500.000đ$ cho toàn bộ hệ thống |
| Khả năng mở rộng | Dễ dàng thêm cảm biến mới, không ảnh hưởng hệ thống hiện tại |

3. PHÂN TÍCH YÊU CẦU

Kết quả mong đợi

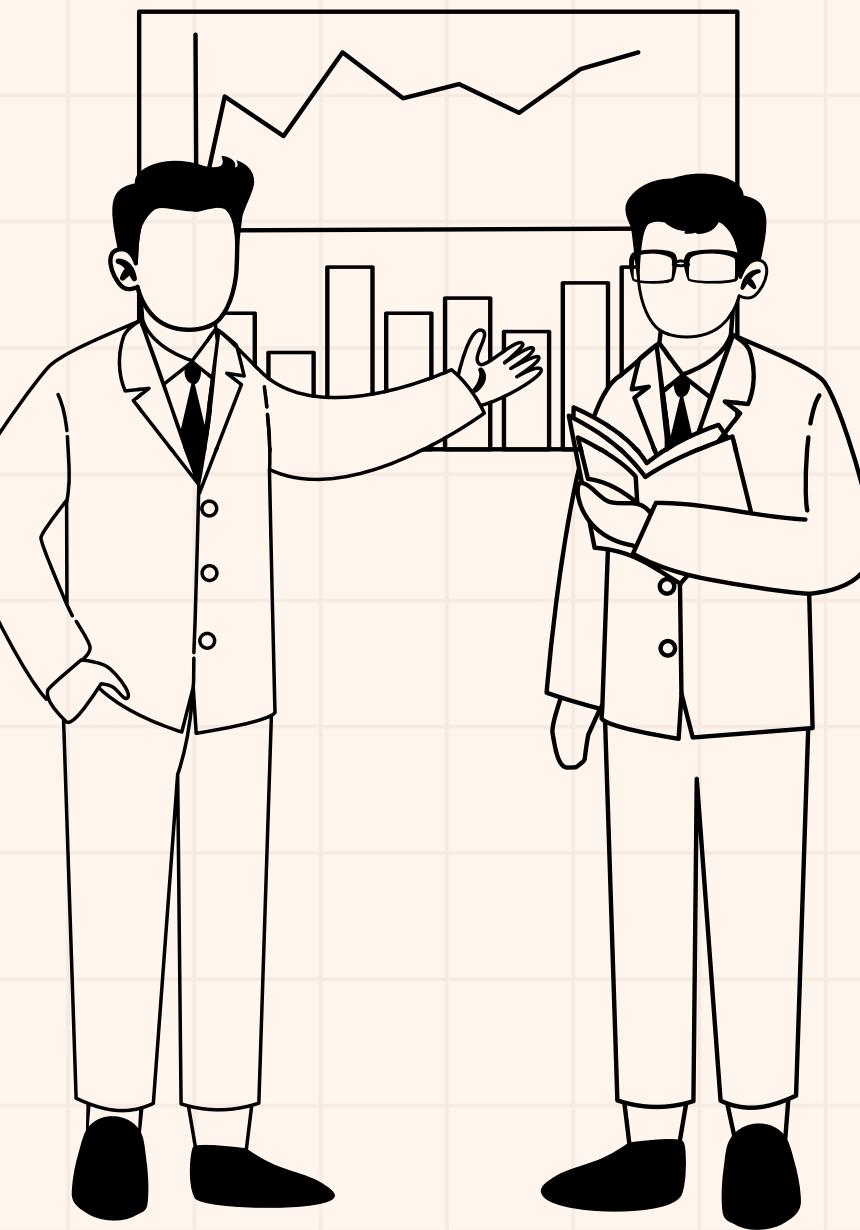
- Tự động hóa quy trình quản lý bãi đỗ xe → tiết kiệm thời gian, tối ưu hiệu suất
- Người quản lý giám sát qua màn hình
- Phát hiện vị trí đỗ trống/có xe và hiển thị cho tài xế
- Dữ liệu thu thập từ biển số phục vụ cho việc theo dõi quản lý xe



3. PHÂN TÍCH YÊU CẦU

Thu thập yêu cầu từ các bên liên quan

- Người dùng cuối(tài xế):
 - Muốn hiển thị rõ ràng vị trí trống
 - Hoạt động ổn định
 - Dễ thao tác khi ra vào bãi
- Doanh nghiệp/ban quản lý bãi:
 - Yêu cầu hệ thống giúp giảm nhân sự trông xe
 - Tự động hóa đóng/mở barie
 - Ghi nhận biển số
 - Thống kê số lượng xe và thời gian đỗ
- Bộ phận kỹ thuật IT:
 - Cần hệ thống tích hợp được với hạ tầng hiện có
 - Truyền dữ liệu ổn định
 - Bảo mật truy cập, dễ bảo trì- mở rộng

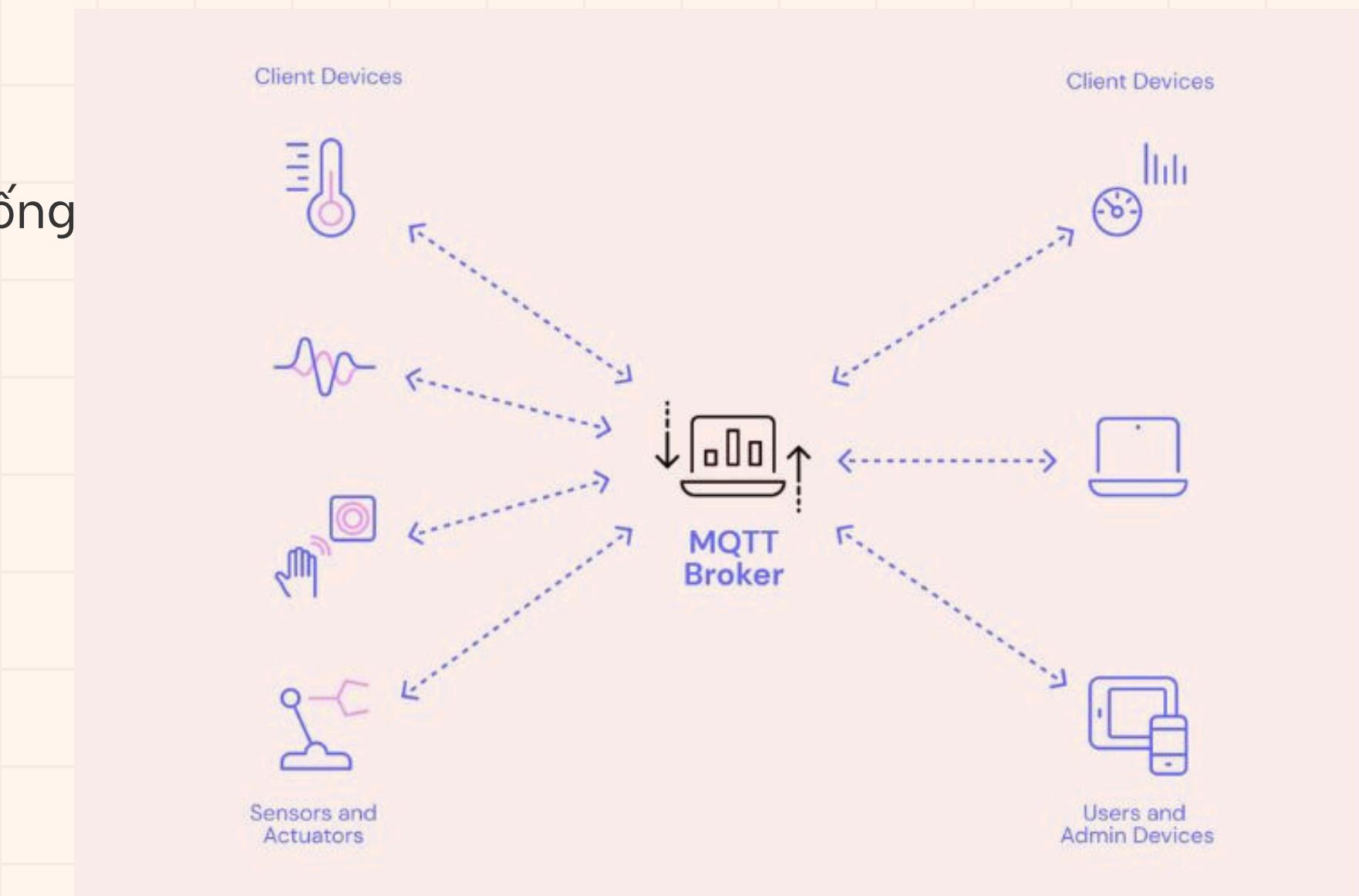


3. PHÂN TÍCH YÊU CẦU

Yêu cầu chức năng

Thu thập và xử lý dữ liệu cảm biến:

- Mỗi ô đồ gắn cảm biến siêu âm xác định có xe hay trống
- Camera LPR tại cổng ghi nhận biển số xe ra/vào
- Dữ liệu gửi về ESP32 → MQTT → Blynk Cloud
- Cơ chế lưu đệm (buffer) và retry khi mất mạng



3. PHÂN TÍCH YÊU CẦU

Yêu cầu chức năng

Hiển thị, phân tích và điều khiển:

- Website hiển thị bản đồ trạng thái
- Bảng thống kê số chỗ trống và lịch sử xe ra/vào
- Gửi lệnh điều khiển ngược(mở barie, cập nhật trạng thái)
- Phân quyền điều khiển: chỉ nhân viên được mở barie thủ công

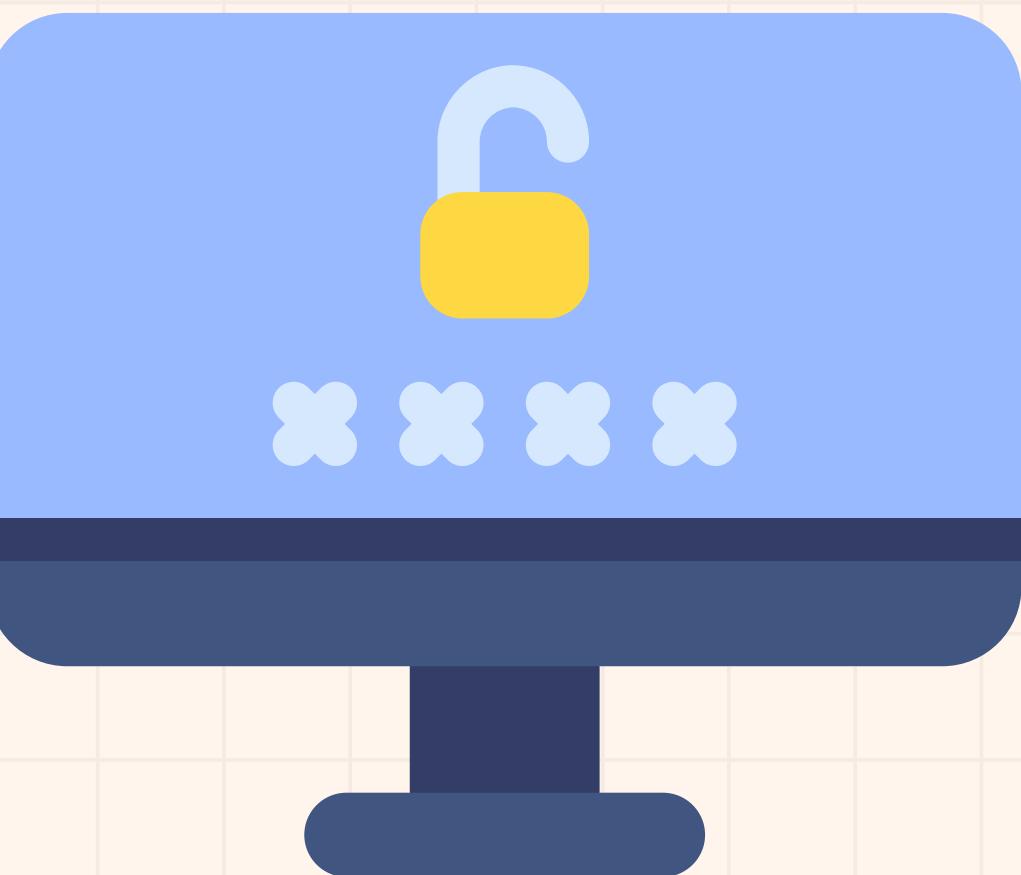


3. PHÂN TÍCH YÊU CẦU

Yêu cầu chức năng

Chức năng mở rộng và bảo mật hệ thống:

