FPT Education
FPT UNIVERSITY

Session 2

Threat Modeling - Four Questions, Asset, Attack Surface, Trust Boundary

Presenter: **Mr. Ngo Tung Son (Ph.D.)**

FPT Education
FPT UNIVERSITY

Abbreviations & Acronyms

Abbreviation	Meaning	Explanation / Vietnamese Note
TM	Threat Modeling	Quy trình nhận diện, phân loại, và quản lý mối đe dọa trong thiết kế hệ thống.
Q1 – Q4	Four Questions of Threat Modeling	Bốn câu hỏi: 1 What are we building ? 2 What can go wrong ? 3 What will we do ? 4 Did we do a good job ?.
DFD	Data Flow Diagram	Sơ đồ luồng dữ liệu, hiển thị entry points, trust boundaries và flows.
Asset	High-value data or resource	Tài sản cần bảo vệ (trị giá cao về kinh doanh hoặc bảo mật).
Attack Surface / EP	Entry Points / Attack Surface	Điểm vào ứng dụng (UI, API, integration) và bề mặt tiếp xúc của tấn công.
STRIDE	Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege	Mô hình phân loại đe dọa (6 nhóm) gắn với CIA/AAA.
L×I	Likelihood × Impact	Công thức đánh giá rủi ro (xác suất × tác động).
Mitigation	Control / Countermeasure	Biện pháp giảm thiểu rủi ro (thiết kế, kiểm thử, quy trình).
Boundary / TB-ID	Trust Boundary / Trust Boundary ID	Ranh giới tin cậy và định danh các điểm giao tiếp giữa vùng tin cậy.
AuthN / AuthZ	Authentication / Authorization	Xác thực và phân quyền — phải kiểm tra theo đối tượng (tại thời điểm dùng).
SDR	Security Design Review	Cổng kiểm soát đầu vào kiến trúc an toàn trước xây dựng.
Bug Bar	Bug Severity Bar / Ship Criterion	Ngưỡng phát hành dựa trên mức nghiêm trọng của lỗi (C/H = 0 trước FSR).
FSR	Final Security Review	Cổng cuối đánh giá bằng chứng và rủi ro còn lại trước Go-Live.
AC-SEC-*	Acceptance Criteria – Security	Tiêu chí chấp nhận liên quan đến bảo mật trong user story/test.
Evidence Store	Security Evidence Repository	Kho lưu bằng chứng cho các gates (SDR, Bug Bar, FSR).
Non-repudiation	Accountability / Cannot Deny Action	Tính truy cứu trách nhiệm, chống phủ nhận hành động.
PII / PHI	Personally Identifiable / Protected Health Information	Dữ liệu nhạy cảm cần ưu tiên trong threat modeling.
Misuse / Abuse Case	Reverse Scenario / Attack Story	Tình huống lạm dụng để phát hiện và đánh giá biện pháp bảo vệ.
Residual Risk	Risk After Mitigation	Rủi ro còn lại sau khi áp dụng biện pháp giảm thiểu.

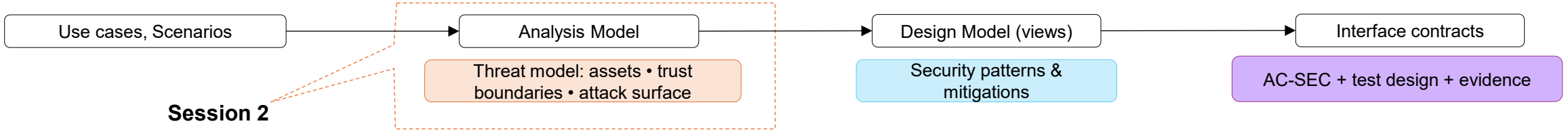
Recap: Security-by-Design, CIA↔AAA, Trust boundary.

SUMMARY

Secure-SDLC = overlay layer on SDLC to establish quality gates (SDR, Bug Bar, FSR) and collect evidence throughout the lifecycle.

- Trust → CIA (Confidentiality-Integrity-Availability) is the goal; Gold Standard (AAA) — Authentication, Authorization, Auditing — is the enforcement mechanism.
  Trust → CIA (Confidentiality-Integrity-Availability) là mục tiêu; Gold Standard (AAA) — Authentication, Authorization, Auditing — là cơ chế thực thi.
- Trust boundaries & attack surface: all data crossing Trust boundaries must be validated + authN/authZ; auditing to ensure accountability and Non-repudiation
  mọi dữ liệu băng qua ranh giới tin cậy phải được validate + authN/authZ; auditing để đảm bảo trách nhiệm giải trình / không chối bỏ.
- Secure-SDLC Overlay across 5 activities; produce evidence in an Evidence Store.
  Lớp bảo mật phủ lên 5 hoạt động; tạo bằng chứng đưa vào kho Evidence..
- Quality Gates: Gates: SDR Bug bar FSR pass/fail by measurable criteria.
  Cổng chất lượng: SDR, Bug Bar, FSR — tiêu chí đo đếm rõ ràng.
- Traceability: REQ → design → code → tests → release → ops.
  Truy vết: Yêu cầu → thiết kế → mã → kiểm thử → phát hành → vận hành.

MODELING ARTEFACTS WITH SECURITY OVERLAY



FPT Education
FPT UNIVERSITY

Objectives

- Understand the Four Questions of Threat Modeling: What are we building? What can go wrong? What will we do? Did we do a good job?
  Bốn Câu Hỏi của Threat Modeling: Chúng ta đang xây dựng gì? Điều gì có thể sai? Chúng ta sẽ làm gì? Chúng ta đã làm tốt chưa?
- Identify core Threat Modeling components: Assets, Attack Surface, and Trust Boundaries.
  Nhận diện các thành phần cốt lõi của Threat Modeling: Tài sản, bề mặt tấn công, và ranh giới tin cậy.
- Practice building a system model using use-cases, context diagrams, and preliminary Data Flow Diagram (DFD-0/DFD-1).
  Thực hành xây dựng mô hình hệ thống sử dụng use-case, sơ đồ bối cảnh và Sơ đồ Luồng Dữ liệu sơ bộ (DFD-0/DFD-1).
- Apply the STRIDE threat taxonomy to systematically enumerate threats.
  Vận dụng danh mục mối đe dọa STRIDE để liệt kê và hệ thống hoá các mối đe doạ.
- Complete a draft Threat Model (v1.0) for a given case study.
  Hoàn thành bản nháp Threat Model v1.0 cho case study (bao gồm DFD và bảng rủi ro).
- Write misuse/abuse scenarios and plan high-level mitigations.
  Viết kịch bản lạm dụng/tấn công và lập kế hoạch giảm thiểu cấp cao.

Information Privacy

WHY PRIVACY IS DISTINCT

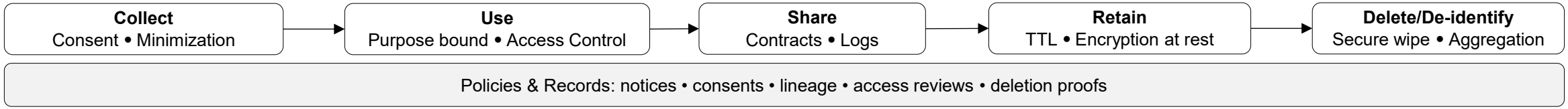
Privacy Yet Tied to Security

- **Beyond Confidentiality.** Privacy extends C-I-A by adding human factors: expectations, policies, law/regulation, culture/psychology.
  Hơn cả Tính bí mật. Quyền riêng tư mở rộng C-I-A bằng các yếu tố con người: kỳ vọng, chính sách, luật/quy định, văn hoá/tâm lý.
- **Societal context.** Software pervades intimate life → more complex issues; norms are evolving.
  Bối cảnh xã hội. Phần mềm đi sâu vào đời sống → vấn đề phức tạp; chuẩn mực đang tiến hoá.
- **Precondition.** Strong privacy is only possible when security is solid.
  Điều kiện tiên quyết. Quyền riêng tư mạnh chỉ khả thi khi an ninh vững chắc.
- **Principles to Respect:**
  - ✓ **Transparency.** Clear, comprehensible policies; align user expectations with actual handling.
    Minh bạch. Chính sách rõ ràng, dễ hiểu; đồng bộ kỳ vọng người dùng với thực tế xử lý dữ liệu.
  - ✓ **Purpose limitation & minimization.** Collect for a specific purpose only; collect the minimum necessary.
    Giới hạn mục đích & tối thiểu hoá. Thu thập cho mục đích cụ thể; chỉ thu thập tối thiểu cần thiết.
  - ✓ **Retention & deletion.** Retain only while useful; securely delete when risk of disclosure > value of keeping.
    Lưu trữ & xoá. Chỉ giữ khi còn hữu ích; xoá an toàn khi rủi ro rò rỉ > giá trị lưu giữ.

