



AUTOMOTIVE HACKATHON / 2026

SHAPE THE AI-DEFINED MOBILITY ERA



Đối tượng được tham dự



Học sinh, sinh viên và kỹ sư IT đam mê về Automotive từ 16 tuổi trở lên có quốc tịch Việt Nam



đội ngũ Ban tổ chức hoặc tham gia đại diện cho công ty/doanh nghiệp/tổ chức thứ 3 sẽ không được đăng ký tham gia cuộc thi này.

Tối thiểu 2 & tối đa 5 thành viên/ Đội

Nếu bạn đã có đội? Đăng ký:



Chưa có đội? Cần FPT ghép đội



VÒNG THI VÀ QUY TRÌNH

01

Vòng 1

Đăng ký đội

Từ nay đến 12.07

Các đội hoàn thành đăng ký và nhận được email confirm từ Ban tổ chức. Ban tổ chức sẽ gửi email cho Đội trưởng để đăng ký bài tập & tham khảo Template nộp Ý tưởng do FPT cung cấp

02

Vòng 1

Nộp proposal/idea

Từ ngày đăng ký đội đến 12.07

Khi Ban tổ chức gửi email xác nhận đội, các đội sẽ nhận được 1 link đăng ký chọn bài tập & 1 link template nộp ý tưởng. Các đội cần hoàn thiện và gửi cho BTC trước ngày 12.07.2026

03

Vòng 2

20 Đội thi trên Virtual Development Platform.

21.07.2026 - 10.08.2026

Ban tổ chức sẽ cung cấp tài khoản đăng nhập nền tảng cho 20 đội có ý tưởng tốt nhất để tham dự Vòng 2. 20 đội sẽ được add vào nhóm chat của cuộc thi và có mentor đồng hành cùng để giải đáp trong suốt cuộc thi.

04

Vòng Chung kết

5 đội xuất sắc tham gia Hackathon Offline 2N1Đ

22.08.2026 - 23.08.2025

5 đội xuất sắc từ vòng 2 sẽ tham gia trực tiếp tại Hà Nội. Ban tổ chức hỗ trợ 5M/1 đội và hỗ trợ chi phí lưu trú, ăn uống trong suốt quá trình 2N1Đ diễn ra cuộc thi

Đã có đội? Đăng ký tại đây



Chưa có đội? Cần ghép?



GIẢI THƯỞNG

Tổng giá trị giải thưởng: 250 triệu VNĐ



Giải Nhất

100 triệu đồng



Giải Nhì

80 triệu đồng



3 Giải Khuyến khích

20 triệu đồng



Bình chọn nhiều nhất

10 triệu đồng

CARSKY

Virtual Engineering Platform

Xây dựng kiến trúc xe với head units, ECUs, networks hoàn toàn bằng phần mềm trước khi phần cứng tồn tại và áp dụng AI tự động cho các bài kiểm thử.



MR. NGUYỄN QUỐC HÙNG

Senior Automotive Engineer, FPT Automotive

MỘT CÂU CHUYỆN CÓ THẬT (GẦN NHƯ VẬY)



Một kỹ sư thay đổi một dòng code của sản phẩm.

Để kiểm tra, anh ta cần một bàn ECU, một bộ dây điện,
và một bo mạch được giao trong ...
“tháng sau”, “quý sau”

Và anh ta đợi. Rồi lại đợi. công việc và đồng đội của anh
ấy cũng đợi như vậy.



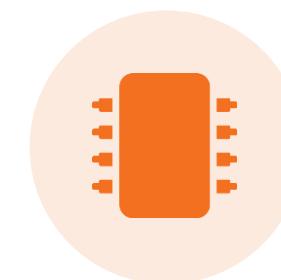
Bài nói này là về việc trả lại những thời gian chờ đợi đó cho anh ta

Tại sao tốc độ phát triển phần mềm ô tô lại chậm như vậy



Nút Thắt Cổ Chai Phần Cứng

Không có bo mạch khi bắt đầu dự án, không giờ đủ bo mạch trong giai đoạn test sản phẩm. Thời gian phát triển bị lãng phí vì chờ đợi, không phải vì lập trình.



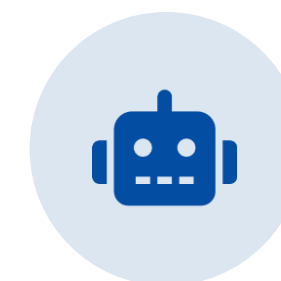
Bị Khóa SoC & Chi Phí BSP

Silicon mới = BSP mới = nhiều tháng chuyển đổi. OEM bị phụ thuộc vào lịch trình của nhà sản xuất chip.



Tích Hợp Muộn, Phát hiện lỗi muộn

Kiểm tra toàn hệ thống chỉ khi phần cứng về đủ. Lỗi xuất hiện ở phân tích hợp càng dẫn đến chi phí sửa chữa cao hơn nhiều.



AI no hand

AI có thể viết code, nhưng không thể triển khai hay xác thực trên phần cứng. Con người vẫn phải cầu nối mọi vòng lặp.

GIẢI PHÁP



Một nền tảng để định nghĩa cấu trúc, liên kết, ảo hóa ECU, mô phỏng mạng CAN, LIN, Ethernet, SOME/IP và chạy tự động hóa kiểm thử bằng AI tất cả từ trình duyệt web.

Kỹ sư thiết kế, triển khai và xác thực các hệ thống đa ECU phức tạp mà không cần phần cứng vật lý — rút ngắn nhiều tuần trong mỗi chu kỳ phát triển.

1. Định nghĩa



2. Ảo hóa



3. Mô phỏng



4. Xác thực

“Toàn bộ thiết kế xe được xử lý bằng phần mềm, trước khi phần cứng tồn tại.”

CÁCH HOẠT ĐỘNG

Ba năng lực tích hợp



CabinSky

RUNTIME ECU ẢO

- Phần mềm vECU (Android, AGL, Linux, AUTOSAR) chạy như container trên cloud
- Không cần bo mạch, không cần cáp kết nối hay flash
- Một image chung cho mọi SoC — Qualcomm, NXP, Telechips
- Không bị chặn bởi BSP. Zero chi phí chuyển đổi



Nyduz

ĐIỀU PHỐI TOPOLOGY

- Định nghĩa mạng lưới xe một lần: CAN, LIN, Ethernet, SOME/IP
- Hoán đổi độ trung thực mỗi nút: mock → vECU → HIL thật
- Không cần viết lại bài kiểm tra
- Tích hợp dSPACE, Vector CANoe & CI/CD đã tồn tại trước đây



MCP Interface

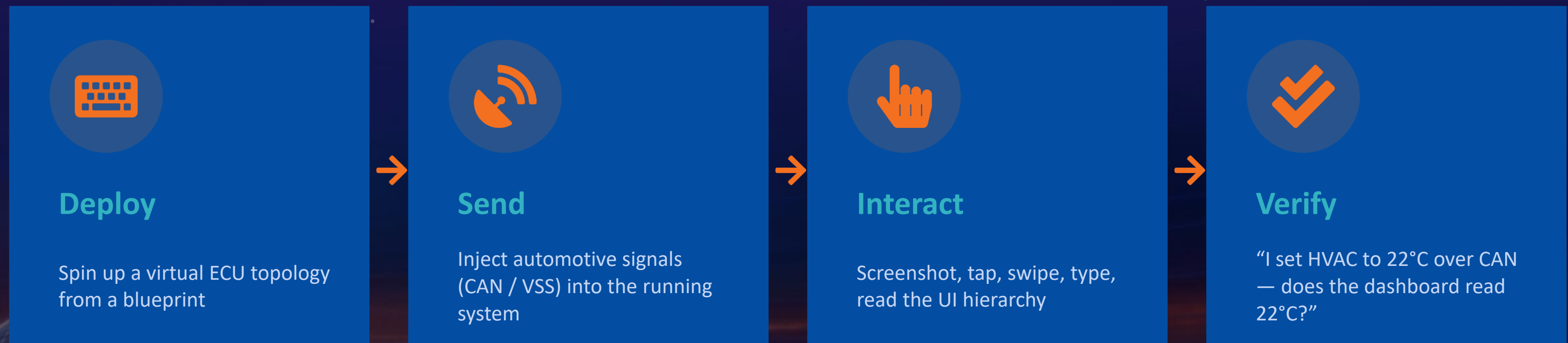
PHÁT TRIỂN CÙNG AI

- Tác nhân AI triển khai, kiểm thử, gỡ lỗi & tái triển khai — không cần giao diện
- Hoạt động với Claude Code, Cursor, Codex hoặc công cụ của bạn
- Vòng CI/CD hoàn chỉnh 24/7, không cần con người trong vòng lặp
- Kỹ sư xem xét kết quả, không thực hiện từng bước

CAPABILITY 3 · MCP INTERFACE



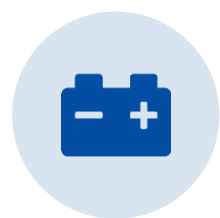
An AI writes the test, deploys the car,
drives the dashboard — and grades itself.