- Quản lý người dùng: tạo tài khoản, phân quyền(khách, nhân viên, quản trị)
- Bảo mật dữ liệu: Mã hóa đường truyền(HTTPS/TLS), xác thực API(JWT)
- Cập nhật OTA: Cho phép cập nhật phần mềm từ xa để bảo trì hệ thống

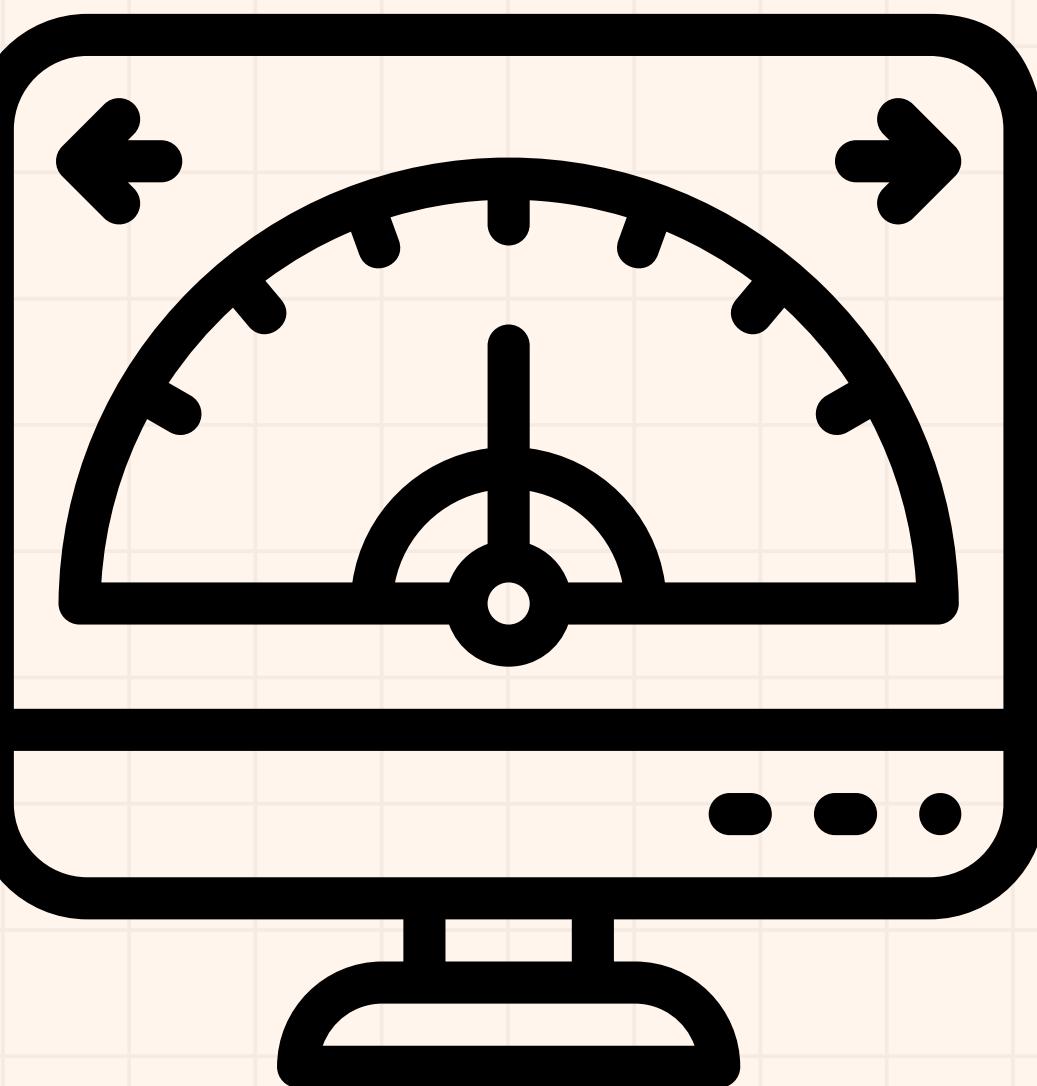


3. PHÂN TÍCH YÊU CẦU

Yêu cầu phi chức năng

Hiệu năng - Bảo mật - Độ tin cậy:

- Độ trễ cập nhật: <3s, nhận diện biến số <5s
- Hệ thống hoạt động ổn định > 99% uptime/tháng
- Dữ liệu mà hóa HTTPS/TLS, xác thực JWT
- ESP32 lưu đệm 1h khi mất kết nối

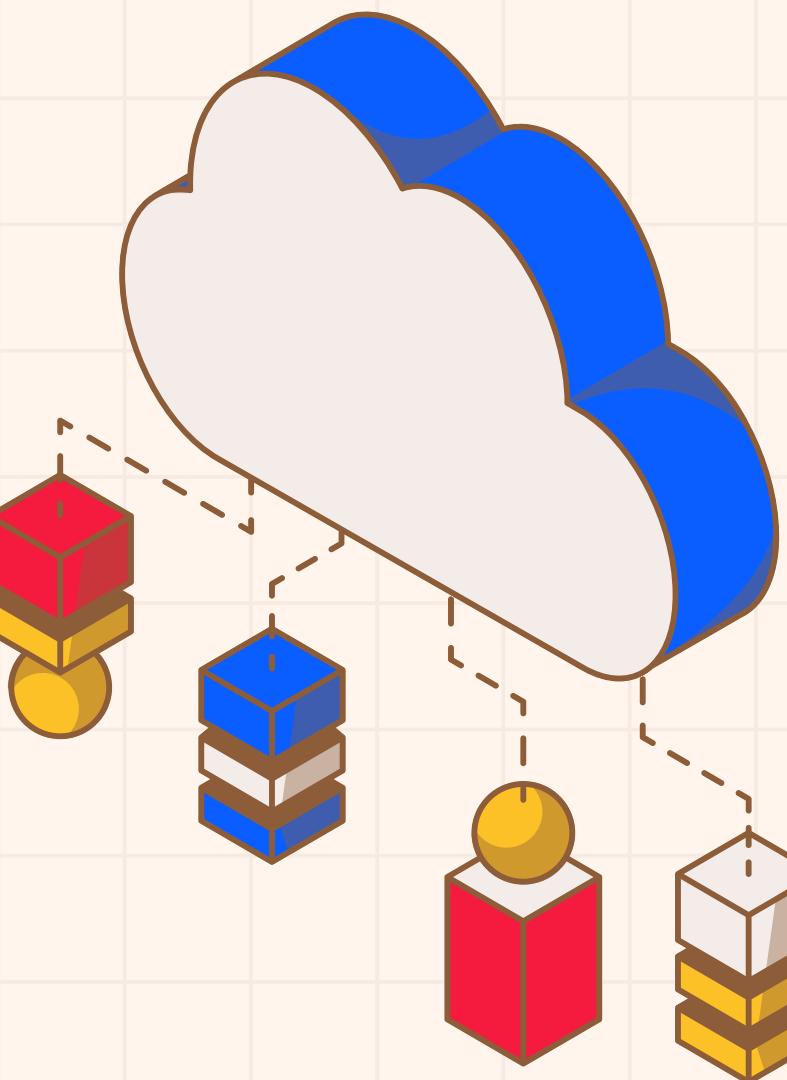


3. PHÂN TÍCH YÊU CẦU

Yêu cầu phi chức năng

Khả năng mở rộng - Chi phí - Năng lượng

- Mở rộng thêm 50% số cảm biến mà không đổi kiến trúc
- Kiến trúc microservice, nâng cấp module độc lập
- Dữ liệu cảm biến nhỏ, camera xử lý cục bộ, chỉ gửi văn bản
- Chi phí cloud \leq 1USD/chỗ/tháng



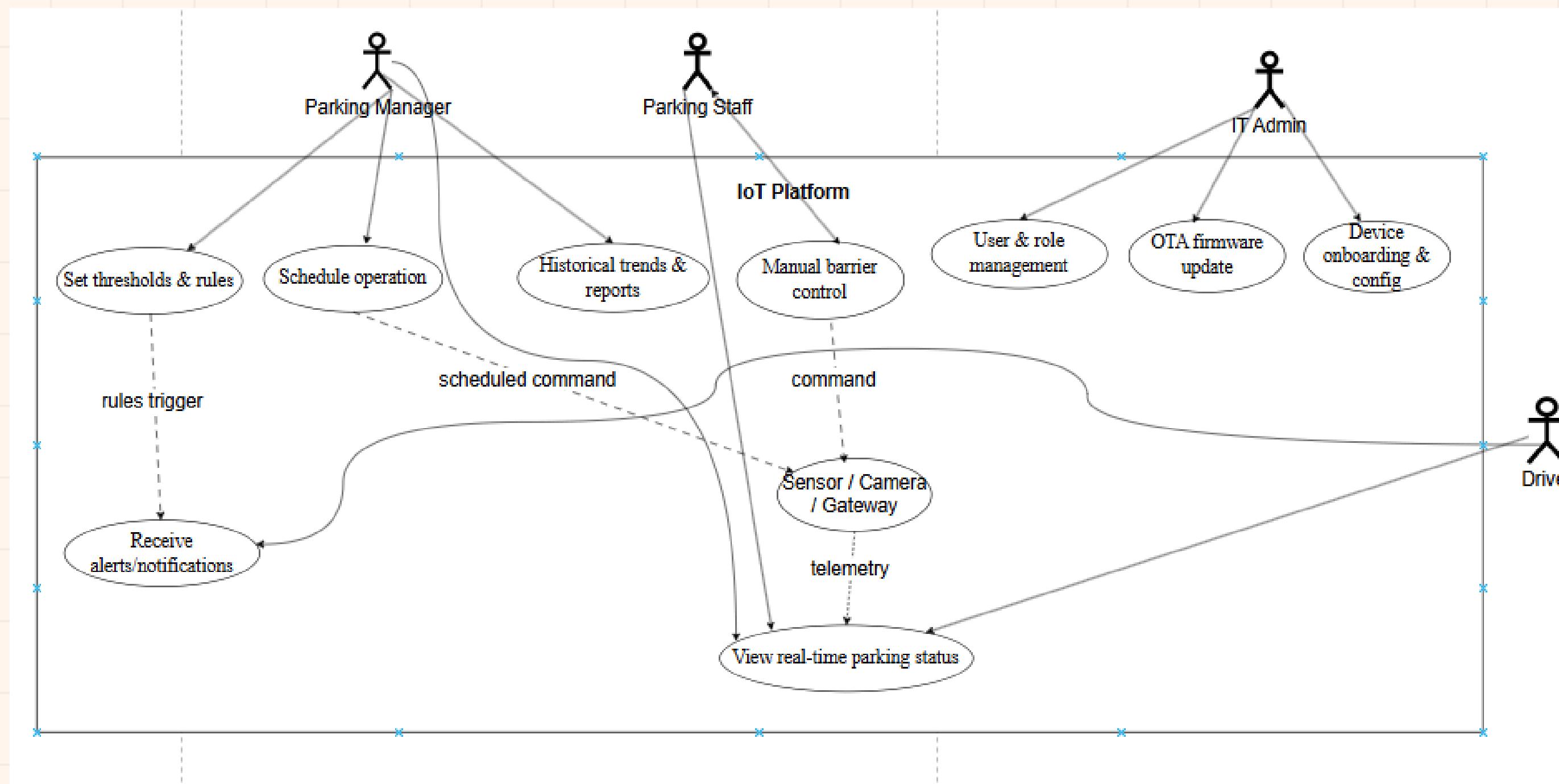
3. PHÂN TÍCH YÊU CẦU

Ràng buộc kỹ thuật và môi trường

- **Môi trường:** Ngoài trời(nắng,bụi, nhiễu) → dùng cảm biến IP65+, che chắn, có thể chuyển sang LoRa/NB-IoT nếu nhiễu cao.
- **Nguồn điện:** Một số khu xa nguồn → dùng pin/ năng lượng mặt trời, gateway có UPS mini
- **Pháp lý:** Tuân thủ băng tần hợp pháp(920MHz), mã hóa dữ liệu, hiển thị cảnh báo camera
- **Xử lý:** ESP32 chỉ làm nhiệm vụ đo và gửi, tính toán nặng đưa lên Server/Cloud
- **Pin & tuổi thọ:** Hạn chế tần suất gửi(5-10s), deep-sleep, theo dõi pin qua dashboard.
- **An toàn:** Có cảm biến chống kẹt, nút dừng khẩn cấp, cách ly mạch điều khiển
- **Bảo trì & mở rộng:** Thiết kế mô-đun, thêm cảm biến dễ bảo trì 1-2 tháng/lần

3. PHÂN TÍCH YÊU CẦU

Mô hình hóa yêu cầu



4. THIẾT KẾ LOGIC

Mục tiêu logic của hệ thống

- Giám sát (Monitoring): Theo dõi trạng thái các chỗ đỗ xe (có xe / trống / lỗi cảm biến).
- Điều khiển (Control): Tự động mở/đóng barie khi xe vào – ra.
- Thu thập dữ liệu (Data acquisition): Lưu dữ liệu xe vào/ra, thời gian đỗ, biển số, doanh thu, mức độ sử dụng bãi.
- Phân tích (Analytics): Thống kê và tối ưu vận hành – ví dụ dự đoán khung giờ cao điểm, gợi ý khu vực trống.