FROM POLICY → SOFTWARE ENFORCEMENT

Policy / Expectation	Software Checks & Balances
Explain data uses clearly to customers Giải thích rõ cách dùng dữ liệu	Consent UI; layered notices; change logs; privacy dashboard Giao diện đồng ý; thông báo theo lớp; nhật ký thay đổi; bảng điều khiển quyền riêng tư
Collect minimal data for specific purpose Thu thập tối thiểu cho mục đích cụ thể	Required-fields audit; schema minimization; telemetry allow-list Rà soát trường bắt buộc; tối giản schema; danh sách cho phép telemetry
Limit use/sharing; maintain records Giới hạn sử dụng/chia sẻ; lưu hồ sơ	Access controls; data lineage; sharing registries; immutable logs Kiểm soát truy cập; truy vết dữ liệu; sổ đăng ký chia sẻ; nhật ký bất biến
Delete when no longer needed Xoá khi không còn cần	Retention policies; TTL jobs; secure wipe; deletion proofs Chính sách lưu giữ; tác vụ TTL; xoá an toàn; bằng chứng xoá
Audit extraordinary access Kiểm toán truy cập ngoại lệ	Break-glass workflow; just-in-time approval; audit trail reviews Quy trình phá kính; phê duyệt tức thời; rà soát nhật ký

PRIVACY BY DESIGN ACROSS THE DATA LIFECYCLE



Threats

CORE MESSAGE

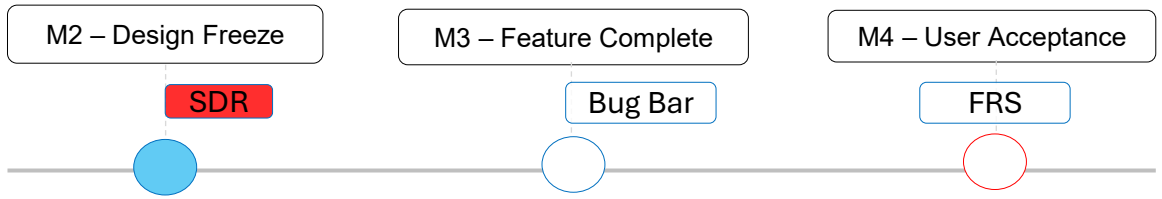
Flip to the attacker's view, enumerate potential threats, then bake defenses into the design

- **Threats are omnipresent;** you can live with them by managing them.
  Mối đe doạ hiện hữu khắp nơi; có thể "sống chung" nếu được quản trị đúng cách.
- **Adopt a *security mindset*:** flip from the builder's perspective to the attacker's.
  Chuyển tư duy sang bảo mật: "đảo" góc nhìn từ người xây dựng sang kẻ tấn công.
- **Stop thinking only about intended use.** See software as code & components with data flowing and being stored.
  Đừng chỉ nghĩ theo cách dùng dự kiến; hãy nhìn hệ thống như mã & thành phần với luồng dữ liệu vào/ra và lưu trữ.
- **Threats include more than adversaries:** bugs, human error, accidents, hardware failures, and more.
  Đe doạ không chỉ là tấn công ác ý: còn có lỗi phần mềm, lỗi con người, sự cố, hỏng hóc phần cứng, v.v.
- **Threat modeling guides decisions across the SDLC,** focusing on concepts & principles, not one rigid method.
  Threat modeling định hướng quyết định xuyên suốt SDLC, tập trung vào khái niệm & nguyên tắc hơn là một phương pháp cố định.
- **Tailor the process.** Early Microsoft practice was heavy-weight; once you grasp the concepts, you can scale effort to fit.
  Tuỳ biến quy trình: thực hành ban đầu khá nặng; nắm vững khái niệm rồi thì có thể co/giãn công sức phù hợp.

- **Raise the bar incrementally, not perfection.** Even partial findings & mitigations can avert major incidents.
  Nâng chuẩn theo bước, không cầu toàn: phát hiện/giảm thiểu một phần vẫn có thể ngăn sự cố lớn.
- **Feedback is sparse.** Foiled attacks go unseen; keep flexing the mindset to see threats better over time.
  Phản hồi thường ít thấy: các cuộc tấn công bị chặn không để lại dấu hiệu; rèn luyện tư duy để "nhìn thấy" tốt hơn.
- **Bonus insight:** threat modeling often reveals improvements unrelated to security (efficiency, simplicity, new features).
  Lợi ích cộng thêm: threat modeling thường gợi ý cải tiến ngoài phạm vi an ninh (hiệu năng, đơn giản hoá, tính năng mới).

ILLUSTRATIVE EXAMPLE

- **Repurposing intent.** A paperclip holds papers by design; bent just right, it becomes a stiff wire to pick locks.
  Tái dụng ngoài mục đích: kẹp giấy được thiết kế để kẹp giấy; uốn đúng cách, nó thành dây cứng để mở khoá.
- **Security mindset heuristic:** ask "what if this component/API is used *off-label*?"
  Kinh nghiệm tư duy: luôn tự hỏi "nếu thành phần/API này bị dùng lệch mục đích thì sao?".
- From usecase to misuse/abuse



FPT Education
FPT UNIVERSITY

The Adversarial Perspective

Exploits are the closest thing to "magic spells" we experience in the real world: Construct the right incantation, gain remote control over device. —Halvar Flake

Khai thác là thứ gần nhất với "phép thuật" mà chúng ta trải nghiệm trong thế giới thực: Tạo ra câu thần chú đúng, giành quyền điều khiển thiết bị từ xa.

1. CORE IDEAS

Adversarial Mindset

- **Humans are the ultimate threat.** Consider capable adversaries without trying to predict every move.
  Con người là mối đe doạ tối hậu; cân nhắc đối thủ giả định nhưng không cố đoán mọi nước đi.
- **Adversary spectrum:** from script-kiddies → organized crime → nation-states.
  Trải phổ đối thủ: từ script-kiddies → nhóm tội phạm có tổ chức → nation-states.
- **Small weaknesses attract attention.** Verbose errors and stack dumps are tempting clues.
  Điểm yếu nhỏ thu hút chú ý; lỗi chi tiết/stack-dump là mồi ngon.
- **Vulnerability chaining.** Several small flaws can compose into a major attack — treat every vuln seriously.
  Chuỗi lỗ hổng: vài khuyết điểm nhỏ kết hợp thành tấn công lớn - coi trọng mọi lỗ hổng.
- **Motives follow value (go where the money is).** High-value targets invite persistent, well-resourced attackers.
  Động cơ theo giá trị: mục tiêu giá trị cao ⇒ đối thủ kiên trì & giàu tài nguyên.
- **Offense is easier than defense.** Attackers need one win; defenders must prioritize wisely.
  Dễ tấn công hơn phòng thủ: kẻ tấn công chỉ cần thắng một lần; người phòng thủ phải ưu tiên khôn ngoan.

2. ADVERSRIAL LENS

Concrete Examples

- **Big targets vs. trivial assets.** Adversaries gravitate to institutions with high payoffs; personal low-value content is rarely worth effort.
  Mục tiêu lớn vs. tài sản tầm thường: kẻ tấn công ưu tiên nơi có lợi ích cao; nội dung cá nhân ít giá trị hiếm khi đáng công.
- **Where the money is.** Motivation predicts diligence: higher value ⇒ more skilled, better-resourced attackers.
  Động cơ theo giá trị: giá trị càng cao ⇒ đối thủ càng kiên trì & có nguồn lực.
- **Chaining small flaws.** (1) Verbose 500 reveals a table name → (2) predictable ID in /invoice/123 → (3) export endpoint lacks auth ⇒ bulk data exfil.
  Xâu chuỗi lỗ hổng nhỏ: lỗi chi tiết → ID đoán được → API xuất dữ liệu thiếu kiểm soát ⇒ rò rỉ lớn.
- **Information leakage as compass.** Stack traces, detailed error codes, public bucket listings — each clue narrows the attack path.
  Rò rỉ thông tin như la bàn: mỗi manh mối giúp kẻ tấn công rút ngắn đường đi.
- **Asymmetric cost.** Offense picks the entry point and tries unlimited times; they only need to succeed once.
  Bất đối xứng chi phí: kẻ tấn công chọn điểm vào và thử không giới hạn, họ chỉ cần thành công một lần.