The differentiator no traditional automotive tool has: 46 REST endpoints consolidated into ~15 AI-friendly MCP tools. The agent is the orchestrator.

Bốn demo, một nền tảng

DEMO 1



Luồng HVAC

ECU → IVI. Xác thực tín hiệu đầu cuối: vSoC → CAN → LIN → GPIO.

DEMO 2



Cụm AGL

Tốc độ trực tiếp trên cụm đồng hồ
AGL: vECU → CAN → KUKSA → HMI.

DEMO 3



Kịch bản TCU

Mở khóa cửa từ xa & chẩn đoán. Di động → cloud → NAD → MCU → CAN → quay lại.

DEMO 4



ECU Dây An Toàn

ECU dây an toàn AUTOSAR phản ứng với trạng thái dây qua CAN; cảnh báo bảng điều khiển cập nhật trực tiếp.



Demo

Một phần xe ảo thực sự, được triển khai trực tiếp

Mỗi demo là cùng một nền tảng — cấu trúc liên kết khác nhau, không cần công cụ mới.



Automotive
Hackathon 2025

THANK YOU!

FPT Software Company Limited

- FPT Cau Giay Building, Duy Tan Street, Cau Giay Ward, Hanoi City, Vietnam

Tel.: +84 (24) 3 768 9048

Fax: +84 (24) 3 768 9049

Đã có đội? Đăng ký tại đây



Chưa có đội? Cần ghép?



AI Automotive Hackathon 2026

Building AI Agents for
Automotive Engineering



MR. MR. TRẦN HỮU TUỆ

Senior AI Consultant, FPT Automotive

7

challenges

RAG · LLM · Agents

core technologies

Python

nền tảng chính

1

PHẦN 1 – AUTOMOTIVE AI VERTICAL

2

PHẦN 2 – 7 CHALLENGES

3

PHẦN 3 – STARTER PACK

4

PHẦN 4 – CHỌN ĐỀ PHÙ HỢP

PHẦN 1 – AUTOMOTIVE AI VERTICAL



Phần mềm ô tô đang phát triển nhanh chóng

Phần mềm ô tô đang tăng trưởng bùng nổ. Xe hiện đại chứa hàng triệu dòng mã nguồn, hàng nghìn yêu cầu kỹ thuật, khối lượng tài liệu khổng lồ và quy trình kiểm thử vô cùng phức tạp. Điều này mở ra cơ hội rất lớn để AI tạo ra giá trị và giải quyết các bài toán thực tế trong ngành ô tô.

Hàng triệu dòng code

Một chiếc xe hiện đại có thể chứa hơn **100 triệu dòng mã nguồn**, vượt cả máy bay chiến đấu F-35. Điều này phản ánh xu hướng **Software-Defined Vehicle (SDV)**, nơi phần mềm ngày càng trở thành yếu tố cốt lõi quyết định tính năng và giá trị của xe.

Quá tải thông tin

Các kỹ sư dành **30–40% thời gian chỉ để tìm kiếm thông tin** trong tài liệu, đặc tả kỹ thuật và các tiêu chuẩn ngành. AI có thể giảm đáng kể thời gian này, giúp kỹ sư tập trung vào những công việc kỹ thuật mang lại giá trị cao hơn.

Độ phức tạp của Automotive

Traceability, test coverage và regression analysis là những bài toán cốt lõi của phần mềm ô tô. Một thay đổi nhỏ có thể ảnh hưởng đến hàng trăm thành phần, khiến việc phân tích tác động và kiểm thử hồi quy trở nên phức tạp và tốn nhiều công sức.

PHẦN 1 – AUTOMOTIVE AI VERTICAL



Vì sao AI có tiềm năng trong Automotive Engineering?

AI không chỉ là một công cụ tìm kiếm thông minh hơn. Trong kỹ thuật ô tô, AI có thể đóng vai trò như một trợ lý (AI Agent) chủ động: phân tích yêu cầu, sinh test case, đánh giá độ bao phủ kiểm thử, đồng thời dự đoán tác động và rủi ro của các thay đổi trong mã nguồn.

Trợ lý Tri thức

AI giúp kỹ sư tra cứu tài liệu kỹ thuật hiệu quả hơn, trả lời các câu hỏi chuyên ngành và tìm kiếm thông tin liên quan trong hàng nghìn trang đặc tả, tiêu chuẩn và tài liệu kỹ thuật.

Nâng cao Năng suất

AI có thể tự động phân tích yêu cầu, sinh test case và đánh giá độ bao phủ kiểm thử, giúp kỹ sư giảm thời gian cho các công việc lặp đi lặp lại và tập trung hơn vào đổi mới, thiết kế và giải quyết các bài toán kỹ thuật.

Tự động hóa Thông minh

UI Navigation Agent, Workflow Automation và Engineering Copilot có thể tự động hóa các quy trình kiểm thử và xác thực phức tạp, giúp nâng cao năng suất, giảm đáng kể khối lượng công việc thủ công và tăng hiệu quả của kỹ sư.

PHẦN 1 – AUTOMOTIVE AI VERTICAL



Chúng tôi kỳ vọng gì ở các đội thi?

Mục tiêu không phải là xây dựng một chatbot. Điều chúng tôi mong đợi là những AI Agent, Reasoning System và Engineering Assistant có thể hỗ trợ kỹ sư giải quyết các bài toán thực tế của ngành ô tô thông qua khả năng phân tích, suy luận và ra quyết định.

AI Agents

AI Agent có khả năng lập kế hoạch, thực hiện chuỗi tác vụ nhiều bước và tự chủ thích ứng với từng tình huống để hoàn thành các mục tiêu phức tạp.

Reasoning Systems

Reasoning System không chỉ trả lời câu hỏi mà còn có khả năng phân tích, suy luận và đưa ra kết luận dựa trên dữ liệu, bằng chứng và ngữ cảnh của các bài toán kỹ thuật thực tế.

Engineering Assistants

Giải pháp cần hướng tới việc giải quyết các bài toán thực tế trong kỹ thuật ô tô, chẳng hạn như requirements traceability, test coverage gaps, regression risk analysis và defect prediction.

PHẦN 2 – GIỚI THIỆU 7 CHALLENGES



7 thử thách. Chọn đề phù hợp. Hiện thực hóa ý tưởng.

#1

KMS AI Agent

AI Agent có khả năng đọc và hiểu bộ tài liệu kỹ thuật ô tô (PDF), sau đó trả lời các câu hỏi của người dùng một cách chính xác dựa trên chính tài liệu đó.



#2

Requirement Breakdown

AI Agent phân tích tài liệu yêu cầu của hệ thống ô tô, tự động tách thành các yêu cầu và yêu cầu con có cấu trúc, đồng thời bảo toàn mối liên kết truy vết (traceability) giữa các yêu cầu.