4. THIẾT KẾ LOGIC

Xác định các thực thể cần quản lý

| Nhóm | Thực thể | Miêu tả logic |
|------------|--------------------------------|------------------------------------------------------------|
| Thiết bị | Cảm biến chỗ đỗ | Gửi trạng thái (trống / có xe) định kỳ hoặc theo sự kiện. |
| Thiết bị | Barie / Cửa vào-ra | Nhận lệnh đóng/mở từ hệ thống điều khiển. |
| Hệ thống | Camera nhận diện biển số | Gửi thông tin biển số khi xe vào hoặc ra. |
| Dữ liệu | Phiên gửi xe (Parking session) | Tập hợp dữ liệu: biển số, thời gian vào - ra, vị trí, phí. |
| Người dùng | Người lái xe | Tương tác qua ứng dụng. |
| Nhân viên | Quản trị viên / bảo vệ | Theo dõi, xử lý ngoại lệ, xem báo cáo. |

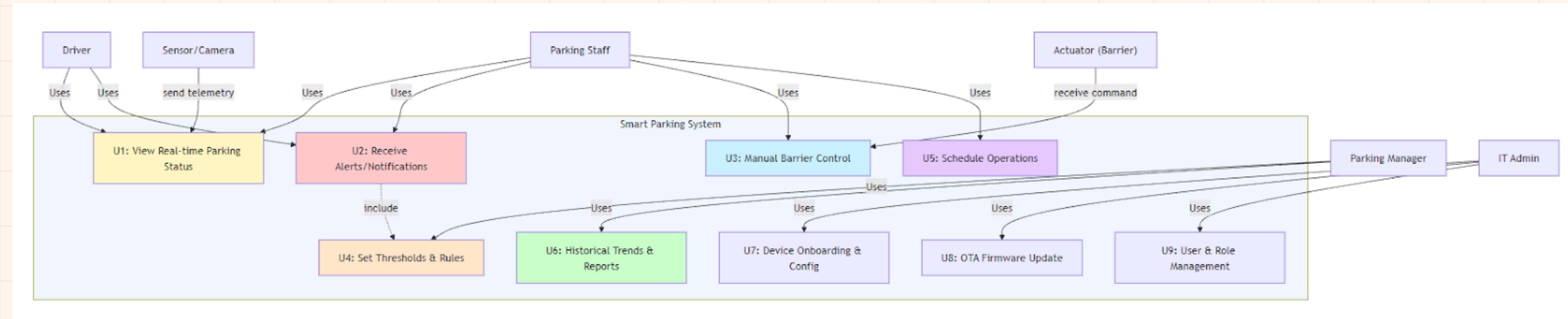
4. THIẾT KẾ LOGIC

Xác định ràng buộc logic

| Loại ràng buộc | Ví dụ |
|-----------------------|-----------------------------------------------------------------------------------|
| Tần suất dữ liệu | Cảm biến chờ đỗ gửi dữ liệu mỗi 5 giây hoặc khi trạng thái thay đổi. |
| Độ trễ (Latency) | Khi xe vào bãi, barie phải mở trong < 1 giây sau khi xác thực biển số. |
| Tính toàn vẹn dữ liệu | Mỗi biển số chỉ có tối đa 1 phiên gửi xe đang hoạt động. |
| An ninh - xác thực | Giao tiếp giữa thiết bị và server qua MQTT + TLS; xác thực người dùng bằng token. |
| Khả năng mở rộng | Hệ thống có thể mở rộng lên hàng trăm chỗ đỗ, nhiều tầng hoặc nhiều bãi. |
| Độ tin cậy | Nếu cảm biến lỗi, hệ thống vẫn cho phép thao tác thủ công. |

4. THIẾT KẾ LOGIC

Usecase chức năng mô tả tác vụ chính



4. THIẾT KẾ LOGIC

Use case chức năng mô tả tác vụ chính

| Tác nhân | Vai trò / Chức năng chính |
|----------------------------------|--------------------------------------------------------------------------------|
| Driver (Người lái xe) | Tương tác với hệ thống để xem chỗ trống, nhận cảnh báo, vào/ra bãi đỗ. |
| Sensor / Camera | Thiết bị đầu cuối gửi dữ liệu cảm biến và biển số xe (telemetry) lên hệ thống. |
| Parking Staff (Nhân viên bãi xe) | Giám sát hệ thống, điều khiển barie thủ công, theo dõi trạng thái chỗ đỗ. |
| Actuator (Barrier) | Nhận lệnh điều khiển từ hệ thống để mở/đóng cổng tự động. |
| Parking Manager (Quản lý bãi xe) | Thiết lập quy tắc, theo dõi báo cáo, phân quyền người dùng. |
| IT Admin (Quản trị hệ thống) | Cấu hình thiết bị, cập nhật firmware, bảo trì và quản lý tài khoản. |

| Mã | Tên Use Case |
|-------------------------------------|------------------------------------------------------------------------------|
| U1 – View Real-time Parking Status | Hiển thị trạng thái từng chỗ đỗ (trống / có xe) theo thời gian thực. |
| U2 – Receive Alerts / Notifications | Gửi cảnh báo khi bãi đầy, cảm biến lỗi, hoặc có sự cố bất thường. |
| U3 – Manual Barrier Control | Cho phép nhân viên mở/đóng barie thủ công trong trường hợp đặc biệt. |
| U4 – Set Thresholds & Rules | Quản lý thiết lập ngưỡng (ví dụ: số chỗ trống tối thiểu) và quy tắc cảnh báo |
| U5 – Schedule Operations | Lập lịch vận hành barie hoặc đèn báo (theo giờ cao điểm, ban đêm...). |
| U6 – Historical Trends & Reports | Xem thống kê lượng xe ra/vào, thời gian đỗ trung bình, biểu đồ hoạt động. |
| U7 – Device Onboarding & Config | Thêm mới, cấu hình cảm biến, camera và gateway trong hệ thống. |
| U8 – OTA Firmware Update | Cập nhật phần mềm thiết bị từ xa để bảo trì và vá lỗi. |
| U9 – User & Role Management | Quản lý tài khoản, phân quyền truy cập (quản lý, nhân viên, khách). |

4. THIẾT KẾ LOGIC

Phân tách hệ thống thành các tầng logic:

Perception Layer

- Thu thập và xử lý dữ liệu cảm biến (SRF05, Camera ANPR, ESP32, Servo).
- Đóng vai trò đầu vào của hệ thống.

Network Layer

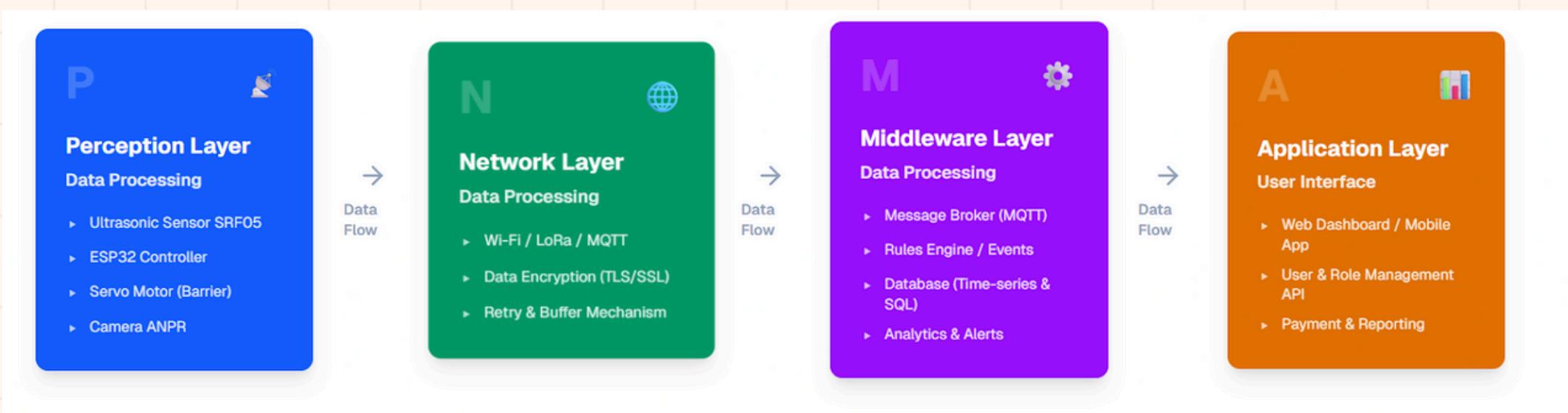
- Truyền dữ liệu qua Wi-Fi/LoRa/MQTT.
- Mã hóa bảo mật (TLS/SSL), có cơ chế retry & buffer.

Middleware Layer

- Trung gian xử lý: Message Broker (MQTT), Rules Engine, Database.
- Phân tích, cảnh báo, quản lý sự kiện.

Application Layer

- Giao diện người dùng: Web, API quản lý người dùng
- Hiển thị, giám sát và điều khiển hệ thống.



4. THIẾT KẾ LOGIC

Thiết kế luồng dữ liệu (data flow)

Luồng hoạt động hệ thống IoT bãi đỗ xe

- Cảm biến / Camera thu thập dữ liệu (trạng thái chỗ, biển số).
- ESP32 Controller gửi dữ liệu lên MQTT Broker (qua TLS bảo mật).
- MQTT Broker lưu dữ liệu vào Time-series DB và truyền tới Rules Engine.
- Rules Engine phân tích, phát hiện sự kiện (bãi đầy, lỗi cảm biến).
- Dashboard / Web hiển thị thông báo, cho phép người dùng gửi lệnh mở barrier.
- Lệnh điều khiển được gửi ngược qua MQTT → ESP32 → Servo Motor thực thi.
- Hệ thống phản hồi (ACK) và cập nhật trạng thái lên giao diện.