Part 1
Four Questions and Modeling Components

Threat Modeling — Why & When

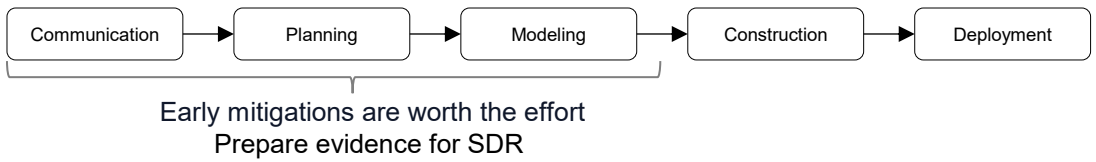
1. WHY

Reduce risk early; cheaper than late fixes

- **Guide security decisions across SDLC:** Use threat modeling as a lens to inform architecture, design, testing, and operations decisions across the lifecycle.
  Định hướng mọi quyết định ảnh hưởng đến an ninh xuyên suốt SDLC; ưu tiên khái niệm & nguyên tắc hơn là công thức cứng.
- **Preemptive, incremental risk reduction:** Reduce risk early in small steps—raise the bar instead of chasing perfection; early wins matter even if feedback is invisible.
  Giảm rủi ro sớm theo từng bước (raise the bar), không cầu toàn; nỗ lực sớm thường "đáng công" dù phản hồi khó thấy.
- **Earlier is cheaper than late fixes:** Finding issues in modeling/design is far cheaper than fixing them late in the lifecycle or after release.
  Phát hiện ở giai đoạn mô hình/thiết kế thường rẻ hơn rất nhiều so với sửa ở cuối vòng đời hay sau phát hành.

Bake defenses into the design

- Embed controls at trust boundaries and high-value surfaces so security is built-in, not bolted-on.
  Nhúng cơ chế phòng vệ ngay trong kiến trúc (giới hạn bề mặt tấn công, kiểm soát tại ranh giới tin cậy) thay vì "vá" về sau.
- **Expected outcomes by end of Modeling:** Outcomes expected after the Modeling phase: evidence-ready security decisions.
  Kết quả mong đợi sau Modeling: giúp quyết định kiến trúc/bảo mật có cơ sở.



2. WHEN

Do it in Modeling; update whenever scope/flows change (assets • boundaries • surfaces)

- **During Modeling (early design):** Perform threat modeling early to surface assets, trust boundaries, and attack surfaces that shape a secure architecture.
  Định hướng mọi quyết định ảnh hưởng đến an ninh xuyên suốt SDLC; ưu tiên khái niệm & nguyên tắc hơn là công thức cứng.
- **Iterate whenever scope or flows change:** Update the model whenever scope, data flows, or design choices change; expand iteratively and focus on high-value assets first.
  Lặp lại/cập nhật mỗi khi phạm vi, luồng dữ liệu, hoặc quyết định thiết kế thay đổi; mở rộng theo từng đợt, tập trung tài sản giá trị cao trước.
- **Typical triggers for a TM update:** Situations that should trigger a threat model refresh:
  Những tình huống nên cập nhật threat model ngay.
  - ✓ New feature or third-party/partner integration.
    Tính năng mới hoặc tích hợp dịch vụ bên ngoài/đối tác.
  - ✓ Changes to data model/flows or storage location of PII/sensitive data.
    Thay đổi mô hình/luồng dữ liệu hoặc nơi lưu trữ PII/dữ liệu nhạy cảm.
  - ✓ Adjustments to AuthN/AuthZ, session/token, or key/secrets policy.
    Điều chỉnh cơ chế xác thực/ủy quyền, session/token, hoặc chính sách khoá/bí mật.
  - ✓ Near-miss/incident, newly discovered vulnerabilities, or environment changes.
    Sự cố suýt xảy ra/đã xảy ra, lỗ hổng mới phát hiện, hoặc thay đổi môi trường vận hành.
  - ✓ New compliance/regulatory requirements
    Yêu cầu tuân thủ/quy định mới.

Four Questions - Overview

QUESTION & INTENT

1) What are we building?

Define scope, assets, actors, and dataflows; draw context/DFD and mark trust boundaries & entry points.

Xác định phạm vi, tài sản, tác nhân và luồng dữ liệu; vẽ context/DFD, đánh dấu ranh giới tin cậy & điểm vào.

artefacts: Context/DFD, Asset list, Trust boundary, Attack surfaces.

2) What can go wrong?

Enumerate threats across boundaries and surfaces; use STRIDE as a taxonomy to spark ideation.

Liệt kê đe doạ tại ranh giới và bề mặt; dùng STRIDE như hệ phân loại để gợi ý.

artefacts: Threat List (incl bugs / human error / accidents / hw failures).

3) What will we do?

Prioritize by Likelihood × Impact; select mitigations (design controls, testing, procedures) and owners/next steps.

Ưu tiên theo Khả năng × Tác động; chọn giảm thiểu (kiểm soát thiết kế, kiểm thử, quy trình) và phân công trách nhiệm.

artefacts: Risk table, Mitigation plan, Test ideas.

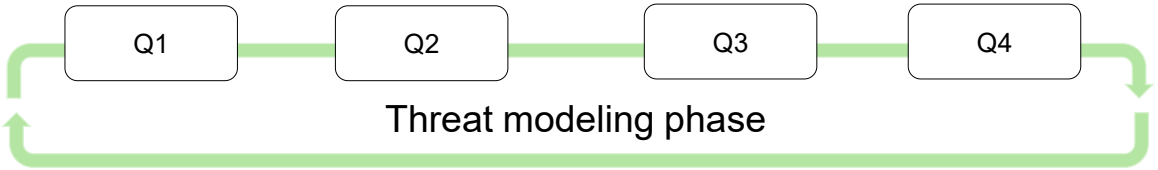
4) Did we do a good job?

Review evidence & residual risk; check coverage and update model as scope changes; prepare for gates.

Rà soát bằng chứng & rủi ro còn lại; kiểm tra độ bao phủ và cập nhật khi phạm vi đổi; chuẩn bị cho các cổng kiểm soát.

artefacts: SDR readiness, Bug Bar status, FSR inputs.

Treat the Four Questions not as a linear sequence but as recurring prompts you revisit throughout threat modeling and the SDLC—especially whenever scope or design changes.



Four Questions – Q1: What are we building?

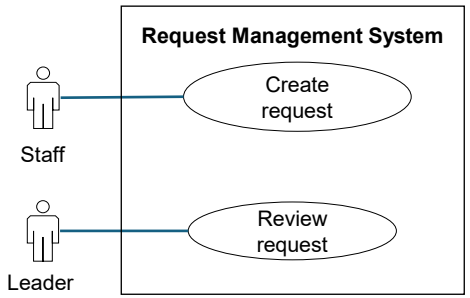
SCOPE AND SYSTEM VIEW

Identify scope, architecture, components, and dataflows.

- **Define scope & value:** Capture business goals/non-goals; in-scope vs out-of-scope; high-value assets & actors.
  Ghi nhận mục tiêu/phi mục tiêu; phạm vi bao hàm/loại trừ; tài sản giá trị cao & tác nhân.
  - ✓ Assets & sensitivity PHI/PII/Fin - owners, locations, retention
    Tài sản & độ nhạy (PHI/PII/tài chính) — chủ sở hữu, nơi lưu trữ, thời hạn giữ.
  - ✓ Primary actors/roles; assumptions & constraints
    Tác nhân/chức năng chính; giả định & ràng buộc.
- **Architecture & flows:** Lay out components, dependencies, data stores, and flows across trust boundaries.
  Bố trí thành phần, phụ thuộc, kho dữ liệu và luồng qua các ranh giới tin cậy.
  - ✓ Context diagram → preliminary DFD (levels 0–1)
  - ✓ Entry points & attack surface (APIs, UIs, events)
  - ✓ Trust boundaries & data classification on each flow

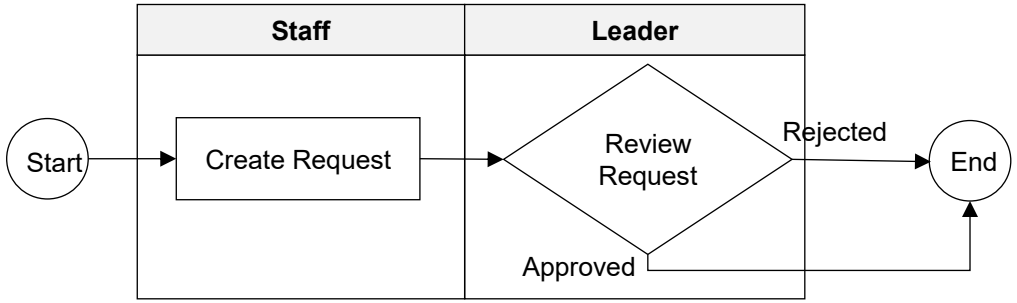
ARTEFACT – USE CASE

- **Use-case diagrams** describe the high-level functions and scope of a system.
  Sơ đồ usecase mô tả tính năng tổng quát và phạm vi hệ thống
- **Use-case** consists diagram and narrative
  Bao gồm : sơ dồ và mô tả