#3

Coverage Analysis Agent

AI Agent phân tích tài liệu yêu cầu và danh sách các yêu cầu, xác định mối tương ứng giữa chúng và tạo báo cáo về độ bao phủ của các yêu cầu.



#4

Req-to-Code Coverage

Xây dựng một AI Agent có khả năng phân tích một yêu cầu và mã nguồn của hệ thống, sau đó xác định yêu cầu đó đã được hiện thực đầy đủ, hiện thực một phần hay chưa được hiện thực.



#5

Regression Impact Analysis

Xây dựng một AI Agent có khả năng phân tích bộ yêu cầu, mã nguồn của hệ thống và một Pull Request (PR), từ đó xác định những yêu cầu nào có thể bị ảnh hưởng bởi các thay đổi trong PR.



#6

PR-to-Bug Prediction

Xây dựng một AI Agent có khả năng phân tích mã nguồn, tài liệu yêu cầu, bug ticket và Pull Request (PR), sau đó **dự đoán khả năng PR có khắc phục được lỗi hay không.**



#7

Vision UI Navigation Agent

Vision-based AI Agent có khả năng quan sát giao diện người dùng (ảnh chụp màn hình hoặc giao diện trực tiếp), hiểu các bước thao tác và tự động điều hướng, thực hiện chúng một cách chính xác.



★ dễ (4–6h) ★★ trung bình (8–12h) ★★★ khó (16h) ★★★★ rất khó (20–24h, dành cho đội mạnh)

PHẦN 2 – GIỚI THIỆU 7 CHALLENGES



Challenge #1 – KMS AI Agent for Automotive Documentation

Bài toán: Các dự án phát triển phần mềm ô tô tạo ra khối lượng rất lớn tài liệu kỹ thuật, bao gồm tài liệu yêu cầu, thiết kế hệ thống, tài liệu mô tả giao diện (Interface Control Document), hướng dẫn tuân thủ, kế hoạch kiểm thử/xác thực và các tiêu chuẩn (ví dụ: an toàn chức năng, an ninh mạng). Kỹ sư thường phải dành nhiều thời gian tìm kiếm thông tin cần thiết trong hàng loạt tài liệu PDF khác nhau.

Mục tiêu

Hỗ trợ semantic search và contextual question answering trên bộ tài liệu lớn.

Trả lời chính xác, dựa trên nội dung của tài liệu.

Trích dẫn nguồn đầy đủ: tên tài liệu, section và số trang.

INPUT & OUTPUT

Đầu vào:

Các tài liệu kỹ thuật ô tô (PDF).

Đầu ra:

Câu trả lời bằng ngôn ngữ tự nhiên, kèm theo tham chiếu có thể truy vết (traceable references) đến tài liệu nguồn.

Giá trị

Độ chính xác và tính liên quan của câu trả lời.

Độ chính xác của nguồn tham chiếu.

Hiệu năng: thời gian phản hồi và khả năng mở rộng.

Khả năng xử lý các truy vấn phức tạp hoặc không rõ ràng.

PHẦN 2 – GIỚI THIỆU 7 CHALLENGES



Challenge #2 – Requirement Breakdown Agent

Bài toán: Tài liệu yêu cầu trong các dự án ô tô thường rất dài và phức tạp, bao gồm các yêu cầu cấp cao (high-level requirements), yêu cầu dẫn xuất (derived requirements), ràng buộc (constraints) và giả định (assumptions). Việc phân rã yêu cầu và quản lý traceability theo phương pháp thủ công không chỉ tốn nhiều thời gian mà còn dễ xảy ra sai sót.

Mục tiêu

Xác định các yêu cầu ở mức thấp nhất (atomic requirement).
Trích xuất cấu trúc phân cấp giữa các yêu cầu (quan hệ cha – con).
Tạo liên kết truy vết (traceability mapping).

Technologies

Đầu vào
Tài liệu yêu cầu trong lĩnh vực ô tô.
Đầu ra
Danh sách các yêu cầu được tổ chức theo cấu trúc phân cấp (hierarchy).
Ma trận truy vết (Traceability Matrix) thể hiện mối liên kết giữa các yêu cầu.

Giá trị

Độ chính xác của việc trích xuất yêu cầu.
Chất lượng của việc xác định cấu trúc phân cấp.
Mức độ đầy đủ và tính nhất quán của liên kết truy vết (traceability).

PHẦN 2 – GIỚI THIỆU 7 CHALLENGES



Challenge #3 – Requirement Coverage Analysis Agent

Bài toán: Trong phát triển phần mềm ô tô, việc đảm bảo mọi tính năng và yêu cầu tuân thủ đều được đề cập đầy đủ trong các tài liệu đặc tả là vô cùng quan trọng. Những khoảng trống về độ bao phủ (coverage gaps) có thể dẫn đến các vấn đề về an toàn, tuân thủ quy định hoặc chức năng của hệ thống.

Mục tiêu

Thực hiện đối sánh ngữ nghĩa (semantic matching) giữa các tập yêu cầu.

Xác định các yêu cầu đã được bao phủ đầy đủ, bao phủ một phần hoặc còn thiếu.

Tự động tạo báo cáo độ bao phủ (coverage report) theo cấu trúc.

Technologies

Đầu vào:

Tài liệu nguồn chứa các yêu cầu.

Danh sách các yêu cầu cần đối chiếu.

Đầu ra:

Bảng ánh xạ giữa các yêu cầu (Requirement Mapping Table).

Báo cáo tổng hợp về độ bao phủ (Coverage Summary Report).

Giá trị

Độ chính xác của việc ánh xạ yêu cầu.
Khả năng phát hiện các yêu cầu chưa được bao phủ.

Tính rõ ràng và khả năng sử dụng của báo cáo.

PHẦN 2 – GIỚI THIỆU 7 CHALLENGES



Challenge #4 – Requirement-to-Code Coverage Analysis

Bài toán: Trong phát triển phần mềm ô tô, việc xác minh rằng các yêu cầu đã được hiện thực chính xác trong mã nguồn là yếu tố then chốt để đảm bảo tuân thủ tiêu chuẩn và an toàn chức năng (ví dụ: ISO 26262).

Mục tiêu

Ánh xạ yêu cầu với các thành phần mã nguồn liên quan.
Xác định trạng thái hiện thực của từng yêu cầu.
Giải thích căn cứ cho kết quả đánh giá.

Technologies

Đầu vào:
Một hoặc nhiều yêu cầu.
Mã nguồn của hệ thống.
Đầu ra:
Kết quả phân loại mức độ hiện thực của từng yêu cầu.
Bảng chứng hỗ trợ cho kết quả đánh giá

Giá trị

Độ chính xác của kết quả phân loại.
Chất lượng của traceability giữa yêu cầu và mã nguồn.
Khả năng giải thích kết quả.

PHẦN 2 – GIỚI THIỆU 7 CHALLENGES



Challenge #5 – Regression Impact Analysis Agent

Bài toán: Khi các thay đổi được đưa vào hệ thống thông qua Pull Request (PR), việc xác định những yêu cầu nào có thể bị ảnh hưởng là rất quan trọng để hạn chế lỗi hồi quy (regression), đặc biệt trong các hệ thống ô tô có yêu cầu cao về an toàn.

Mục tiêu

Phân tích code diff trong Pull Request (PR).
Đối chiếu các thay đổi trong mã nguồn với các yêu cầu liên quan.
Xác định các yêu cầu có thể bị tác động (impacted requirements).

Technologies

Đầu vào
Bộ yêu cầu.
Mã nguồn.
Pull Request (PR).
Đầu ra
Danh sách các yêu cầu có khả năng bị ảnh hưởng.
Mức độ rủi ro của từng yêu cầu.

Giá trị

Độ chính xác của kết quả dự đoán tác động.
Khả năng giảm thiểu kết quả dương tính giả (false positives) và âm tính giả (false negatives).
Mức độ rõ ràng của phần giải thích rủi ro.