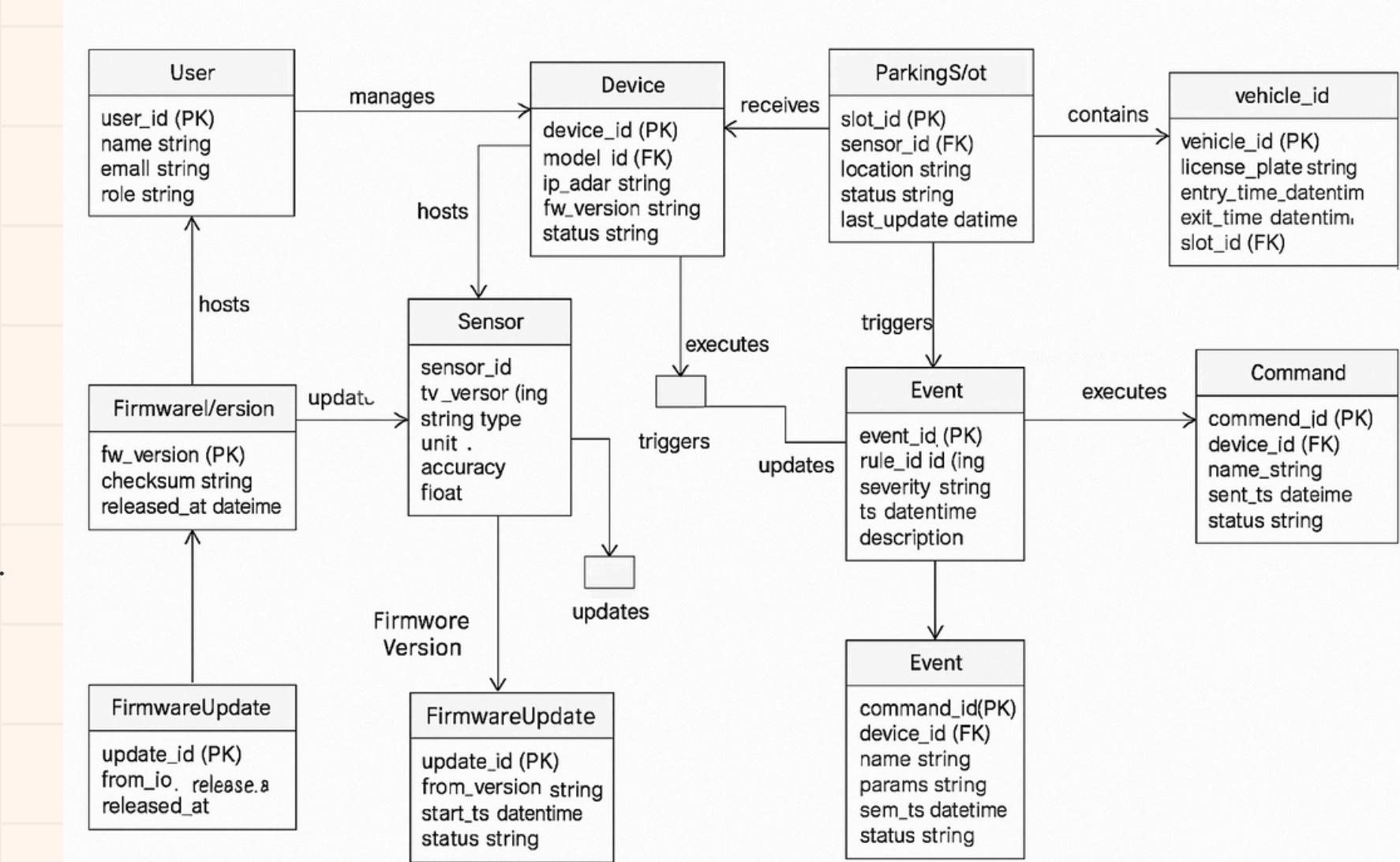


4. THIẾT KẾ LOGIC

Thiết kế luồng dữ liệu (data flow)

Quan hệ giữa các bảng

- User – Device: 1-N → Một người dùng có thể quản lý nhiều thiết bị.
- User – ParkingSlot: 1-N → Một người dùng có thể giám sát nhiều ô đỗ.
- Device – Sensor: 1-N → Một thiết bị có thể chứa nhiều cảm biến.
- Sensor – Telemetry: 1-N → Một cảm biến tạo ra nhiều dữ liệu đo (telemetry).
- ParkingSlot – Sensor: 1-1 → Mỗi ô đỗ gắn với một cảm biến.
- ParkingSlot – Vehicle: 1-N → Một ô đỗ có thể chứa nhiều xe theo thời gian.
- Device – Event: 1-N → Một thiết bị có thể phát sinh nhiều sự kiện.
- Device – Command: 1-N → Một thiết bị có thể nhận nhiều lệnh điều khiển.
- Device – FirmwareUpdate: 1-N → Một thiết bị có thể có nhiều lần cập nhật firmware.
- FirmwareVersion – FirmwareUpdate: 1-N → Một phiên bản firmware có thể được dùng trong nhiều lần cập nhật.

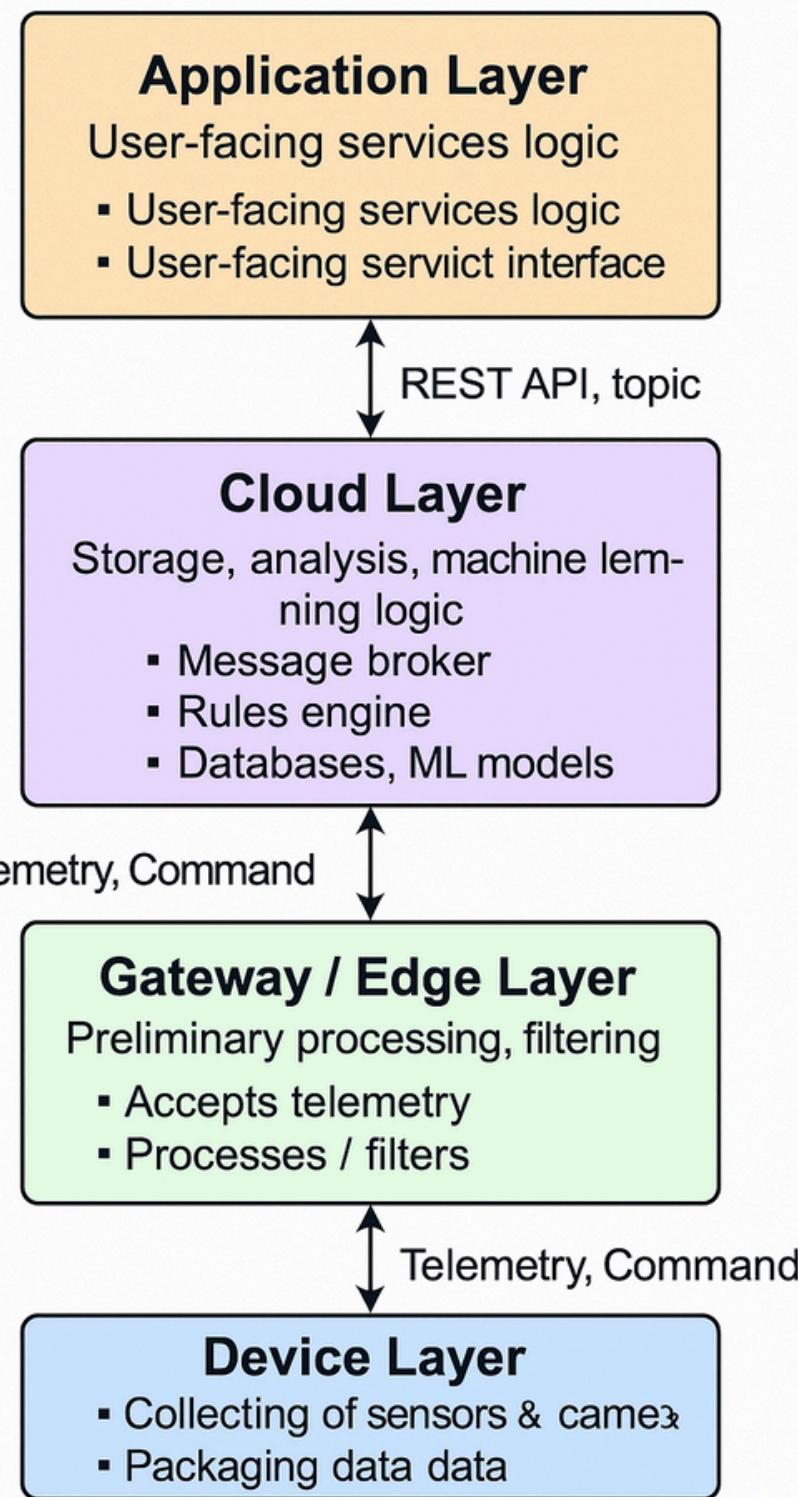


4. THIẾT KẾ LOGIC

Thiết kế kiến trúc logic hệ thống

Cấu trúc 4 tầng:

- Device Layer:
 - Thu thập dữ liệu từ cảm biến, camera...
 - Đóng gói và gửi dữ liệu thô (telemetry).
- Gateway/Edge Layer:
 - Xử lý sơ bộ, lọc nhiễu, hợp nhất dữ liệu.
 - Gửi dữ liệu hợp lệ lên đám mây.
- Cloud Layer:
 - Lưu trữ, phân tích, và học máy.
 - Quản lý luồng dữ liệu qua message broker, rules engine.
- Application Layer:
 - Cung cấp giao diện và dịch vụ cho người dùng.
 - Kết nối qua REST API, topic.

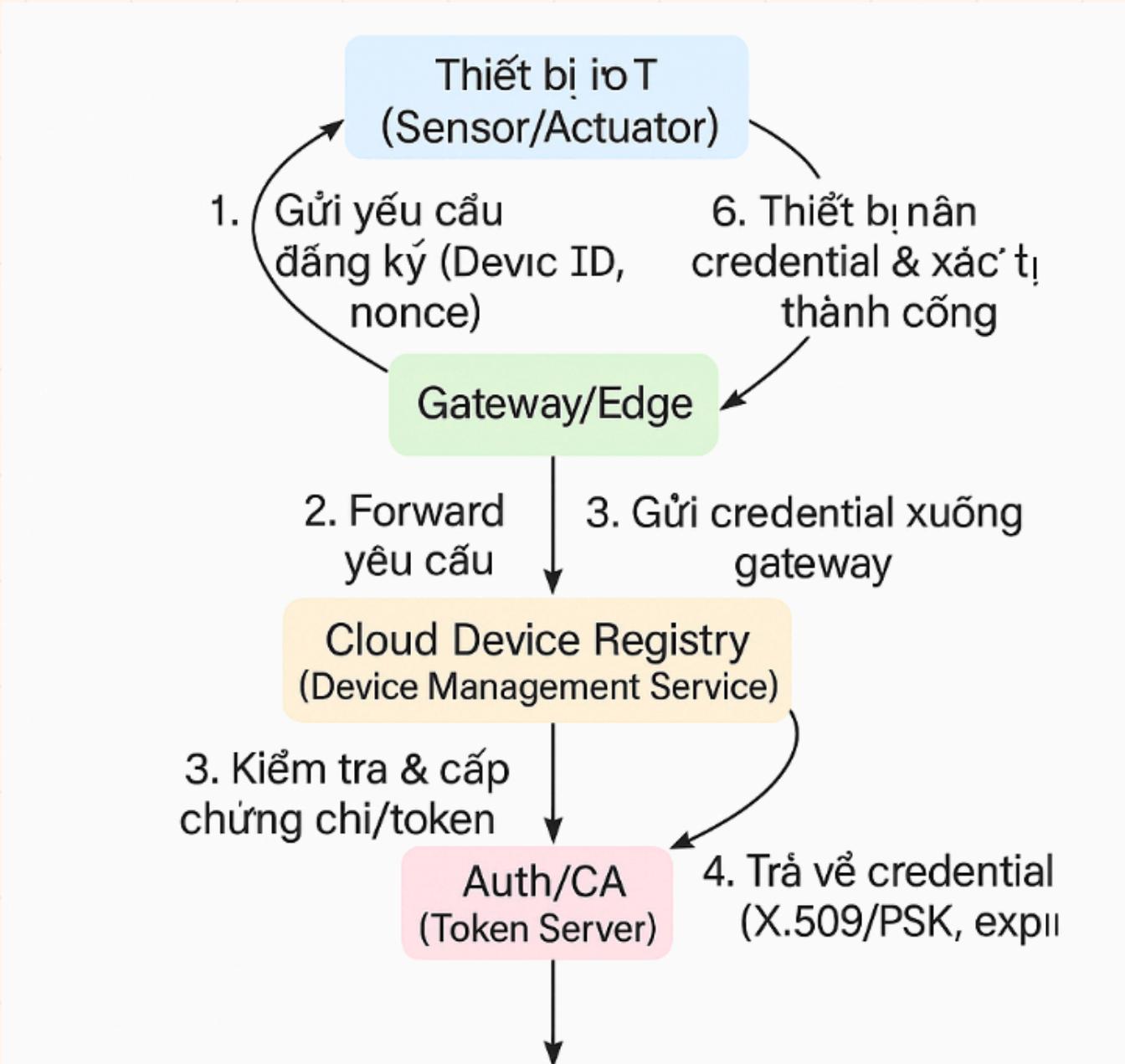


4. THIẾT KẾ LOGIC

Thiết kế bảo mật

Quy trình xác thực thiết bị:

- Thiết bị IoT gửi yêu cầu đăng ký (Device ID, nonce) lên Gateway.
- Gateway chuyển tiếp yêu cầu đến Cloud Device Registry.
- Cloud kiểm tra, yêu cầu Auth/CA cấp chứng chỉ hoặc token.
- Auth/CA trả về credential (X.509 certificate hoặc PSK).
- Cloud gửi credential xuống Gateway, rồi tới thiết bị.
- Thiết bị IoT nhận và xác thực thành công, hoàn tất đăng ký an toàn.