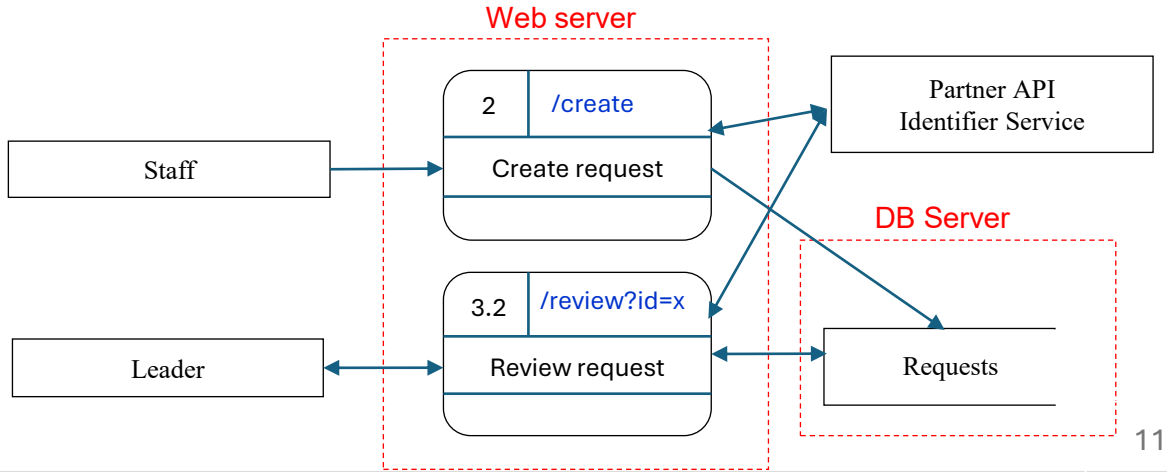
ARTEFACT – ACTIVITY DIAGRAM

Activity diagram provides a view of the behavior of a system by describing the sequence of actions in a process.
cung cấp cái nhìn về hành vi của một hệ thống bằng cách mô tả trình tự các hành động trong một quy trình.



ARTEFACT – DATA FLOW DIAGRAM

Preliminary DFD-0/DFD-1 with trust boundaries & entry points
DFD-0/DFD-1 sơ bộ có ranh giới tin cậy & điểm vào.



Four Questions – Q2: What can go wrong?

SECURITY HAT – START CONCRETE FIRST

Think at entry points and boundary crossings; enumerate specific scenarios before generalizing with STRIDE.

- **Concrete scenarios at entry points:** Begin with real flows and assets at risk; map each to a STRIDE category.
  Bắt đầu từ luồng thực tế và tài sản rủi ro; ánh xạ mỗi kịch bản vào STRIDE.
- **Verbose error reveals schema** at `/auth/login` → attacker pivots to credential stuffing.
  Lỗi chi tiết làm lộ cấu trúc DB tại */auth/login* → kẻ tấn công chuyển sang nhồi thông tin đăng nhập.
- **Token replay across boundary**: stolen session token is reused from a different network.
  Tái sử dụng token qua ranh giới: token bị lộ được dùng lại từ mạng khác.
- **Insecure export endpoint** `/export?file=student.csv` without auth/rate-limit → bulk data exfiltration.
  Endpoint xuất dữ liệu thiếu xác thực/giới hạn tốc → rò rỉ dữ liệu hàng loạt.
- **Partner webhook abuse**: forged requests from outside cross the boundary into internal service.
  Lạm dụng webhook đối tác: yêu cầu giả mạo từ ngoài vượt ranh giới vào dịch vụ nội bộ.
- **Insufficient logging** prevents accountability on money-movement actions.
  Ghi log không đủ làm mất khả năng quy trách nhiệm trong thao tác chuyển tiền.

Four Questions – Q3: What will we do? (Mitigation)

MITIGATION 3 LEVERS

WHAT WE CHANGE

- Less likely : Harden inputs & policies; shrink attack surface; strong AuthN/Z.
  Giảm khả năng: kiểm tra đầu vào/policy; thu nhỏ bề mặt tấn công; xác thực/ủy quyền mạnh.
- Less harmful: Smaller blast radius Least privilege; minimize & delete data; rate limits/quotas.
  Giảm tác hại: đặc quyền tối thiểu; tối thiểu hóa & xóa dữ liệu; giới hạn tần suất/hạn mức.
- More detectable :Faster detection & recovery Tamper-evident audit logs; monitoring & alerts; backups & restore drills.
  Phát hiện/phục hồi: log chống sửa; giám sát & cảnh báo; sao lưu & diễn tập khôi phục.

DEFENSE IN DEPTH – INDEPENDENT LAYERS

Multiple layers of independent protection to reduce error duplication

- Policy & Validation (contracts, input rules)
- AuthN/AuthZ (deny-by-default, scopes)
- Sandbox/Isolation (process/container)
- Audit/Monitoring (tamper-evident logs)

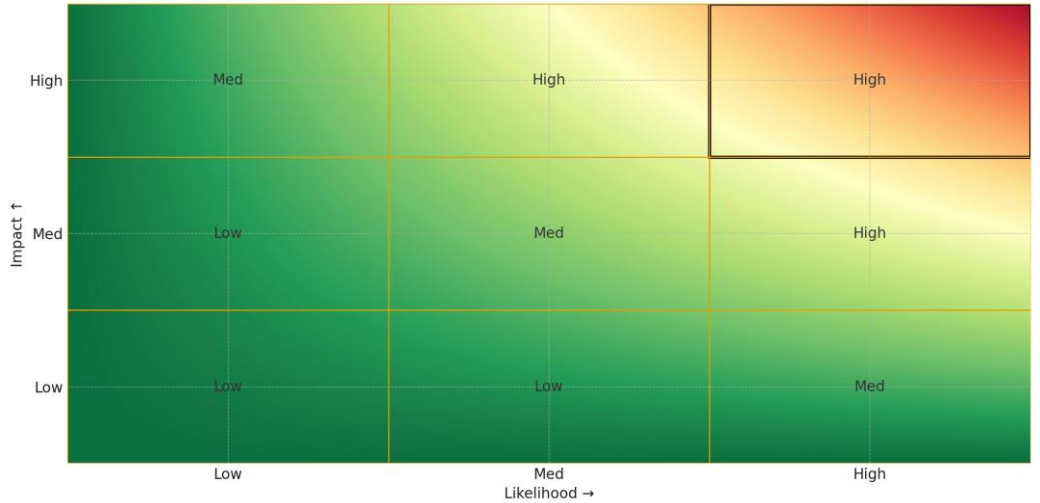
Use sparingly for critical decisions; independent checks cut joint failure probability.

Dùng có chọn lọc cho quyết định trọng yếu; lớp độc lập giúp giảm xác suất lỗi trùng.

PRIORITIZATION & PRINCIPLES

Prioritize by L×I

- **Prioritize by L×I** — Start with high-value assets & obvious big risks; quick wins later. L= likelihood, I = impact
  Bắt đầu từ tài sản/rủi ro “to & rõ ràng”; các điểm dễ xử lý sau.



- **Least Privilege** — Just enough access; deny-by-default; avoid shared admin.
  Đặc quyền tối thiểu — truy cập vừa đủ; mặc định từ chối; tránh tài khoản admin dùng chung.
- **Defense-in-Depth** — Independent layers to cut joint failures.
  Phòng thủ nhiều lớp — các lớp độc lập để giảm lỗi trùng.
- **Auditing & Logs** — Evidence without secrets/PII; structured; monitored.
  Ghi vết & log — bằng chứng không chứa bí mật/PII; có cấu trúc; được giám sát.

Four Questions – Q4: Did we do a good job?