PHẦN 2 – GIỚI THIỆU 7 CHALLENGES



Challenge #6 – PR-to-Bug Prediction Agent

Bài toán: Trong các dự án phần mềm ô tô quy mô lớn, việc xác định một Pull Request (PR) có thực sự khắc phục lỗi đã được báo cáo hay không rất phức tạp, do tồn tại nhiều mối phụ thuộc và liên kết truy vết (traceability) giữa yêu cầu và mã nguồn.

Mục tiêu

Phân tích mô tả lỗi (bug) và các yêu cầu liên quan.
Phân tích các thay đổi trong Pull Request (PR).
Dự đoán khả năng PR khắc phục được lỗi, kèm theo điểm tin cậy (confidence score).

Technologies

Đầu vào:
Mã nguồn của hệ thống.
Tài liệu yêu cầu.
Bug ticket.
Pull Request (PR).
Đầu ra:
Xác suất PR có thể khắc phục lỗi (probability score).
Giải thích căn cứ cho kết quả dự đoán (reasoning).

Giá trị

Độ chính xác của dự đoán.
Chất lượng và tính rõ ràng của phần giải thích.
Tính hữu ích đối với reviewer.

PHẦN 2 – GIỚI THIỆU 7 CHALLENGES



Challenge #7 – Vision-Based UI Navigation Agent

Bài toán: Các HMI (Human-Machine Interface) trên ô tô và các công cụ kỹ thuật nội bộ thường yêu cầu nhiều thao tác điều hướng thủ công lặp đi lặp lại trong quá trình kiểm thử và xác thực. Việc tự động hóa các thao tác trên giao diện sẽ giúp nâng cao hiệu quả kiểm thử và giảm công sức thực hiện thủ công.

Mục tiêu

Nhận diện và hiểu các thành phần trên giao diện thông qua hình ảnh.
Ánh xạ các hướng dẫn bằng văn bản thành các thao tác trên giao diện.
Thực hiện các bước điều hướng và thao tác một cách chính xác, ổn định.

Technologies

Đầu vào:
Giao diện người dùng (UI).
Danh sách các bước thao tác.
Đầu ra:
Kết quả thực thi tự động các bước điều hướng và thao tác trên giao diện.
Báo cáo kết quả thành công/thất bại của từng bước hoặc toàn bộ quy trình.

Giá trị

Độ chính xác của việc nhận diện thành phần giao diện.
Tỷ lệ hoàn thành tác vụ.
Khả năng thích ứng với sự thay đổi của giao diện.

PHẦN 3 – STARTER PACK



Starter Kit chung cho tất cả các challenges trong AI Hackathon

✓ FastAPI Starter Template

Project đầy đủ với FastAPI, uvicorn, Pydantic. Dual-purpose: Web API + Solution Pipeline. Bật lên chạy ngay.

✓ Dual-Purpose Architecture

src/main.py (Web API) + src/solve.py (Solution Pipeline). Evaluator hệ thống dùng solve.py để chấm điểm.

✓ Smart Archiver & Submit

Auto zip khi submit: include source, source, data/, artifacts/. Exclude .venv/, tests/. JWT-secured portal upload.

✓ AI/ML Scaffold sẵn sàng

agents/ (LLM), pipelines/ (batch), repositories/ (persistence) — thư mục mục trống, thêm code AI vào đây.

✓ Workshop "AAOS for hackers"

Giả lập toàn bộ luồng: Load dataset → Run solution → Score in-memory. Debug trước khi submit thật.

✓ Tech Stack hiện đại

Python 3.12+, FastAPI, uv (package manager siêu nhanh), Ruff, MyPy, Docker & Docker Compose.

✓ Clean Architecture

Tách biệt api/ (HTTP), services/ (Business Logic), core/ (Config), schemas/ (Pydantic models).

✓ Sample datasets

Dataset mẫu cho từng challenge: tài liệu automotive, requirements, code samples, test cases.

✓ Evaluation metrics

Mỗi challenge có metric rõ ràng: precision, recall, F1 cho NLP; coverage %, false positive rate cho analysis.

PHẦN 3 – STARTER PACK



NLP / LLM TEAMS

Có background NLP, language model, RAG

- #1 KMS AI Agent
- #2 Requirement Breakdown

Chọn #1, #2 hoặc #3. Đây là sân chơi lý tưởng cho LLM & RAG teams.

SOFTWARE ENGINEERING TEAMS

Có background SE, code analysis, dependency

- #4 Req-to-Code Coverage
- #5 Regression Impact Analysis
- #6 PR-to-Bug Prediction

Tận dụng static analysis, dependency graphs, và code LLM. Chọn 1 challenge và đào sâu.

COMPUTER VISION TEAMS

Có background vision, image processing, VLM

- #7 Vision UI Navigation Agent

Challenge duy nhất sử dụng Vision-Language Models. Đây là sân của bạn — đào sâu vào screenshot understanding và agent execution.

FULL-STACK AI TEAMS

Có mix skills: NLP + code + systems

- #7 Vision UI Nav
- #4 Req-to-Code Coverage
- #3 + #4 combo

Combine NLP + code analysis cho maximum impact. #3 + #4 là combo tự nhiên tạo end-to-end solution.

ĐỘI NGŨ HƯỚNG DẪN



Thử thách	Người hướng dẫn
AI #1 KMS AI Agent for Automotive Documentation	LinhND16@fpt.com
AI #2 Requirement Breakdown & Traceability Agent	LyNTT43@fpt.com
AI #3 Requirement Coverage Analysis Agent	LyNTT43@fpt.com
AI #4 Requirement-to-Code Coverage Agent	HaiDN20@fpt.com
AI #5 Regression Impact Analysis Agent	PhongNQ4@fpt.com
AI #6 PR-to-Bug Fix Likelihood Prediction Agent	NhuNDQ2@fpt.com
AI #7 Vision-Based UI Navigation Agent	NhuNDQ2@fpt.com



Automotive
Hackathon 2025

THANK YOU!

FPT Software Company Limited

- FPT Cau Giay Building, Duy Tan Street, Cau Giay Ward, Hanoi City, Vietnam

Tel.: +84 (24) 3 768 9048

Fax: +84 (24) 3 768 9049

Đã có đội? Đăng ký tại đây



Chưa có đội? Cần ghép?



Digital Cockpit



MR. VŨ MINH ĐỨC

Senior Automotive Engineer, FPT Automotive

4

challenges

Android Automotive

core technologies

AAOS

nền tảng chính

1

PHẦN 1 – Tổng Quan

2

PHẦN 2 – 4 Bài Tập

3

PHẦN 3 – Tài Nguyên cuộc thi

4

PHẦN 4 – Gợi ý chọn đề thi & mentor

Phần 1 – Tổng Quan

Cockpit là điểm tương tác trung tâm giữa chiếc xe và người lái.

Buồng lái hiện đại là một hệ thống tích hợp gồm màn hình, cảm biến và kết nối.

Người dùng tương tác với xe gần như hoàn toàn thông qua cockpit — từ cụm đồng hồ, màn hình IVI, điều khiển điều hòa đến trợ lý giọng nói.

Đây là trung tâm của trải nghiệm người dùng trên xe, đồng thời là nơi thể hiện rõ nhất sự khác biệt giữa các thương hiệu ô tô.

Tương Tác Với Người Dùng

Khác với DMS hay AVE chỉ hoạt động ở nền, CDC sở hữu giao diện trực quan.

Khi được triển khai tốt, CDC có khả năng tạo ra những màn demo trên sân khấu đầy ấn tượng và thuyết phục.

Nền Tảng Mở

Android Automotive OS là nền tảng mã nguồn mở, hỗ trợ emulator và sở hữu cộng đồng phát triển rộng lớn.

Nhờ đó, đội ngũ có thể tận dụng hiệu quả các nguồn tài nguyên công khai để tăng tốc phát triển.