5. THIẾT KẾ VẬT LÝ

Lựa chọn và thiết kế thiết bị cảm biến

Cảm biến:

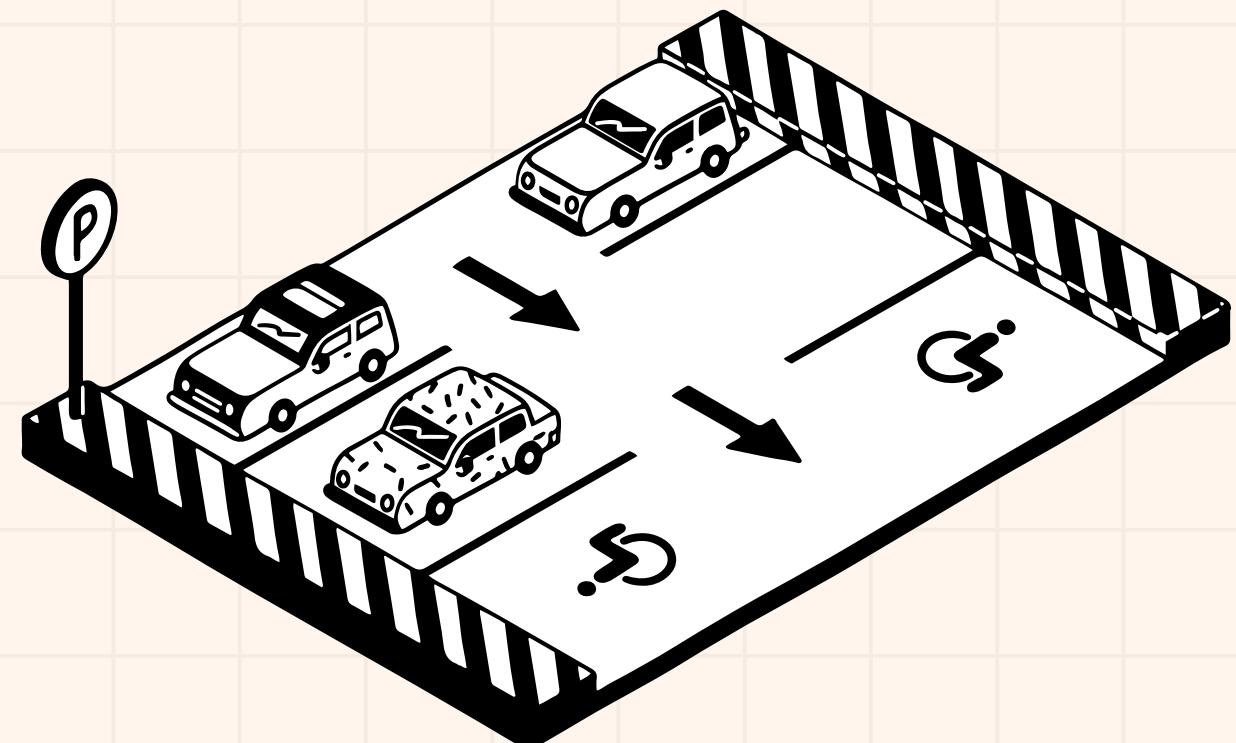
- SRF05 - Cảm biến siêu âm (dải đo 2-400cm, sai số 0.3cm) xác nhận trạng thái có xe/trống tùng chỗ
- Camera LPR nhận dạng biển số xe tại cổng ra vào
- Gắn thực địa:
 - SRF05 lắp trên trần hoặc phí trước ô đỗ, góc chiếu vuông góc
 - Camera LPR cố định tại cổng ra/vào, tầm nhìn rõ toàn biển số

Khối vi điều khiển:

- ESP32(Kit Wifi + Bluetooth, SH340/CP2102) kết nối cảm biến:
 - SRF05 qua GPIO digital
 - Camera LPR qua Ethernet/Wi-fi
- Nhiệm vụ logic:
 - Lấy mẫu cảm biến định kỳ
 - Xử lý dữ liệu, lọc nhiễu/debounce để xác định trạng thái chính xác
 - Đóng gói dữ liệu dạng JSON/CBOR và gửi MQTT → Blynk Cloud
 - Quản lý điều khiển servo barie(SG90) khi xe ra/vào

Khối cấp nguồn:

- Nguồn chính: DC 5V ổn định, cung cấp cho ESP32, SRF05, SG90



5. THIẾT KẾ VẬT LÝ

Lựa chọn và thiết kế kết nối mạng

- Công nghệ truyền thông: Wi-fi(ESP32 kết nối Internet trực tiếp, phù hợp bâi xe có hạ tầng mạng)
- Giao thức dữ liệu: MQTT - nhẹ, hỗ trợ publish/subscribe , dễ tích hợp với Blynk Cloud
- Bảo mật: TLS/SSL cho MQTT + xác thực token/JWT



5. THIẾT KẾ VẬT LÝ

Thiết kế hạ tầng xử lý, lưu trữ

Edge/Fog computing:

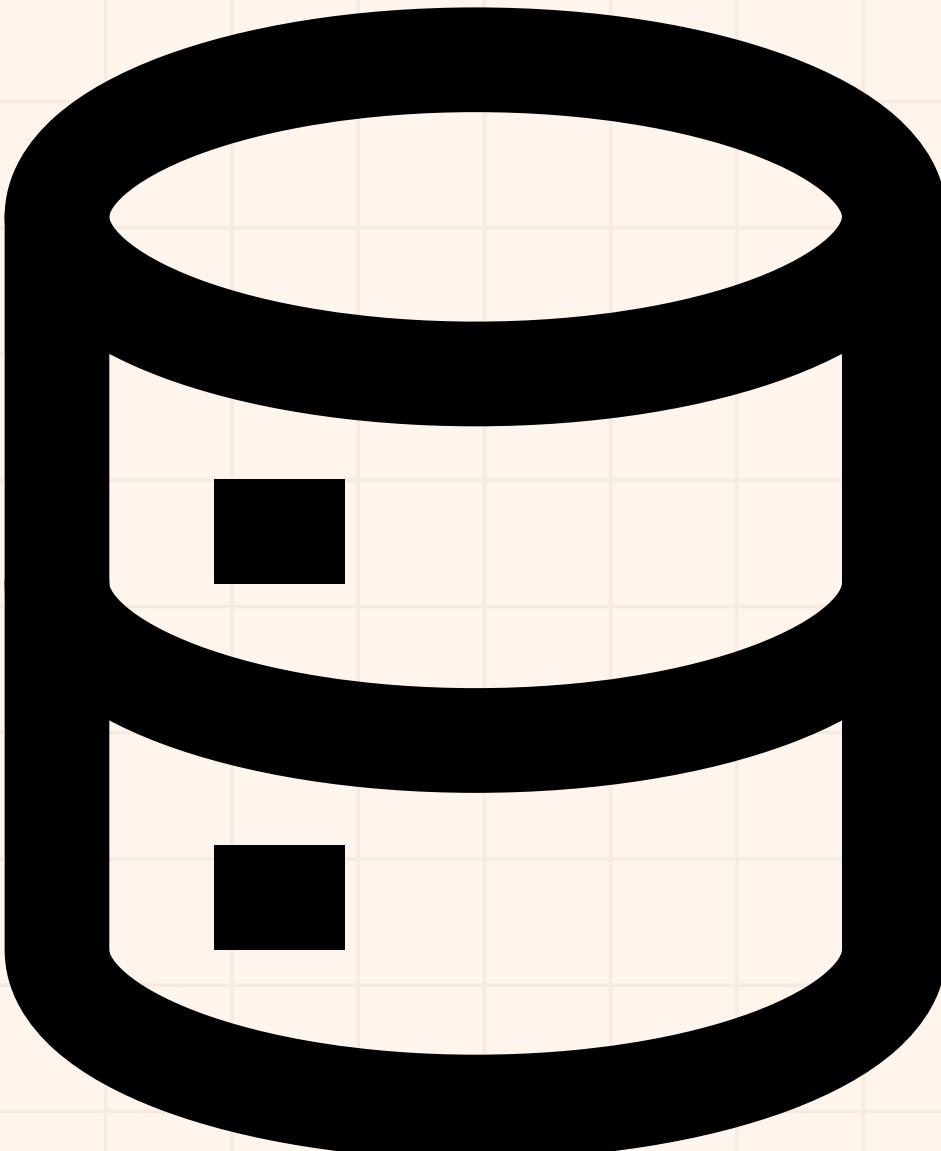
- Gateway(ESP32 hoặc server trung gian) xử lý sơ bộ dữ liệu từ cảm biến, lọc nhiễu, xác nhận trạng thái ô đỗ
- Giảm tải cho Cloud, tăng tốc độ phản hồi

Cloud:

- Blynk Cloud đảm nhận xử lý backend, quản lý luồng dữ liệu, sự kiện vào/ra bãi và giao tiếp Web Application

Cơ sở dữ liệu:

- Lưu lịch sử trạng thái ô đỗ, lịch sử ra vào và biển số xe
- SQL: SQLite



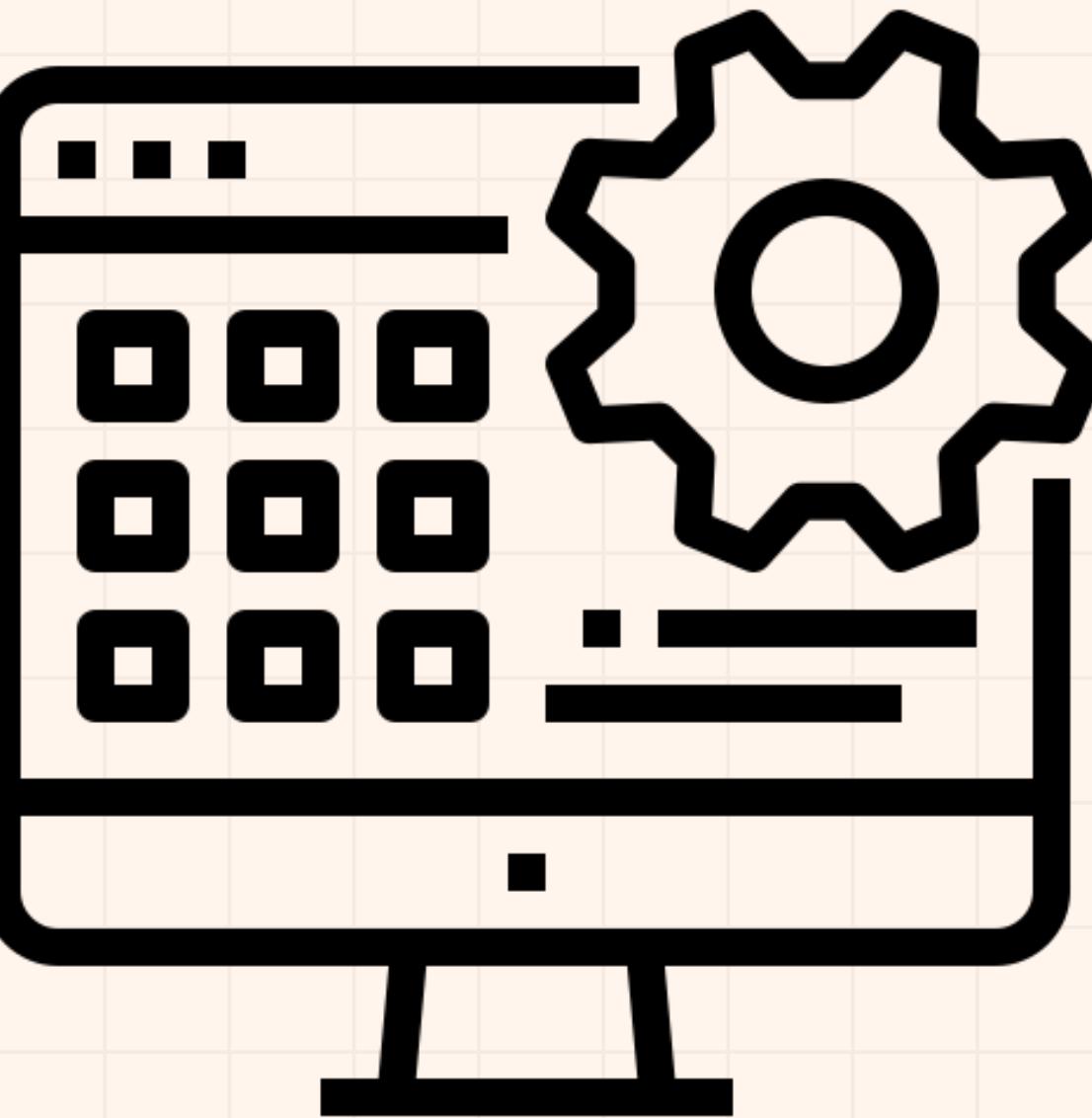
5. THIẾT KẾ VẬT LÝ

Thiết kế ứng dụng người dùng

Mục tiêu: cung cấp ứng dụng cho người dùng để xem thông tin bãi xe, vị trí chỗ trống giúp vào/ra bãi thuận tiện hơn

Thành phần chính:

- Giao diện người dùng gồm các màn hình: đăng nhập
đăng kí, hiển thị chỗ trống, lịch sử gửi xe
- Luồng hoạt động: mở app → đăng nhập → lấy dữ liệu
từ server, hiển thị chỗ trống → hệ thống nhận diện
biển số xe → lưu thông tin sau khi đỗ xe tại bãi

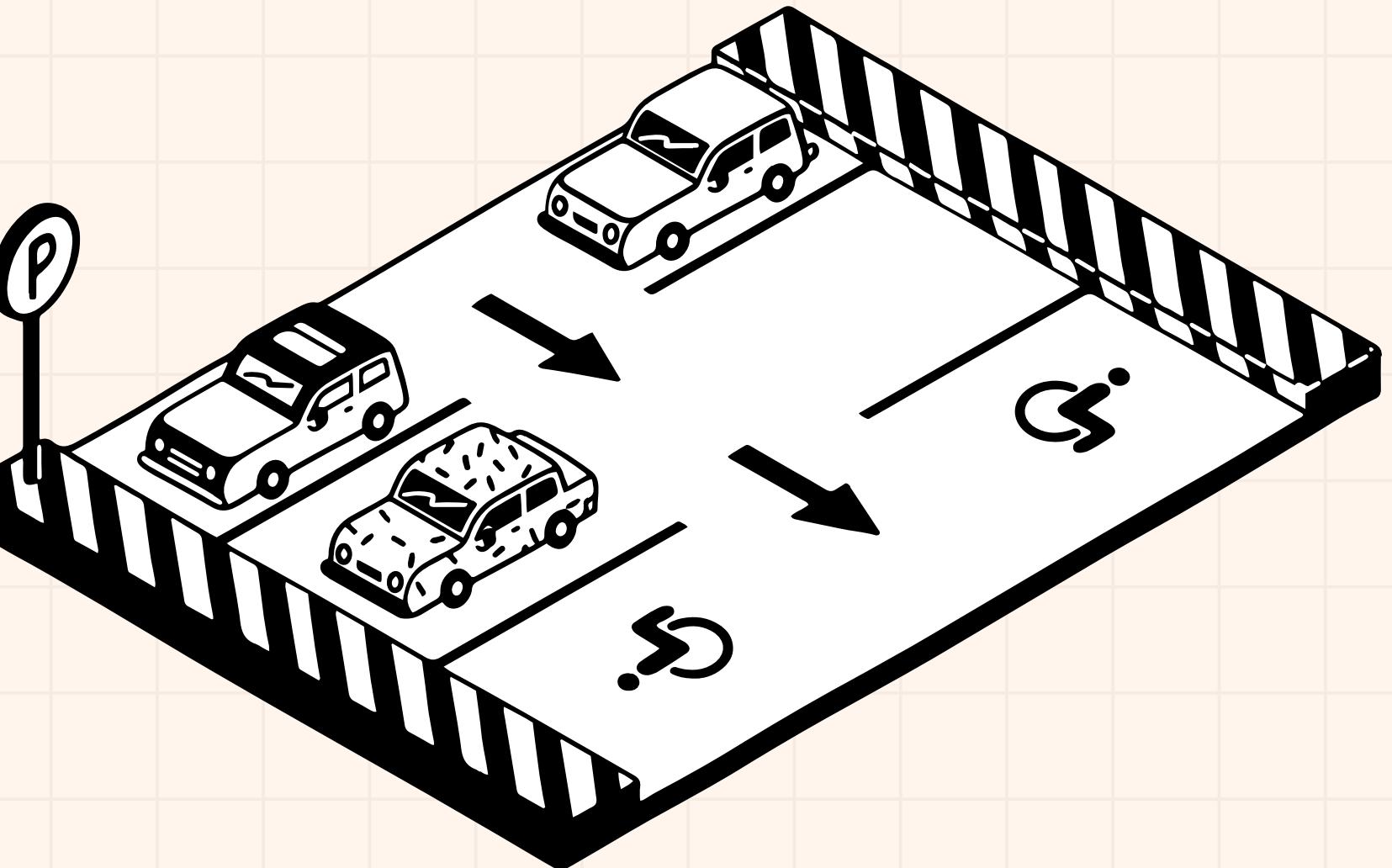


3. THIẾT KẾ VẬT LÝ

Thiết kế ứng dụng người dùng

Các công nghệ sử dụng

- Frontend: ReactJS
- Backend: NodeJS
- Database: SQLite
- Python service: Python, OpenCV, YOLOv5/YOLOv8



5. THIẾT KẾ VẬT LÝ

Thiết kế bảo mật và quản lý

Bảo mật:

- Xác thực và phân quyền: sử dụng JWT token
- Mã hóa dữ liệu: hash password

Quản lí thiết bị và dữ liệu:

- Gateway IoT: nhận dữ liệu từ camera và ESP32
- Camera nhận diện biển số xe, chuyển dữ liệu hình ảnh hoặc kết quả nhận diện về Gateway.

- Cảm biến vị trí đồ: gửi kết quả đầy/trống về gateway
- Data được lưu trên server: đầy đủ, an toàn



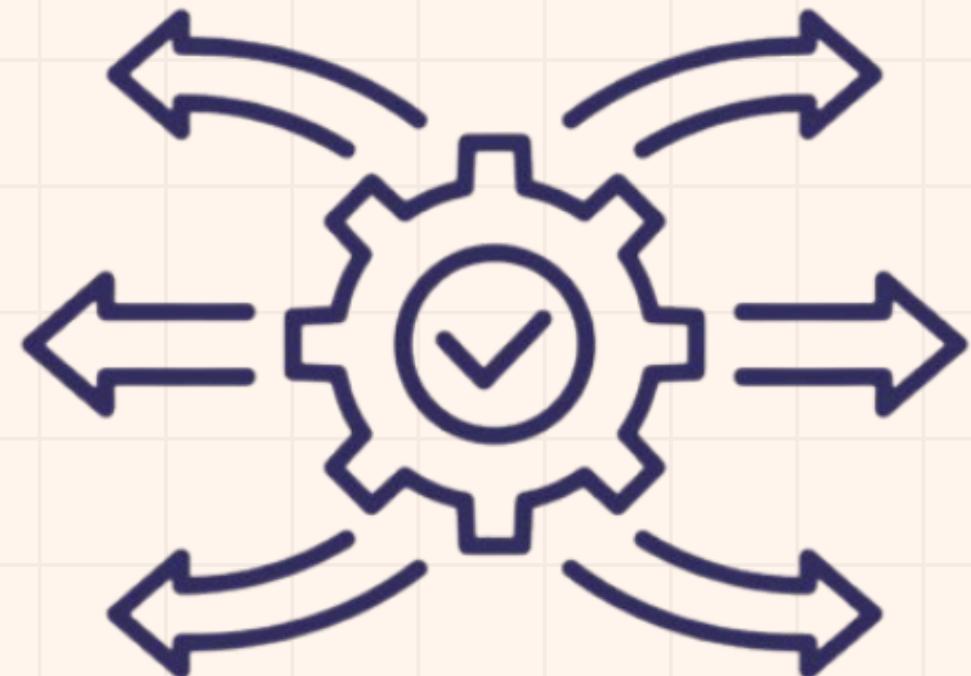
5. THIẾT KẾ VẬT LÝ

Thiết kế triển khai và vận hành

Mục tiêu: mô tả hệ thống được triển khai trong thực tế

Triển khai:

- Hệ thống triển khai theo mô hình Client–Server kết hợp IoT.
- Camera nhận diện biển số và cảm biến (ESP32) gửi trạng thái chỗ trống.
- Gateway IoT thu thập và xử lý sơ bộ dữ liệu từ thiết bị → gửi lên Server.
- Server Backend + CSDL lưu trữ tập trung để dễ quản lý và truy xuất lịch sử.
- Quy trình vận hành: Xe vào bãi → nhận diện biển số → kiểm tra → mở barie
→ xe rời bãi.