1. QUALITY CHECK

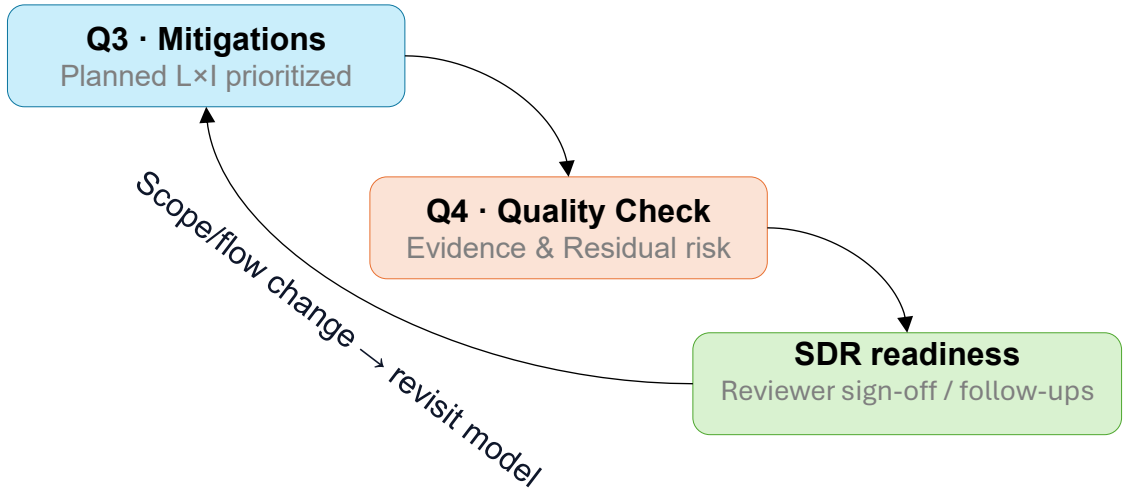
Residual risk & evidence

- **Assess residual risk coverage**
  Đánh giá rủi ro còn lại và mức độ bao phủ.
  - ✓ **Top threats addressed or justified:** mitigations in place or risk explicitly accepted with rationale.
    Các đe doạ chính đã được xử lý hoặc chấp nhận có biện giải.
  - ✓ **Coverage across assets & boundaries** : no blind spots on high-value assets or trust boundaries.
    Bao phủ theo tài sản & ranh giới tin cậy; không còn điểm mù.
  - ✓ **Clarity on scope & assumptions** : what's in/out; assumptions recorded and reviewed.
    Rõ phạm vi & giả định; ghi nhận và rà soát.
- **Verification & evidence**
  Tiêu chí kiểm chứng và bằng chứng.
  - ✓ Mitigations verified : evidence that controls work as designed (tests, reviews, walkthroughs).
    Biện pháp đã được kiểm chứng hoạt động đúng thiết kế (kiểm thử, review, walkthrough).
  - ✓ Explicit risk acceptance **:** if any mitigation is omitted, record why it's tolerable and conditions to revisit.
    Nếu bỏ qua giảm thiểu: ghi rõ lý do chấp nhận và điều kiện xem xét lại.
  - ✓ Follow-up tracked **:** agreed changes assigned & status tracked to closure.
    Theo dõi thực hiện: thay đổi đã có người phụ trách và trạng thái đến khi hoàn tất.

2. GATE READINESS & ITERATION

Ready for SDR (security design review)

- **Concise assessment** — summary of threats, mitigations, and residual risk; reviewer sign-off criteria clear.
  Đánh giá ngắn gọn: đe doạ, giảm thiểu, rủi ro còn lại; tiêu chí ký duyệt rõ.
- **Track in project system** — create/attach an SDR item to track recommendations and follow-ups.
  Theo dõi trong hệ thống dự án: tạo mục SDR để bám khuyến nghị và việc cần làm.
- **Revise the model** — update threats/assumptions when scope or environment changes.
  Cập nhật mô hình khi phạm vi hoặc môi trường đổi.
- **Re-check accepted risks** — conditions that made them tolerable may no longer hold.
  Rà soát rủi ro đã chấp nhận: điều kiện chấp nhận có thể không còn đúng.



Assets & Prioritization

CORE DEFINITION

Anything of value worth protecting in the system.

- **High-value data** — PII/PHI/Fin, trade secrets, source code, telemetry.
  Dữ liệu giá trị cao — PII/PHI/tài chính, bí mật kinh doanh, mã nguồn, telemetry.
- **Critical services** — AuthN/AuthZ, billing/payments, key-management, admin console.
  Dịch vụ quan trọng — xác thực/ủy quyền, thanh toán, quản lý khóa, bảng điều khiển quản trị.
- **Special privileges** — admin roles, deployment creds, signing keys/tokens.
  Quyền đặc biệt — vai trò admin, thông tin triển khai, khóa/ký số, token đặc quyền.
- **Secrets** — API keys, long-lived tokens, certificates.
  Bí mật — khóa API, token sống lâu, chứng thư.

Describe each asset: Owner • Sensitivity • Where stored • Retention • Impact if lost.

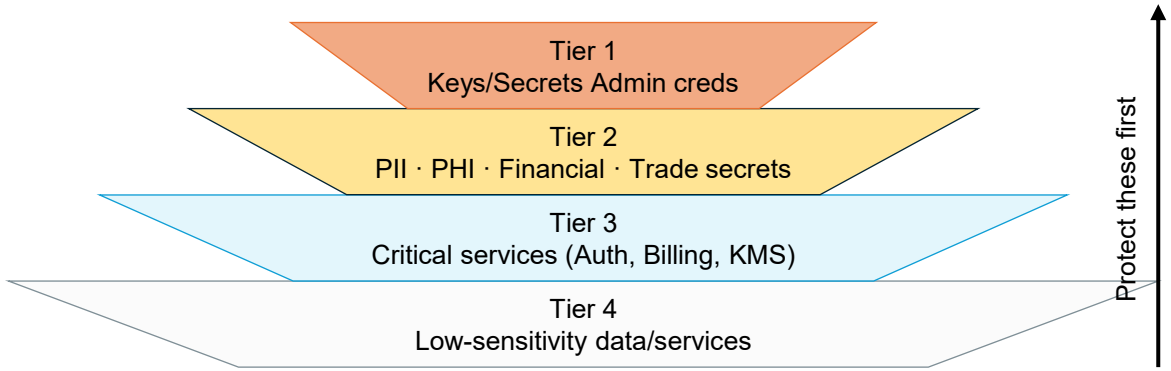
Chủ sở hữu • Độ nhạy • Nơi lưu • Thời hạn giữ • Tác động nếu mất.

Asset Name	Type	Data Category	Owner	Sensitivity	Tier	Location	Retention
Customer PII	Data	PII	Data Steward	High	2	DB: prod-eu-1 (PostgreSQL)	2 years
Payment Card Data	Data	Financial (PCI)	Payments Lead	High	2	Tokenized via PSP; vault (PCI segment)	No storage (tokenized)
Access Tokens	Secret	Secrets	IAM Owner	High	1	KMS-backed store	Rotate every 90 days
Signing Key	Secret	Keys/Certificates	Security	High	1	HSM	Rotate annually; retire on incident

PRIORITIZE ASSETS & PROTECTION SCOPE

Allocate resources for more important things

- Tier by sensitivity: Classify into tiers; protect the highest tier first.
  Phân tầng theo độ nhạy; bảo vệ tầng cao nhất trước.



Optional: P(asset) = S(asset) × E(asset), with S(Tier1..4)=4,3,2,1 ·
Priority = Sensitivity × Exposure Level

- **Set protection scope:** Start with top-tier assets and their flows; include storage, transit, and boundary crossings.
  Bắt đầu từ tài sản tầng cao và các luồng của chúng; bao gồm lưu trữ, truyền tải, và băng qua ranh giới.
- Mark locations & copies; minimize & delete what you can.
  Đánh dấu nơi lưu và các bản sao; tối thiểu hoá và xoá khi có thể.
- Assign owners; define evidence for controls (logs/tests).
  Chỉ định chủ sở hữu; xác định bằng chứng cho kiểm soát (log/kiểm thử).

Attack Surface - Concept

1. DEFINITION & SCOPE

Attack surface — what it means

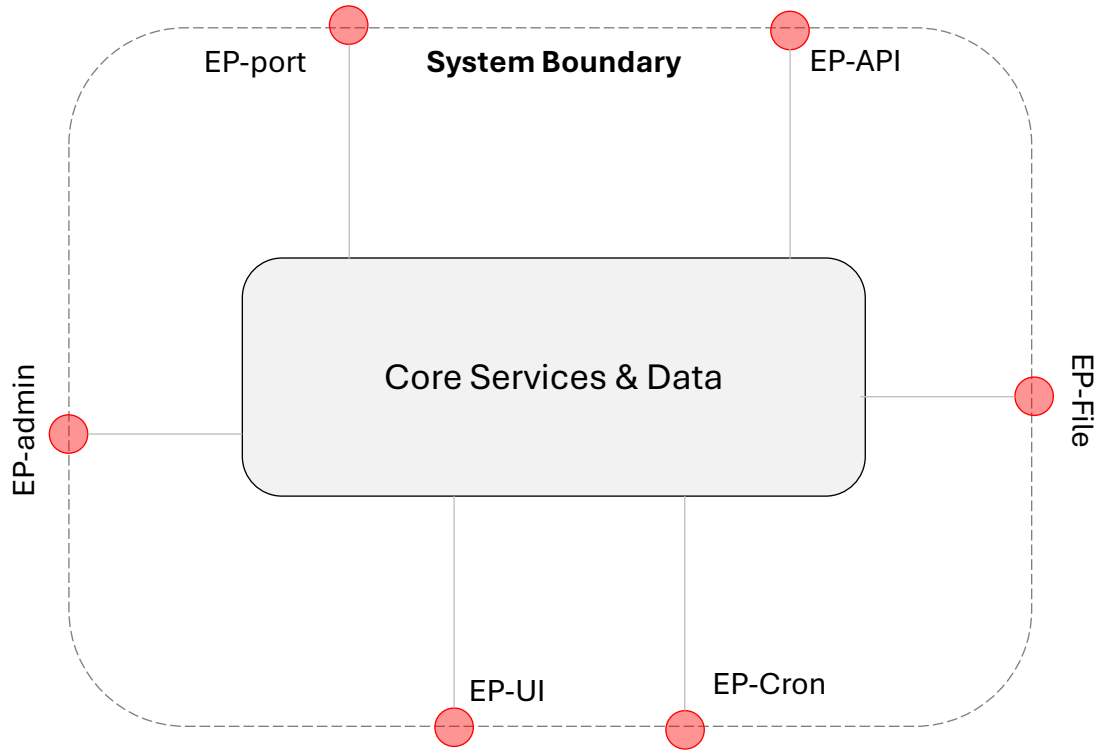
- All the places an attacker can attempt to interact with or influence the system.
  Tổng các điểm mà kẻ tấn công có thể tương tác/ảnh hưởng đến hệ thống.
  - ✓ Examples: APIs , UIs , Network ports , Files , Jobs/Cron Jobs/Cron .
    Ví dụ: API, giao diện người dùng, cổng mạng, tập tin, tác vụ định kỳ, bảng điều khiển/admin.
  - ✓ Goal: inventory comprehensively; tie each item to a concrete **Entry Point (EP)** ID.
    Mục tiêu: liệt kê đầy đủ; gắn từng mục với Entry Point (EP) cụ thể.