Hệ Thống tích hợp

Cockpit là trung tâm hiển thị, nơi hội tụ tín hiệu từ tất cả các vertical — từ DMS, AVE đến AI.

Nhờ đó, CDC trở thành một nền tảng tích hợp tự nhiên, dễ dàng kết nối và mở rộng với nhiều hệ thống khác nhau.

Bài Tập Của Phần DC

Bốn bài toán trong domain Cockpit, bao phủ cả Android Automotive OS và các bài toán mô phỏng — từ infotainment, HVAC full-stack đến voice AI và chẩn đoán xe.

1

Media Player with Album Art & Metadata

MEDIUM · Android Automotive OS

Rich, multi-source in-car music playback with metadata, caching and voice control.



2

Climate Control System with VHAL

HARD · Android Automotive OS

Full-stack HVAC from HMI through Framework and System Service down to Vehicle HAL.



3

Voice-Controlled Assistant

MEDIUM · Simulation-based

Hands-free car control and delivery voice flows with low-latency intent recognition.



4

DTC Monitor & Analyzer

MEDIUM · Simulation-based

Real-time OBD-II/UDS trouble-code simulation, analysis and export — no hardware.



★ dễ (4–6h) ★★ trung bình (8–12h) ★★★ khó (16h) ★★★★ rất khó (20–24h+, dành cho đội mạnh)

PHẦN 2 – GIỚI THIỆU 4 CHALLENGES



Challenge #1 – Media Player with Album Art & Metadata

Bài toán: Trải nghiệm âm nhạc trên xe nâng cao — hỗ trợ đa nguồn, tích hợp điều khiển bằng giọng nói, tối ưu theo ngữ cảnh ngày/đêm.

Tổng quan

- Một media player trên xe, mang lại trải nghiệm phát mượt và nhanh, tối ưu cho môi trường automotive.
- Hỗ trợ đa nguồn — từ bộ nhớ nội bộ, USB đến dịch vụ streaming — đảm bảo tính linh hoạt cho nhiều kịch bản sử dụng.

Chức năng chính

- Điều khiển phát lại đầy đủ — phát, tạm dừng, chuyển bài, tua nhanh, phát ngẫu nhiên và lặp lại
- Hiển thị metadata phong phú bao gồm ảnh album, nghệ sĩ, tiêu đề và tiến trình phát
- Hiển thị ảnh album động với cơ chế caching để tối ưu hiệu năng
- Duyệt thư viện theo nghệ sĩ, album và thể loại; hỗ trợ playlist, mục yêu thích và danh sách phát gần đây

Tiêu chí đánh giá

- Kiến trúc — MediaBrowserService + MediaSession với thiết kế tách lớp rõ ràng
- UX — giao diện tối ưu cho môi trường lái xe, với các vùng chạm lớn, dễ thao tác
- Hiệu năng — caching album art, tua mượt, đảm bảo không ANR khi tải cao
- Khả năng mở rộng — interface plugin được chuẩn hóa, sẵn sàng cho các nguồn media trong tương lai

PHẦN 2 – GIỚI THIỆU 4 CHALLENGES



Challenge #2 – Climate Control System with VHAL

Bài Toán: Full-stack HVAC — App → Framework → System Service → Vehicle HAL

Tổng Quan

Một hệ thống HVAC tích hợp trên Android Automotive OS và VHAL, kết nối ứng dụng với phần cứng xe trên toàn bộ stack — từ Application đến VHAL — cho phép điều khiển và luồng dữ liệu end-to-end.

Chức năng chính

- End-to-end trên toàn bộ 4 tầng: App (HMI), Framework, System Service và VHAL
- Chức năng HVAC cốt lõi: nhiệt độ, tốc độ quạt, chế độ gió và điều hòa ghế
- Đồng bộ UI ↔ xe theo thời gian thực thông qua các property của VHAL

Tiêu Chí Đánh Giá

- Full-stack end-to-end — kết nối đầy đủ 4 tầng, không chỉ là mock UI
- VHAL correctness — đảm bảo property ID, luồng get/set và callback chính xác
- Real-time sync — độ trễ thấp, không có trạng thái lỗi thời
- Testability — có thể kiểm chứng mà không cần phần cứng ECU thực tế

PHẦN 2 – GIỚI THIỆU 4 CHALLENGES



Challenge #3 – Voice-Controlled Assistant

Bài Toán: Tương tác trên xe không cần chạm — điều khiển xe, hỗ trợ các kịch bản dịch vụ, xử lý ngôn ngữ tự nhiên (NLP) với độ trễ thấp

Tổng Quan

Một trợ lý giọng nói cho phép người lái tương tác với các chức năng và dịch vụ trên xe bằng ngôn ngữ tự nhiên, nâng cao độ an toàn và tiện dụng.

Được thiết kế với khả năng mở rộng, hỗ trợ tích hợp với các hệ thống trên xe, dịch vụ cloud và các năng lực AI voice trong tương lai.

Chức năng chính

- Điều khiển xe — khóa/mở cửa, điều chỉnh âm lượng, media và điều hòa bằng giọng nói
- Dịch vụ cơ bản — thời tiết, gọi điện rảnh tay, infotainment
- Kịch bản giao nhận — điều hướng tuyến đường, theo dõi trạng thái đơn hàng và hướng dẫn từng chặng

Tiêu chí đánh giá

- Coverage — bao phủ cả điều khiển xe và kịch bản giao nhận
- Accuracy — nhận diện intent chính xác trong nhiều ngữ cảnh và điều kiện nhiễu
- Latency — phản hồi dưới 1.5s với các lệnh xử lý tại edge
- Extensibility — quản lý intent rõ ràng, không cần tái cấu trúc core

PHẦN 2 – GIỚI THIỆU 4 CHALLENGES



Challenge #4 – DTC Monitor & Analyzer

Bài Toán: Mô phỏng mã lỗi chẩn đoán (DTC) theo thời gian thực — tuân thủ chuẩn OBD-II/UDS, không yêu cầu ECU vật lý

Tổng Quan

Một công cụ chẩn đoán giúp theo dõi và phân tích các mã lỗi chẩn đoán (DTC) trong môi trường mô phỏng, tuân theo chuẩn OBD-II/UDS — theo dõi mã lỗi theo thời gian thực và phân tích tần suất, xu hướng và phân loại, được mô phỏng hoàn toàn và sẵn sàng mở rộng để tích hợp với ECU thực.

Chức năng chính

- Giám sát DTC — tạo, cập nhật và xóa mã lỗi theo thời gian thực
- Hỗ trợ định dạng lỗi theo chuẩn OBD-II / UDS (các nhóm P, C, B, U)
- Luồng dữ liệu trực tiếp (live feed) với trạng thái (active, pending, stored) và mức độ nghiêm trọng
- Phân tích DTC — tần suất, xu hướng lịch sử và phân loại
- Phát hiện lỗi lặp lại và phân tích tương quan giữa các lỗi xảy ra đồng thời
- Phân loại theo hệ thống: động cơ, hộp số, thân xe, khung gầm và mạng

Tiêu chí đánh giá

- Simulation fidelity — mô phỏng lỗi sát thực tế, hỗ trợ kịch bản concurrent và sequential
- Analysis depth — phân tích sâu về tần suất, xu hướng và tương quan lỗi
- Extensibility — thiết kế rõ ràng, dễ dàng chuyển đổi từ simulator sang ECU thực