6. KẾT QUẢ ĐẠT ĐƯỢC

- Hệ thống đã thu thập được dữ liệu từ 5 cảm biến siêu âm và cập nhật trạng thái chỗ trống theo chu kỳ.
- Servo điều khiển barie hoạt động ổn định, đóng/mở theo logic hệ thống.
- Hoàn thiện kết nối phần cứng → Gateway (ESP32/Raspberry Pi) → Server cơ bản.
- Camera nhận diện biển số xe (LPR) dựa trên YOLO + OpenCV + OCR.
→ Đang hoàn thành



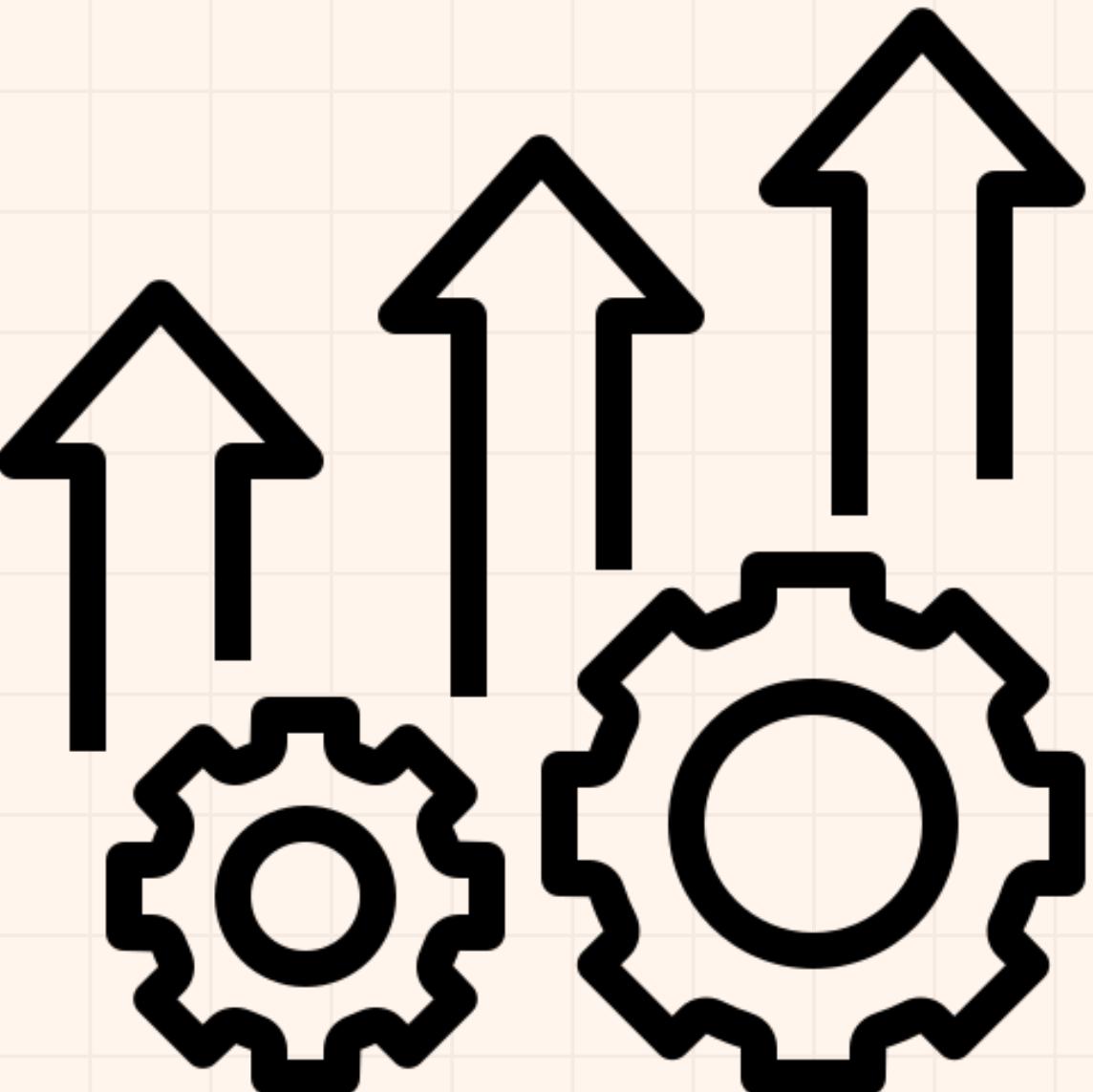
6. KẾT LUẬN VÀ HƯỚNG MỞ RỘNG

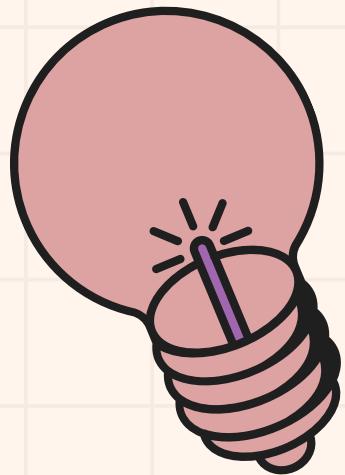
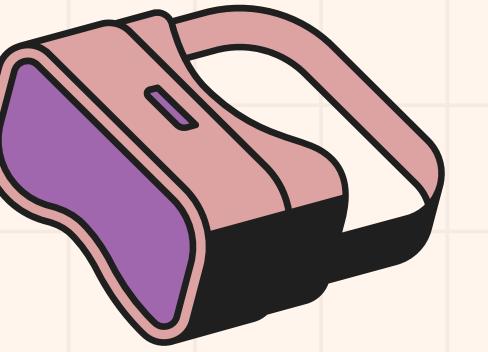
Kết luận:

- Hệ thống đã tự động hóa được quy trình kiểm soát bãi đỗ xe.
- Giảm sự can thiệp thủ công, tăng độ chính xác và tính an toàn.
- Nhận diện biển số hỗ trợ quản lý ra/vào chính xác và lưu vết dài hạn.

Hướng mở rộng:

- Tích hợp triển khai trên quy mô lớn
- Thêm thanh toán tự động qua QR/ví điện tử hoặc trừ thẳng vào tài khoản người dùng sau mỗi lần gửi
- Tích hợp thêm bản đồ điều hướng ở trong bãi → thuận tiện hơn trong việc tìm chỗ





THANK YOU



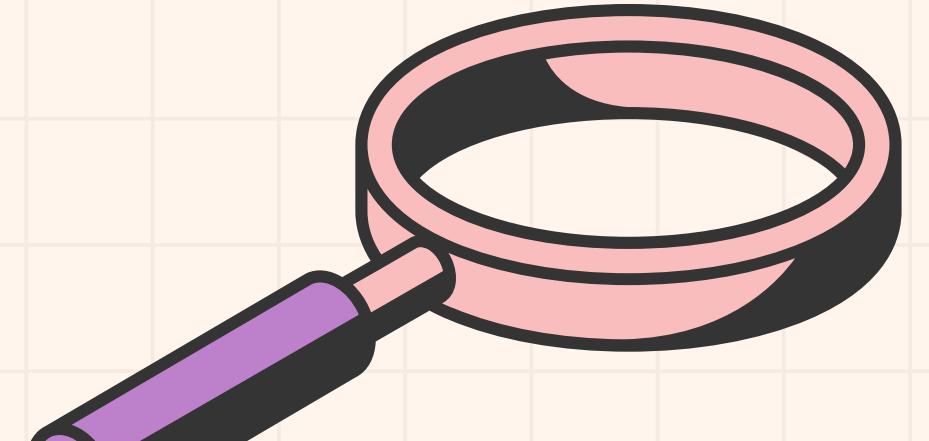
+123-456-7890



WWW.REALLYGREATSITE.COM



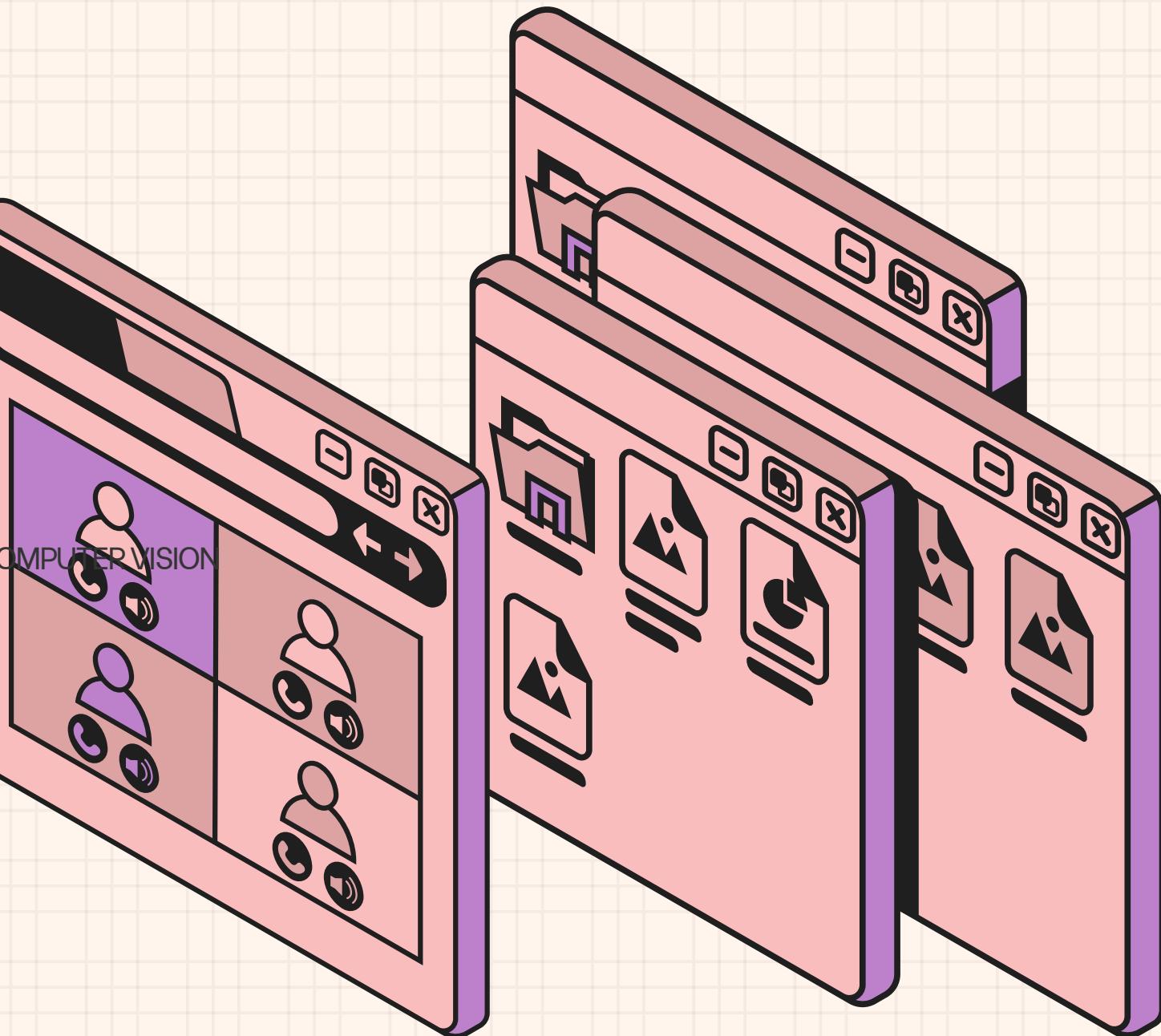
123 ANYWHERE ST., ANY CITY



SOFTWARE COMPONENTS

A project refers to a sequence of tasks that are carefully planned and executed to achieve a specific objective. Projects are essential in various fields, such as business, engineering, and education, and often require collaboration and resources. Each project has a start and an end, along with defined goals and deliverables. The success of a project is usually determined by how effectively the team meets the objectives within the allocated time and resources.