- Reduce what's exposed; harden what remains.
  Giảm phơi bày; gia cố phần còn lại.
  - ✓ Remove unused endpoints; restrict by network & identity.
    Xoá endpoint không dùng; giới hạn theo mạng & danh tính.
  - ✓ Standard controls at interfaces: authn/z, input contracts, rate-limits, audit logs.
    Áp kiểm soát chuẩn ở giao diện: xác thực/ủy quyền, hợp đồng đầu vào, giới hạn tần suất, ghi vết.

2. ILLUSTRATION

Attack surface map

- Mark entry points around the system boundary; this "ring" is the attack surface.
  Đánh dấu điểm vào quanh ranh giới hệ thống; "vành đai" đó là attack surface.



Entry Points & Interfaces

CATALOG THE ENTRY POINTS

HTTP Endpoint, message queues, cron jobs, console

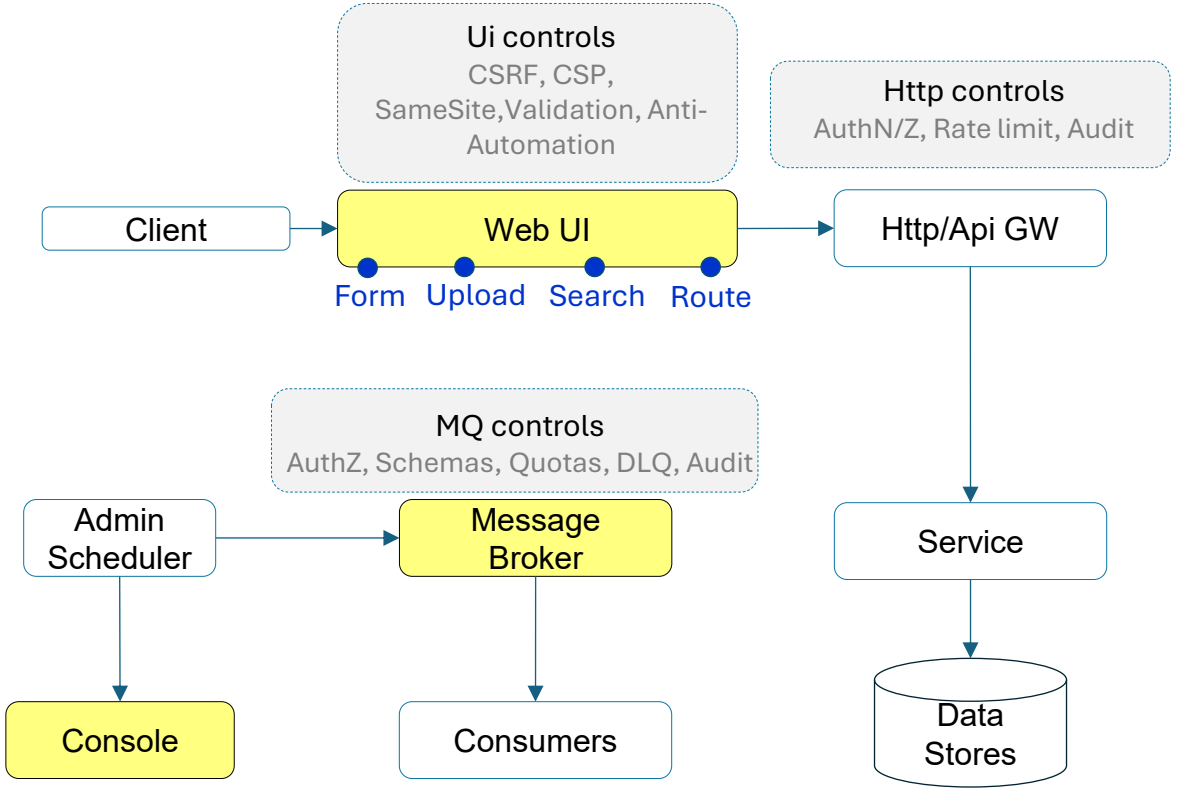
- **Web/UI pages & screens**: forms, uploads, search fields, SPA routes, deep links.
  Trang/screen giao diện: biểu mẫu, tải tệp, ô tìm kiếm, tuyến SPA, liên kết sâu.
- **HTTP endpoints** (REST/GraphQL): versioned, documented, tested.
  Endpoint HTTP (REST/GraphQL): quản lý phiên bản, có tài liệu, được kiểm thử.
- **Message queues** (Kafka/SQS/Rabbit): producers/consumers, schemas, DLQs.
  Hàng đợi thông điệp: nhà phát/tiêu thụ, lược đồ, hàng đợi lỗi (DLQ).
- **Cron/Schedulers**: jobs, triggers, permissions, blast radius.
  Cron/Lịch: tác vụ, kích hoạt, quyền hạn, phạm vi ảnh hưởng.
- **Admin consoles**: internal-only, SSO/MFA, audit by default.
  Bảng điều khiển quản trị: chỉ nội bộ, SSO/MFA, ghi vết mặc định.

Interfaces = chokepoints: Centralize controls where traffic must pass.
Tập trung kiểm soát ở nơi mọi lưu lượng phải đi qua.

VISUALIZATION

Interfaces as security chokepoints

Place independent controls at each interface to reduce joint failure risk.
Đặt kiểm soát độc lập tại mỗi giao diện để giảm rủi ro lỗi trùng.



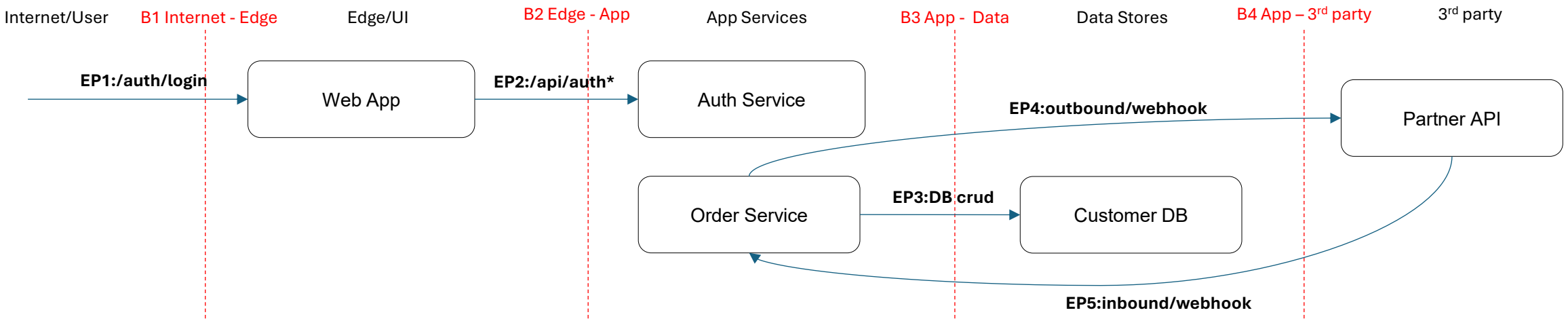
Trust Boundary

DEFINITION

Trust bouderies concept and practical example

- **Definition:** A demarcation between components with different trust levels; any data crossing must be verified.
  Đường phân tách mức tin cậy khác nhau; mọi dữ liệu băng qua phải được xác minh.
- **Examples:** Internet→DMZ; Frontend→Backend; App→DB; Partner API.
  Ví dụ: Internet→DMZ; Frontend→Backend; App→DB; Partner API.
- **SSH daemon** separates privileged and unprivileged processes; web frontend straddling the Internet boundary.
  SSH daemon tách tiến trình đặc quyền và không đặc quyền; web frontend ở ranh giới Internet.
- **Interfaces/protocols:** Always cross via explicit interface/protocol; minimize side-effects.
  Luôn chuyển qua giao diện/protocol rõ ràng; giảm tác động side-effects.