3. Tài Nguyên Cuộc Thi

Tài nguyên sử dụng trong cuộc thi

✓ Dùng Chung 4 bài toán

AAOS14 – google trout trên nền tảng ảo hóa FPT

✓ Media Player

Virtual USB device kết nối sẵn với AAOS

✓ HVAC-VHAL

Default VHAL trên AAOS

✓ Voice Control

Mic, speaker , Minimal pipeline cho việc xử lý âm thanh --> Text và ngược lại

✓ DTC

Default AAOS, UDS default



PHẦN 4 – GỢI Ý CHỌN TOPIC THEO THỂ MẠNH



Topic	Độ khó	Thiên về	Phù hợp nếu bạn mạnh về
1. Media Player	★★★	Thuần ứng dụng xe hơi	Tìm hiểu chức năng và lập trình ứng dụng
2. Climate control system	★★★★	Android automotive	full-stack, làm xuyên các tầng trong hệ thống kiến trúc AAOS
3. Voice control assistant	★★★	Tương tác hệ thống, có thể áp dụng AI	Nắm rõ các giao thức trong hệ thống aaos
4. DTC Monitor & Analyzer	★★★	Liên kết ECU trong hệ thống, bóc tách dữ liệu	Nếu bạn hiểu giao thức nói chuyện giữa các ECU và khả năng phân tích DTC

Q&A & LIÊN HỆ



Câu hỏi? · Liên hệ mentor · Tài nguyên



Hoàng Mạnh Kiểm
Senior Engineer

FA.CDC



Vũ Minh Đức
Senior Engineer

FA.CDC



Lê Minh Tuệ
Senior Engineer

FA.CDC

Slack channel: [#cdc-vertical](#) (workspace cuộc thi)

Email cockpit lead: kiemhm@fpt.com · trả lời 24h

Office hours: Thứ 3 & 5, 19:00–20:00 (3 tuần Vòng 2)

Repo starter pack: github.com/fpt-automotive-hackathon/cdc-starter

Workshop record: "AAOS for hackers" · public từ 25/06

FAQ chung: fptautomotive-hackathon.com



Automotive
Hackathon 2025

THANK YOU!

FPT Software Company Limited

- FPT Cau Giay Building, Duy Tan Street, Cau Giay Ward, Hanoi City, Vietnam

Tel.: +84 (24) 3 768 9048

Fax: +84 (24) 3 768 9049

Đã có đội? Đăng ký tại đây



Chưa có đội? Cần ghép?



AutoSAR & Vehicle Middleware



MR. NGUYỄN VĂN LINH

AIDV Technical Manager, FPT Automotive

3

đề bài gợi ý



4 mức độ khó

1

GIỚI THIỆU VỀ SOVD & SDV

2

GIỚI THIỆU 4 CHALLENGES

3

STARTER PACK

4

GỢI Ý CHỌN ĐỀ

PHẦN 1 - GIỚI THIỆU VỀ SOVD & SDV



SOVD (Service-Oriented Vehicle Diagnostics) là chuẩn API chẩn đoán hướng dịch vụ, cloud-native (ISO 17978) cho Software-Defined Vehicle, cho phép truy cập và chẩn đoán thống nhất mọi ECU, HPC và ứng dụng trong xe qua REST/JSON

AUTOSAR — OS chuẩn hóa của xe

Chuẩn kiến trúc phần mềm nhúng cho ô tô (AUTomotive Open System ARchitecture), gồm 2 nhánh:

- Classic AUTOSAR: chạy trên MCU real-time (vd TC397) — phanh, cửa, đèn, motor; độ trễ thấp, an toàn (ASIL).
- Adaptive AUTOSAR: chạy trên HPC (Linux), hướng dịch vụ (SOA) — ADAS, cloud. Nền cho SOVD.

OTA — Cập nhật phần mềm từ xa

- FOTA: cập nhật firmware ECU.
- SOTA: cập nhật ứng dụng / dịch vụ cấp cao.
- Cần kiểm tra trước, xác minh sau, rollback an toàn

SOVD — Chẩn đoán hướng dịch vụ

- API chuẩn hóa (ISO 17978), tương thích Adaptive AUTOSAR, cloud-native, dùng REST + JSON.
- Truy cập thống nhất tới ECU, HPC, App, Container, Zone; cùng chạy với các công nghệ UDS / OBD-II cũ (chậm, ECU-centric).

PHẦN 2 – GIỚI THIỆU 3 CHALLENGES



Challenge #1 – SOVD ECU State Provider

MỤC TIÊU Xây dựng mô hình trạng thái vận hành của ECU, truy cập qua API của SOVD: Quản lý **Off / Accessory / Ignition / Running**; hỗ trợ đọc & cập nhật; từ chối các chuyển trạng thái không hợp lệ.

Giải quyết vấn đề gì?

- ECU phải luôn ở trạng thái xác định & an toàn; remote client cần biết và điều khiển trạng thái ECU từ xa.
- Trước khi OTA phải đảm bảo ECU ở trạng thái cho phép — nền tảng cho **remote operation & pre-OTA check**.
- Ngăn lệnh nguy hiểm (Off → Running) bằng kiểm tra chuyển trạng thái hợp lệ.

Keywords & Học được gì

State Machine, FSM, ECU Lifecycle, Vehicle State Manager, SOVD REST API, State Transition Validation, ara::diag

- Thiết kế **máy trạng thái (state machine)** an toàn, rõ ràng.
- Hiểu vòng đời vận hành của ECU.
- Thiết kế **API CRUD** cho trạng thái và xử lý lỗi / validation.

Cơ hội sau khi hoàn thành

- Kỹ năng lõi cho **Linux Embedded / AUTOSAR Adaptive Engineer** và **Vehicle State Manager / SOVD**.
- Module có trong blueprint SOVD của FPT (“Vehicle State Mngr”).
- Tham gia dự án **remote operation & OTA**.

PHẦN 2 – GIỚI THIỆU 3 CHALLENGES



Challenge #2 – Vehicle Data Gateway

MỤC TIÊU Xây dựng một **Vehicle Data Gateway** có khả năng chuyển đổi và giải mã CAN frame thành dữ liệu tín hiệu mức cao, đồng thời cung cấp giao diện truy cập qua **REST API** và **streaming thời gian thực**.

Giải quyết vấn đề gì?

- Dữ liệu CAN ở dạng nhị phân thô, khó dùng; gateway biến tín hiệu thô thành dữ liệu có nghĩa, truy cập qua web.
- Là cầu nối giữa mạng in-car (CAN) và thế giới SOVD / cloud (VSS, data streaming).
- Phục vụ remote client, fleet monitoring, predictive maintenance (UC-1 Query Data / Streaming).

Keywords & Học được gì

CAN/CAN FD, DBC file, Signal Decoding, REST API, Event Streaming, VSS, SOME/IP, Filtering

- Pipeline giải mã CAN: raw frame → physical value (scaling, offset, byte order).
- Thiết kế **REST API** và streaming / event-subscription.
- Tư duy **data engineering** trên dữ liệu xe, đóng gói triển khai độc lập.

Cơ hội sau khi hoàn thành

- Bộ kỹ năng full-stack automotive data — cho **Data Gateway / Connected Vehicle Engineer**.
- Hiểu cả lớp thấp (CAN) lẫn lớp cao (REST / streaming / VSS).
- Nền tảng cho dự án **SOVD Data Server, VSS Feeder, fleet analytics** của FPT Automotive.

PHẦN 2 – GIỚI THIỆU 3 CHALLENGES



Challenge #3 – SOVD Service Model Validator

MỤC TIÊU Kiểm tra tính đúng đắn của **Service Model** theo quy tắc cấu trúc của SOVD: Parse file JSON service model; xác minh trường bắt buộc & quy tắc đặt tên; tạo báo cáo kiểm tra (validation report).

Giải quyết vấn đề gì?

- SOVD chuẩn hóa mô tả dịch vụ chẩn đoán bằng JSON; model sai / thiếu trường có thể hỏng cả chuỗi chẩn đoán & OTA.
- Đảm bảo **chất lượng & tuân thủ chuẩn (compliance / QA)**: phát hiện lỗi cấu hình sớm.
- Đảm bảo interoperability giữa các OEM / supplier — đúng tinh thần chuẩn hóa của SOVD.