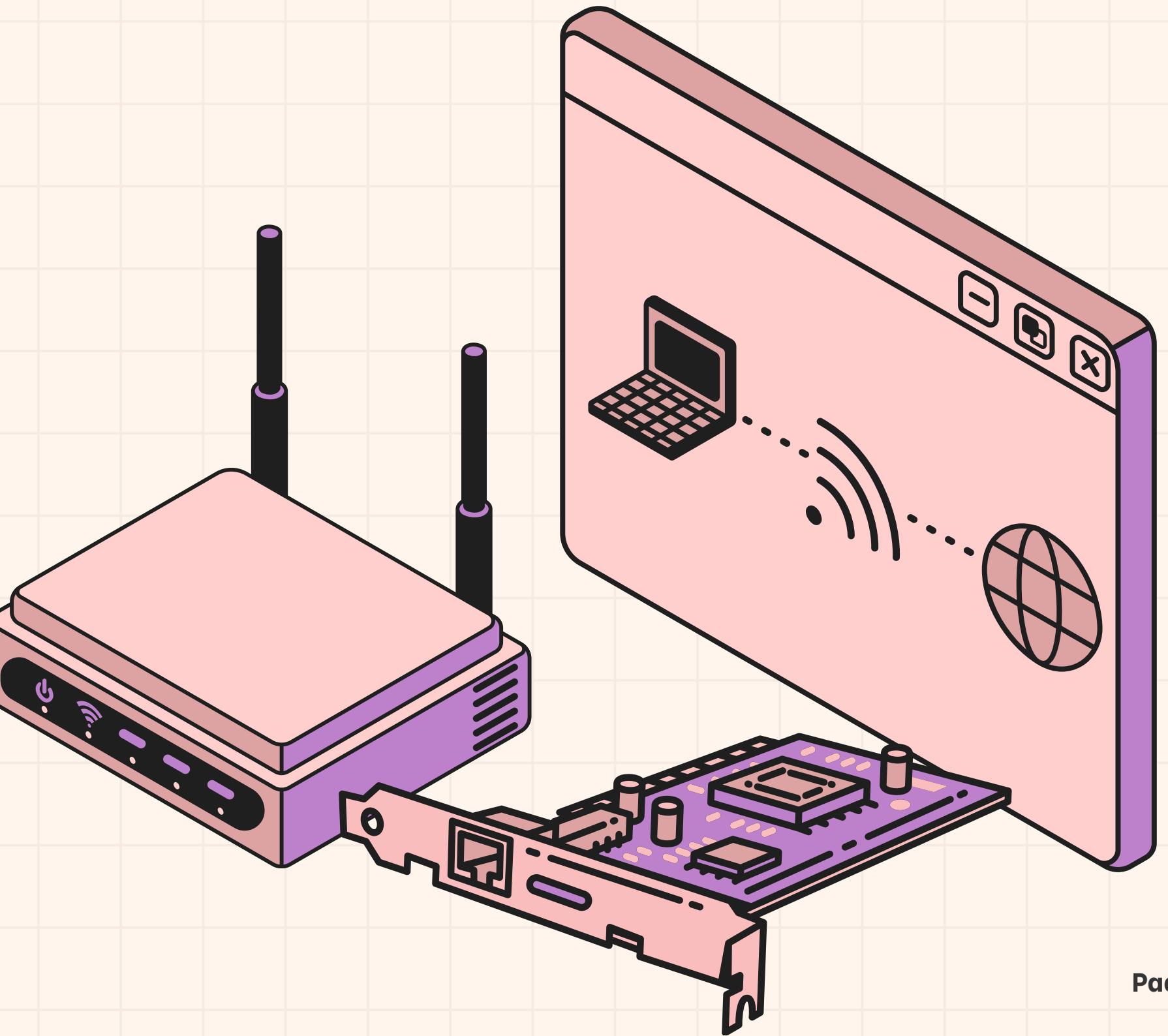
A project refers to a sequence of tasks that are carefully planned and executed to achieve a specific objective. Projects are essential in various fields, such as business, engineering, and education, and often require collaboration and resources.



NETWORKING & THE INTERNET

A project is a series of planned tasks with a clear objective, often involving multiple steps. It can be carried out by individuals or teams and is designed to achieve specific outcomes within a set timeframe.

A project is a series of planned tasks with a clear objective, often involving multiple steps. It can be carried out by individuals or teams and is designed to achieve specific outcomes within a set timeframe.



DATA MANAGEMENT

T



DATABASE

A project is a series of planned tasks with a clear objective, often involving multiple steps. It can be carried out by individuals or teams and is designed to achieve specific outcomes within a set timeframe.



DATA & SECURITY

A project is a series of planned tasks with a clear objective, often involving multiple steps. It can be carried out by individuals or teams and is designed to achieve specific outcomes within a set timeframe.



DATA ANALYTICS

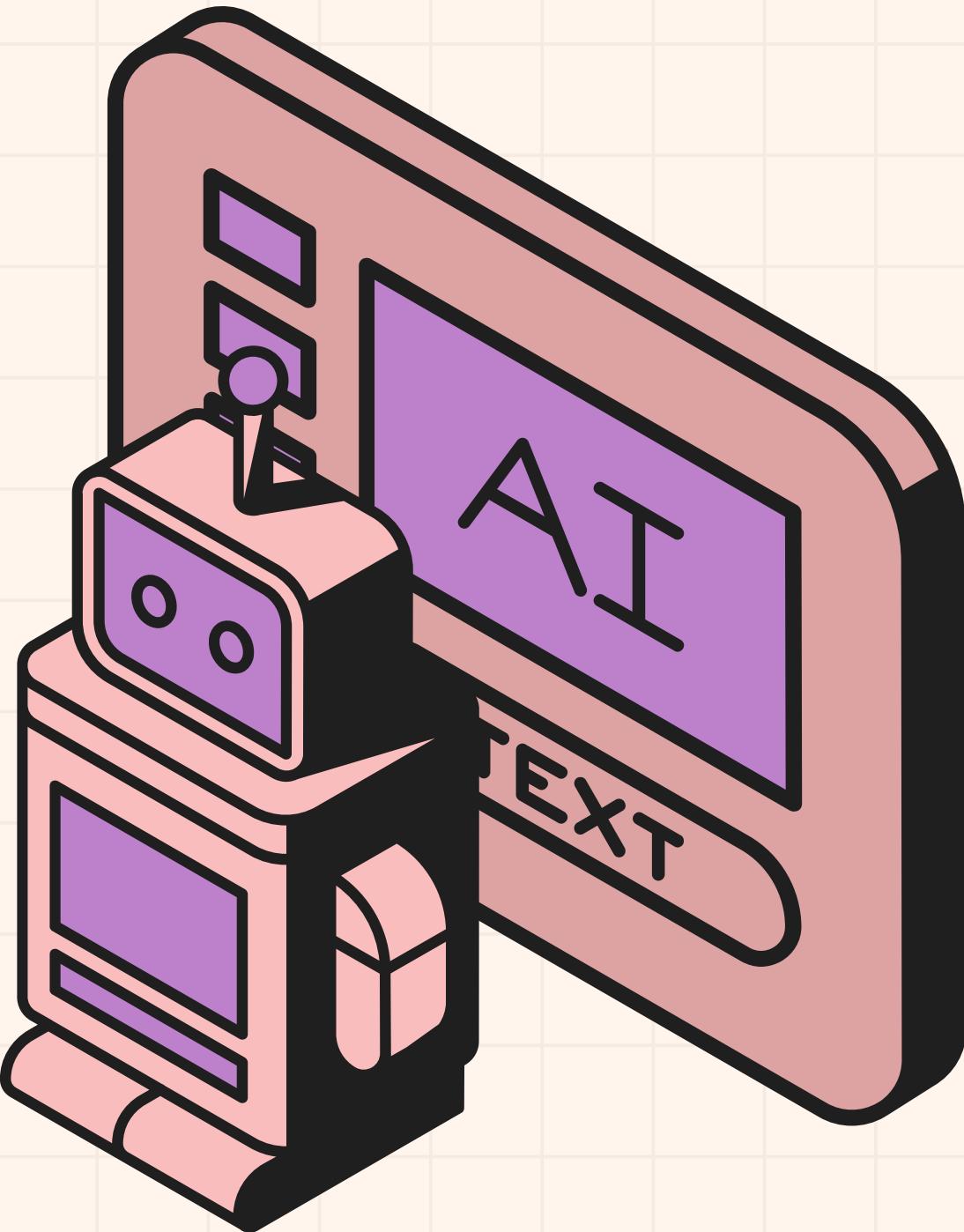
A project is a series of planned tasks with a clear objective, often involving multiple steps. It can be carried out by individuals or teams and is designed to achieve specific outcomes within a set timeframe.



ARTIFICIAL INTELLIGENCE

A project is a series of planned tasks with a clear objective, often involving multiple steps. It can be carried out by individuals or teams and is designed to achieve specific outcomes within a set timeframe.

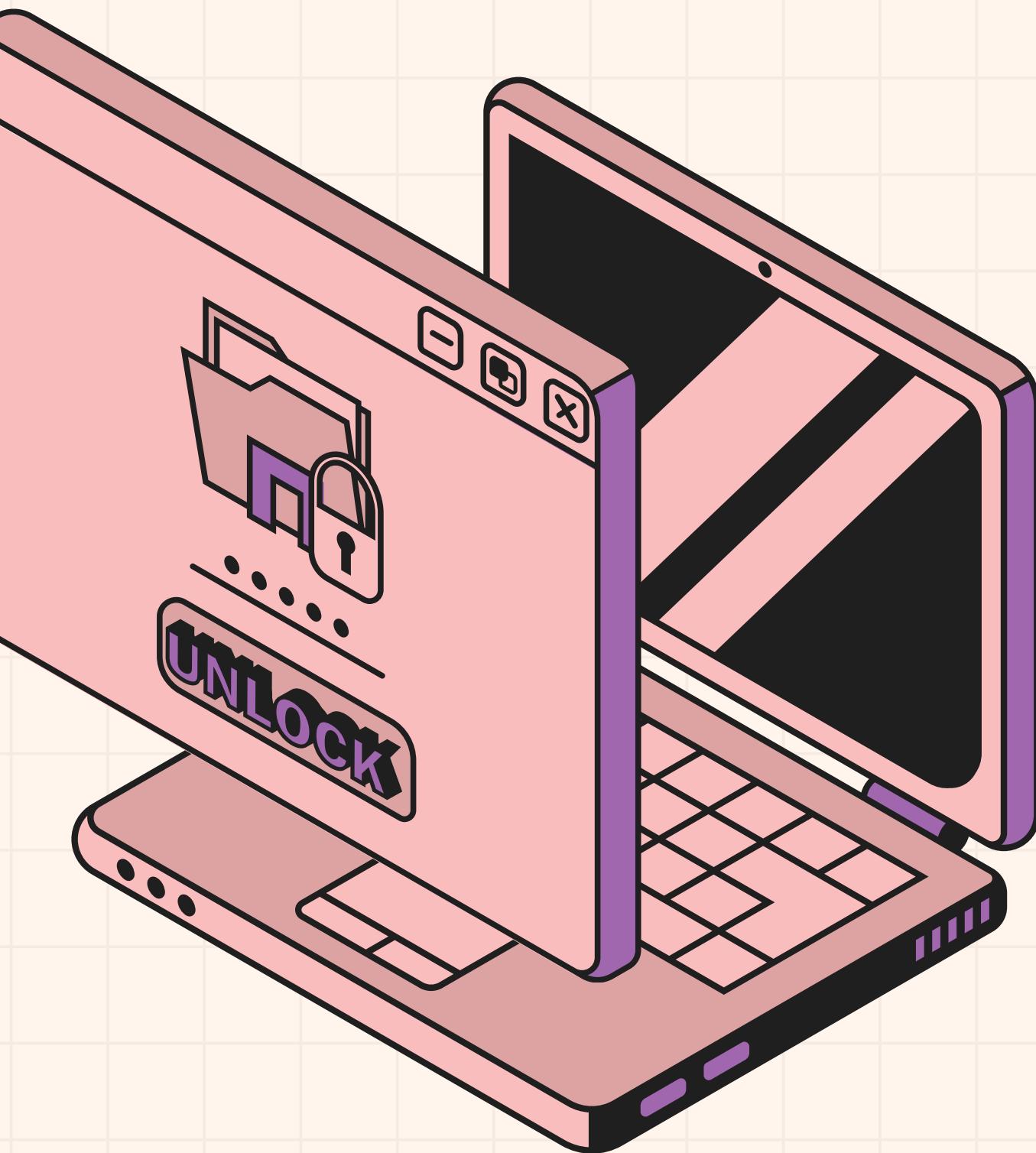
A project is a series of planned tasks with a clear objective, often involving multiple steps. It can be carried out by individuals or teams and is designed to achieve specific outcomes within a set timeframe.



CYBER SECURITY

A project is a series of planned tasks with a clear objective, often involving multiple steps. It can be carried out by individuals or teams and is designed to achieve specific outcomes within a set timeframe.

A project is a series of planned tasks with a clear objective, often involving multiple steps. It can be carried out by individuals or teams and is designed to achieve specific outcomes within a set timeframe.



FUTURE IN TECHNOLOGY

A project is a series of planned tasks with a clear objective, often involving multiple steps. It can be carried out by individuals or teams and is designed to achieve specific outcomes within a set timeframe.

A project is a series of planned tasks with a clear objective, often involving multiple steps. It can be carried out by individuals or teams and is designed to achieve specific outcomes within a set timeframe.



COMPUTER TECHNOLOGY

A project is a series of planned tasks with a clear objective, often involving multiple steps. It can be carried out by individuals or teams and is designed to achieve specific outcomes within a set timeframe.