TRUST BOUNDARY - EXAMPLE



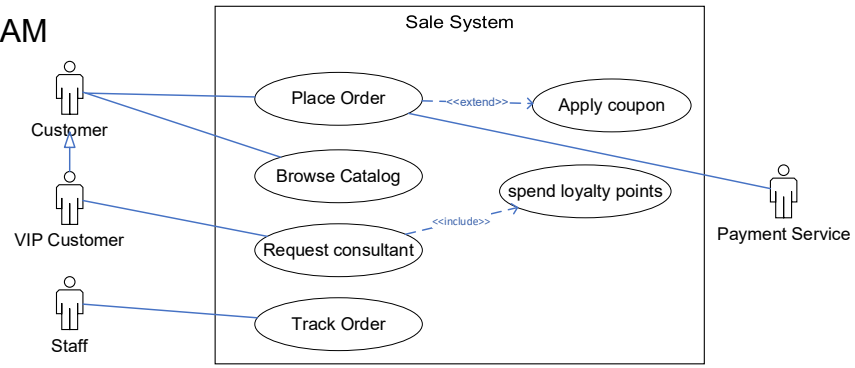
Use-case Diagram

DEFINITION

What a use-case diagram shows

- High-level functions & scope of a system.
  Các chức năng mức cao và phạm vi của hệ thống.
- Interactions between the system and its actors (external environment).
  Tương tác giữa hệ thống và các tác nhân (môi trường bên ngoài).
- What the system does, not internal implementation
  Diễn tả hệ thống làm gì, không mô tả cách triển khai bên trong.
- Key elements includes actor, use-case, system boundary, associations
  Các thành phần chính bao gồm : actor, use-case, system boundary, associations .
  - ✓ Actors — people, organizations, machines, or other systems.
    Tác nhân: con người, tổ chức, máy móc, hay hệ thống khác.
  - ✓ Use cases — goals/services the system provides (ovals).
    Ca sử dụng: mục tiêu/dịch vụ hệ thống cung cấp (hình oval).
  - ✓ System boundary — rectangle to separate internal use cases from external actors (visual aid).
    Ranh giới hệ thống: hình chữ nhật tách ca sử dụng bên trong khỏi tác nhân bên ngoài (trợ giúp trực quan).
  - ✓ Associations — lines showing which actors participate in which use cases.
    Liên kết: đường thẳng cho thấy tác nhân tham gia ca sử dụng nào.

USE-CASE DIAGRAM



USE-CASE DESCRIPTION

Use-case field	Description
Use case name	An active verb phrase that describes a particular task.
Subject area	A use role or other grouping mechanism that can be used to group use cases.
Business event	A trigger that stimulates activity within the business. Many business events occur at the interface point between the business and one of the external entities with which it interacts. Business events must be observable.
Actors	The actor that initiates this use case and all users who participate in this use case.
Use case overview	A description of the overall scope and content of the use case.
Preconditions	Constraints that must be met for the use case to be taken by the solution developer and used to create a workflow. This might include a required sequencing of use cases. For example, one or more other use cases might need to be performed successfully for this use case to begin.
Termination outcome	A list of the successful and unsuccessful ways this use case might end. What are the possible ending results?
Condition affecting termination outcome	A list of the conditions under which the corresponding termination outcome occurs.
Use case description	A brief description of events for the most likely termination outcome. List the actions the actor does and how the system responds.
Use case associations	A list of other use cases that are associated with this use case.
Traceability to	A list of other related documents, models, and products that are associated with this use case.
Input summary	A brief summary that lists the data input by the actor.
Output summary	A brief summary that lists the data output by the system.
Usability index	A number based on how this use case ranked in terms of satisfaction, importance, and frequency.
Use case notes	Information that is not directly part of this use case but that the solution developer needs to be aware of while working on the workflow.

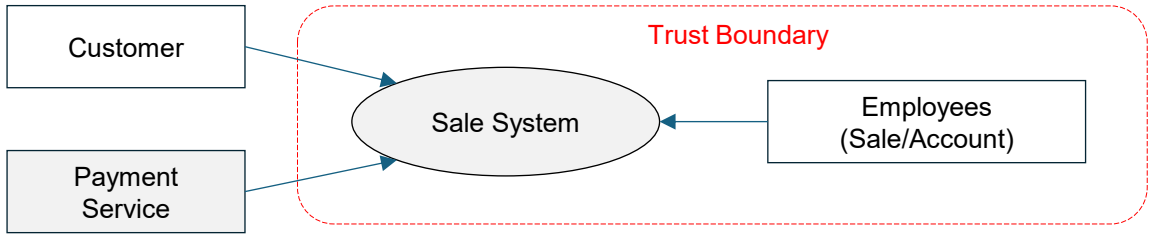
Context Diagram & Preliminary Data Flow Diagram

CONCEPT

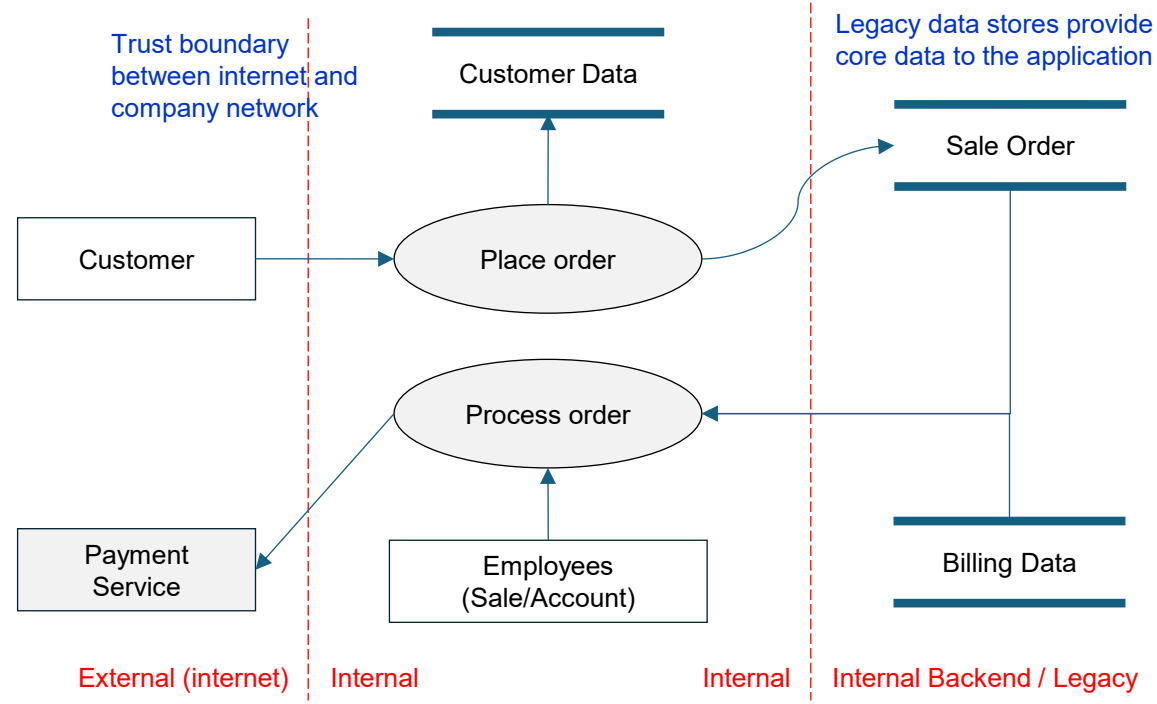
- **Context Diagram**
  - ✓ **Black-box system view**: show the system as a single process and surrounding external entities.
    Xem hệ thống như "hộp đen" và các hệ thống/tác nhân lân cận.
  - ✓ **Label data on arrows**: annotate exchanged data types (e.g., PII, Orders, Payments, Notifications).
    Gắn loại dữ liệu trên mũi tên (PII, đơn hàng, thanh toán, thông báo…).
  - ✓ **Trust boundary:** the system boundary is a key trust boundary—treat external inputs as untrusted; verify at crossing.
    Ranh giới tin cậy: ranh hệ thống là ranh chủ đạo — coi dữ liệu bên ngoài là không tin cậy; xác minh khi băng qua.
- **Preliminary DFD**
  - ✓ **DFD-0: top-level processes & major data stores/flows; focus on where data crosses trust boundaries.**
    DFD-0: các tiến trình mức cao & kho dữ liệu/luồng chính; nhấn điểm băng ranh tin cậy.
  - ✓ **DFD-1:** expand one DFD-0 process into sub-processes and detailed flows.
    DFD-1: bẻ nhỏ một tiến trình DFD-0 thành các tiểu tiến trình và luồng chi tiết.
  - ✓ **Security lens:** highlight untrusted sources, boundary crossings, validation/authN/authZ, logging, and data classification.
    Góc nhìn bảo mật: tô sáng nguồn không tin cậy, điểm băng ranh, kiểm tra hợp lệ/xác thực/ủy quyền, logging và phân loại dữ liệu.
- **Checklist (security)**