Keywords & Học được gì

SOVD Spec (ISO 17978), JSON Schema/Validation, Service Model, Compliance/Conformance, Parsing, Naming Convention, Report Generation

- Đọc & áp dụng đặc tả chuẩn (spec) vào code.
- Kỹ thuật **parse & validate JSON** theo schema / rule.
- Tư duy **conformance testing** và sinh báo cáo tự động.

Cơ hội sau khi hoàn thành

- Hiểu sâu cấu trúc chuẩn SOVD ở mức đặc tả — nền tảng cho **SOVD framework developer, tooling, QA / Conformance Engineer**.
- Topic “nhẹ” hạ tầng nhưng “nặng” hiểu chuẩn.
- Phù hợp bút phá ở vòng lý thuyết (30% SDV / SOVD / OTA).

PHẦN 2 – STARTER PACK



Starter Pack chung dành cho các challenges

✓ A8 Platform

Môi trường khởi tạo và tích hợp hệ thống

✓ Adaptive Autosar Source Code

Source code Adaptive Autosar platform

✓ SOVD Source Code

Source code SOVD

✓ Adaptive Autosar Specifications

Tài liệu đặc tả Adaptive Autosar

✓ SOVD Specifications

Tài liệu đặc tả SOVD

✓ Sample Application

Code ứng dụng mẫu

PHẦN 3 – GỢI Ý CHỌN TOPIC



Topic	Độ khó	Thiên về	Phù hợp nếu bạn mạnh về
1. ECU State Provider	★★	Logic / State machine + API	Embedded, tư duy thiết kế
2. Vehicle Data Gateway	★★★	CAN + REST + Streaming	Data, full-stack, làm sản phẩm hoàn chỉnh
3. Service Model Validator	★★	Đọc spec + Parse / Validate JSON	Chuẩn hóa, QA, chi tiết tỉ mỉ

GỢI Ý NHANH

Cả 3 topic đều ăn điểm phần lý thuyết Final Round (30% SDV / SOVD / OTA). Muốn sản phẩm gây ấn tượng & sát thực tế nhất → **Topic 2**. Muốn cân bằng nỗ lực và độ hoàn chỉnh → **Topic 1**. Muốn hiểu chuẩn để trả lời lý thuyết tốt → **Topic 3**.



Automotive
Hackathon 2025

THANK YOU!

FPT Software Company Limited

- FPT Cau Giay Building, Duy Tan Street, Cau Giay Ward, Hanoi City, Vietnam

Tel.: +84 (24) 3 768 9048

Fax: +84 (24) 3 768 9049

Đã có đội? Đăng ký tại đây



Chưa có đội? Cần ghép?



Connected Car Service



MR. ĐẶNG THÀNH LONG

Innovation Hub Manager FPT Automotive

3

đề bài gợi ý



3 mức độ khó

CCS

nền tảng chính

1

GIỚI THIỆU VỀ VERTICAL

2

GIỚI THIỆU CÁC BÀI TẬP

3

STARTER PACK TOÀN VERTICAL

4

GỢI Ý CHỌN ĐỀ THEO BACKGROUND

PHẦN 1 - GIỚI THIỆU VỀ VERTICAL



DMS Out-Car — Giám sát fleet từ xa qua telemetry & video

SCOPE NOTE

Điểm xuất phát là out-car: Telemetry (speed, brake, steering, lane offset...) được cấp qua simulator; driver-facing/road-facing video vẫn là input hợp lệ. Output cốt lõi phục vụ Fleet Manager / OEM quan sát từ xa — dashboard, event log, scoring report, remote alert.

Remote back-to-car là bonus: Nếu team xây dựng được luồng ngược — gửi lệnh hoặc cảnh báo xuống in-car (IVI alert, speed advisory, driver coaching...) — đây là điểm thưởng cao ở cả 3 đề, không phải yêu cầu bắt buộc.

Out-Car Scope

Telemetry simulator cấp tín hiệu xe; driver-facing video/road-facing video làm input phụ trợ. Đội thi không cần on-vehicle setup — toàn bộ data có sẵn qua nền tảng giả lập.

Fleet Manager Dashboard

Core deliverable là remote dashboard: ranking tài xế, event log, near-miss heatmap, score breakdown, post-trip report. UX phục vụ người không phải engineer.

Remote Back-to-Car (Bonus)

Hướng mở rộng cao điểm: đóng vòng lặp Fleet ↔ Vehicle — gửi alert, coaching notification, hoặc speed advisory ngược vào xe. Thiết kế kiến trúc hai chiều, có safety fallback.

PHẦN 2 – GIỚI THIỆU 3 CHALLENGES



Challenge #1 – FLEET SAFE DRIVING SCORE ENGINE

MỤC TIÊU Xây dựng hệ thống chấm điểm Fleet Safe Driving Score [0–100] per trip, per driver, per vehicle. Score tổng hợp từ driver behavior (attention, drowsiness từ driver-facing video) và vehicle events (harsh braking, lane departure, speeding từ telemetry simulator). Output cốt lõi là fleet-level dashboard: ranking tài xế, phân tích hành vi toàn đội và báo cáo sau chuyến.

Giải quyết vấn đề gì?

Weighting logic phải phản ánh risk thực tế ở cấp fleet: phanh gấp 1 lần có weight khác với buồn ngủ. Score phải stable & auditable — Fleet Manager cần giải thích được tại sao tài xế bị trừ điểm khi có khiếu nại. Cross-trip normalization: highway vs urban phải fair.

INPUT DATA

- Driver-facing video: attention + drowsiness signals (optional)
- Road-facing video: lane marking, lead vehicle
- Telemetry simulator: steering, brake, throttle, speed, lane_offset

OUTPUT YÊU CẦU (OUT-CAR)

- Fleet Dashboard: bảng xếp hạng tài xế, sortable/filterable theo ngày/tuần/tháng — [Fleet Manager]
- Per-trip Score Report: điểm tổng + breakdown từng hành vi, export PDF/JSON — [OEM Analytics]
- Score Timeline Chart: đoạn nào tốt/xấu, map lên GPS trace — [OEM Analytics]
- Top 3 hành vi nguy hiểm nhất của fleet trong kỳ, có timestamp evidence — [Fleet Manager]
- REST API endpoint: trả về score + breakdown theo vehicle_id / trip_id — [Integration]

PHẦN 2 – GIỚI THIỆU 3 CHALLENGES



Challenge #2 – DRIVER INTELLIGENCE PLATFORM

MỤC TIÊU Xây dựng hệ thống Collision Risk Monitor dựa trên camera (vision-only, không radar): ước tính Time-To-Collision (TTC) liên tục từ road-facing video và telemetry simulator, phát hiện và ghi lại các sự kiện rủi ro va chạm cao. Pipeline xử lý data từ xe và stream kết quả lên cloud liên tục — output cốt lõi là remote risk dashboard phục vụ Fleet Manager theo dõi từ xa.

Giải quyết vấn đề gì?

Stereo depth estimation accuracy — phải ước tính TTC đủ tin cậy để không gây false alarm hàng loạt. Phân biệt xe máy tạt đầu (TTC giảm đột ngột) vs xe giảm tốc từ từ để Fleet Manager đọc đúng severity. False positive rate cao = dashboard noise = Fleet Manager mất tin tưởng.

INPUT DATA

- Road-facing video: vehicle, pedestrian, obstacle detection
- Telemetry simulator: ego speed, brake state, steering angle
- Driver-facing video: drowsiness/non-drowsiness

OUTPUT YÊU CẦU (OUT-CAR)

- TTC Stream Log: time-series TTC per frame với timestamp, confidence, object type — CSV/JSON — [OEM]
- Collision Risk Event List: sự kiện TTC vượt ngưỡng, kèm clip evidence & severity — [Fleet Manager]
- Fleet Risk Dashboard: near-miss per vehicle/driver, heatmap tuyến nguy hiểm — [Fleet Manager]
- Per-trip Risk Summary Report: số sự kiện, TTC ngắn nhất, phân bố object type — [OEM]
- Annotated video export: TTC overlay + risk level label cho từng sự kiện — [Evidence]

PHẦN 2 – GIỚI THIỆU 3 CHALLENGES



Challenge #3 – VISION-BASED COLLISION RISK MONITOR

MỤC TIÊU Đe khó nhất: xây dựng Driver Intelligence Platform — nền tảng giám sát fleet toàn diện kết hợp tất cả DMS signals thành một hệ thống duy nhất phục vụ OEM và Fleet Manager. Ba tầng tích hợp: (1) Live Fleet Monitor — trạng thái real-time từng xe, (2) Behavior Analytics — aggregate driver behavior thành insight hành động được, (3) Risk Intelligence — fusion score tổng hợp có explainability.

Giải quyết vấn đề gì?

Fusion nhiều DMS signals không đồng bộ: video latency khác telemetry latency, cần alignment pipeline. Dashboard phải scalable — từ 1 xe demo đến fleet 100 xe về thiết kế. Explainability Unified Risk Score: Fleet Manager không phải engineer, cần plain-language explanation chứ không phải SHAP raw values. False alarm fatigue: log nhiều noise → bỏ qua → hệ thống mất giá trị.

INPUT DATA

- Driver-facing video stream (live replay hoặc offline)
- Road-facing video + full telemetry (simulator)
- Telemetry simulator: ego speed, brake state, steering angle

OUTPUT YÊU CẦU (OUT-CAR)

- Live Fleet Map Dashboard: vị trí + trạng thái risk real-time, color-coded theo risk — [Fleet Manager]
- Driver Behavior Analytics Panel: trend attention/drowsiness/distraction theo tài xế/tuyến/thời gian — [OEM]
- Unified Risk Score per trip: driver state + road risk + vehicle events, có SHAP breakdown — [OEM]
- Fleet-level Alert Log: sự kiện nguy hiểm toàn đội, filterable, exportable — [Fleet Manager]
- Post-trip Coaching Report tự động: 3 điểm cần cải thiện per driver, gửi email/API — [Fleet Manager]
- Integration architecture diagram: DMS signals → pipeline → dashboard — [Documentation]

REMOTE BACK-TO-CAR BONUS XUYÊN SUỐT 3 ĐỀ



Đóng vòng lặp Fleet ↔ Vehicle: hướng mở rộng cao điểm hackathon

Cả 3 đề đều có chung một hướng mở rộng: từ out-car platform gửi ngược intervention hoặc feedback xuống in-car. Đây không phải yêu cầu bắt buộc — nhưng được tính điểm thưởng cao nhất, vì thể hiện đội thi nắm được luồng dữ liệu hai chiều, latency budget, và safety design của một sản phẩm DMS thực tế.

ĐỀ 01

TRUNG BÌNH

Safe Driving Score

Post-trip

Platform gửi coaching notification / speed advisory xuống xe sau chuyến qua IVI hoặc driver app

Real-time

Score & alert hiển thị trực tiếp cho tài xế trong xe ngay trong lúc đang lái

ĐỀ 02

NÂNG CAO

Collision Risk Monitor

Real-time warning

Khi TTC vượt ngưỡng nguy hiểm, gửi cảnh báo ngược vào xe (IVI / haptic / voice) với latency < 200ms

Safety fallback

Pipeline có fallback khi mất kết nối — không được phép trigger false alarm trong xe.

ĐỀ 03

CHUYÊN SÂU

Driver Intelligence Platform

Bonus cao nhất

Platform đóng vòng lặp hai chiều Fleet ↔ Vehicle: phát hiện risk → gửi intervention (speed advisory, IVI coaching)

End-to-end demo

Thiết kế kiến trúc và demo full luồng — phần được tính điểm thưởng lớn nhất toàn hackathon.

Lưu ý: Team chọn hướng phù hợp năng lực; không bắt buộc làm cả 2 chiều. Latency, safety fallback, và explainability là 3 tiêu chí Mentor sẽ chấm bonus này.

STARTER PARK TOÀN VERTICAL



Cả 3 đề đều sử dụng chung 1 dataset và starter kit, mỗi đội có thể sẽ sử dụng dataset này khác nhau tùy thuộc vào đề bài mình chọn. Ngoài ra, các đội cũng sẽ được cung cấp startkit để làm quen và dựa theo đó để phát triển tiếp.

Starter Kit

Bộ starter kit sẽ được cung cấp cho thí sinh

1/. Dataset Loader: Load trip data - ảnh road/driver, depth, ground truth. API

2/. Baseline TTC Predictor: TTC baseline ~50/100 điểm bằng stereo SGBM + ROI median (không AI). Teams phải vượt qua

3/. Explorer Trip Noteboook: Visualize driver state timeline, TTC, risk, sample frames

4/. README: Quick start, API reference, submission format, ideas to beat baseline

Data dùng chung cho toàn vertical

Mỗi trip có 1 bộ data duy nhất, các đề thi khai thác different aspects

```
data/T01/
├─ kitti/image_2/ ← Đề 6, 8: ảnh road camera (left RGB)
├─ kitti/image_3/ ← Đề 6, 8: ảnh road camera (right RGB - stereo)
├─ kitti/depth/   ← Đề 6: depth ground truth (keyframes)
├─ kitti/calib/   ← Đề 6: camera calibration
├─ driver/        ← Đề 8: ảnh in-cabin (NTHU composited)
└─ T01.json.gz    ← Đề 6, 8, 10: tất cả ground truth labels
```

Tại sao 1 dataset phục vụ 3 đề

Design có chủ ý

- Đề 6 dùng road camera (stereo) → output TTC stream
- Đề 8 dùng driver camera → output driver state stream
- Đề 10 fusion cả 2 stream → composite score per trip

Teams có thể tham gia 1 đề riêng hoặc end-to-end cả 3 (đội mạnh sẽ làm Đề 10 vì đòi hỏi giải xong Đề 6 và 8 trước).

GỢI Ý CHỌN ĐỀ THEO BACKGROUND



MỚI BẮT ĐẦU

Chưa có Automotive background

- DMS-06: bắt đầu với starterkit để hiểu rõ các thông tin telemetric, tính scoring dựa trên driver behavior. Có thể cân nhắc hướng tích hợp ngược với driver app.

CÓ AUTOMOTIVE NỀN

Có background về Automotive

- DMS-08: phát triển CV pipeline để tối ưu TTC, tích hợp ngược với in-car (IVI)

ĐỘI MẠNH

Có nhiều kinh nghiệm Automotive / CV

- DMS-10: tích hợp các phần của DMS-06/08, ngoài ra thêm các phần live fleet management, coaching (IVI integration – optional) ...

MUỐN CROSS-VERTICAL

Muốn ăn bonus đa vertical

- DMS-06/08/10: cân nhắc tích hợp thêm AI (i.e AI Assistant, RAG...), in-car intervention (Cockpit), UBI...

Q&A & LIÊN HỆ



Câu hỏi? · Liên hệ mentor · Tài nguyên



Đặng Thanh Long
Innovation Hub Manager

Tổng quan và mentor vertical · DMS-06, 08, 10



Trần Minh Trung
Solution Architect

Mentor: DMS-06, 08, 10

Slack channel: #ccs-vertical (workspace cuộc thi)

Office hours: Thứ 2 - 6, 10AM – 3PM (3 tuần Vòng 2)

Workshop record: TBD

Email vertical lead: longdt50@fpt.com · trả lời 24h

Repo starter pack: github.com/fpt-automotive-hackathon/ccs-starter

FAQ chung: fptautomotive-hackathon.com



Automotive
Hackathon 2025

THANK YOU!

FPT Software Company Limited

- FPT Cau Giay Building, Duy Tan Street, Cau Giay Ward, Hanoi City, Vietnam

Tel.: +84 (24) 3 768 9048

Fax: +84 (24) 3 768 9049

Đã có đội? Đăng ký tại đây



Chưa có đội? Cần ghép?