Mark trust boundaries	AuthN/AuthZ	Least privilege	Protocol contracts
Untrusted → trusted	Validate inputs	Encrypt in transit	Audit logs

CONTEXT DIAGRAM



DFD



Part 2
STRIDE Model for Threat Classification

Categorizing Threats with STRIDE

CONCEPT & MEANING

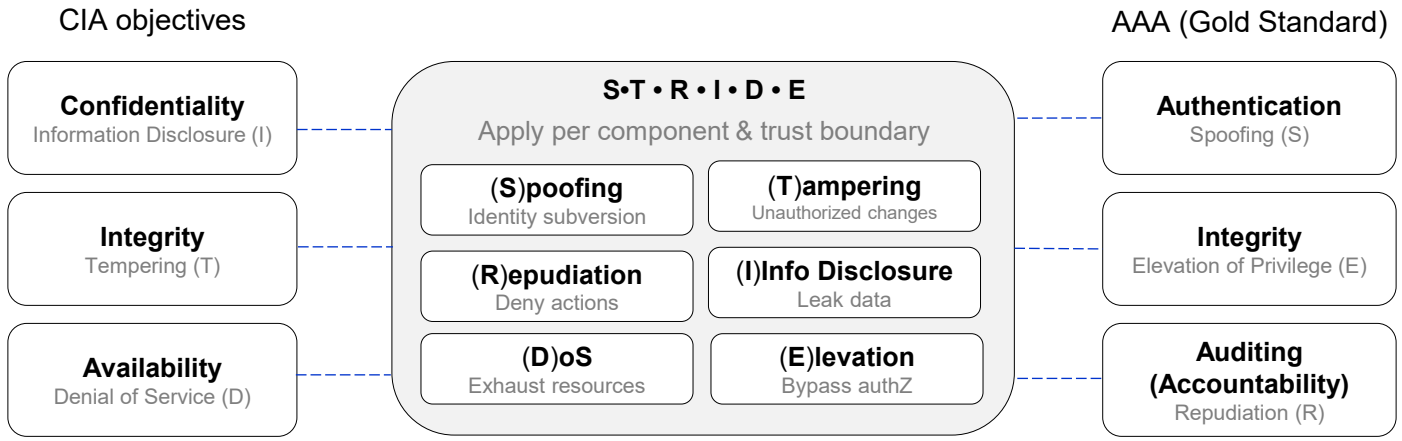
What STRIDE is

- **STRIDE** is Microsoft's threat classification model.
  STRIDE là mô hình phân loại mối đe doạ (threat classification) của Microsoft
- **Taxonomy, not a full method.** A mnemonic of six threat categories to ensure coverage.
  Hệ phân loại, không phải phương pháp đầy đủ; 6 hạng mục giúp không bỏ sót.
- **Apply to each element & crossing.** Walk the model component-by-component and boundary-by-boundary.
  Áp dụng cho từng thành phần và từng lần băng qua ranh giới.
- **Kickoff technique.** Write S-T-R-I-D-E on the board; brainstorm quickly before deep dive.
  Khởi động: ghi S-T-R-I-D-E lên bảng; động não nhanh trước khi đi sâu.

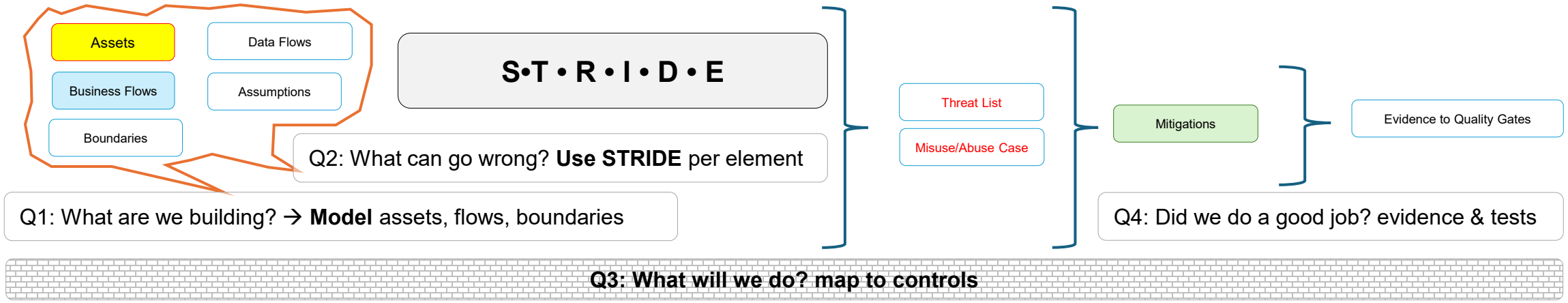
ILLUSTRATION

STRIDE ↔ CIA & AAA

Three map to CIA; three to AAA (Gold Standard).
Ba hạng mục ánh xạ CIA; ba hạng mục ánh xạ AAA.



Where STRIDE fits (Matching with 4 Questions)

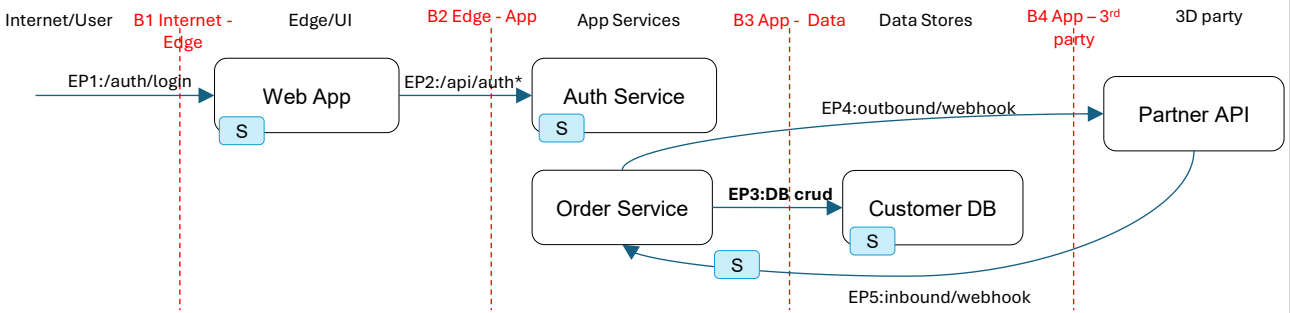


STRIDE – S: Spoofing

DEFINITION & OBJECTIVE

- **Spoofing** = the attacker pretends to be a legitimate entity (user/service/device) to bypass **authentication (AuthN)**.
  Mạo danh = kẻ tấn công giả dạng *thực thể hợp lệ* (người dùng/dịch vụ/thiết bị) để vượt qua xác thực .
- At trust boundaries, verify identity on every crossing.
  Mọi lần băng qua *trust boundary* phải xác minh danh tính.
- **Typical Scenarios mapped to EP/Boundaries**
  - ✓ **Password theft / phishing / credential stuffing** at EP1: /auth/login (B1 Internet→Edge).
    Đánh cắp mật khẩu / phishing / credential stuffing tại EP1 /auth/login (B1 Internet→Edge).
  - ✓ **Token replay / session hijack** — stolen token/cookie is *reused* across boundaries (B1/B2/B4).
    Token replay / chiếm phiên — token/cookie bị lộ rồi *dùng lại* qua ranh giới (B1/B2/B4).
  - ✓ **Service-to-service spoofing** on B2 Edge→App & B4 App↔3rd-Party: forged webhook, internal call without service identity.
    Giả danh dịch vụ trên B2 Edge→App & B4 App↔3rd-Party: webhook giả mạo, gọi nội bộ không có danh tính máy-khách.
  - ✓ **App→DB impersonation** on B3 App→Data: shared credentials, broad privileges, no service-identity binding.
    Giả danh tới DB trên B3 App→Data: chuỗi kết nối dùng chung, quyền rộng, không ràng buộc danh tính dịch vụ.
- **Risks & Impact / Rủi ro & Tác động**
- **Unauthorized access** to accounts/assets (PII/funds/orders).
  Truy cập trái phép vào tài khoản/tài sản (PII/tiền/đơn hàng).
- **Broken chain of trust** leading to forged transactions/command injection.
  Chuỗi tin cậy hỏng dẫn tới giả mạo giao dịch/tiêm lệnh.

ILLUSTRATION



MITIGATIONS — MAP TO Q3 LEVERS

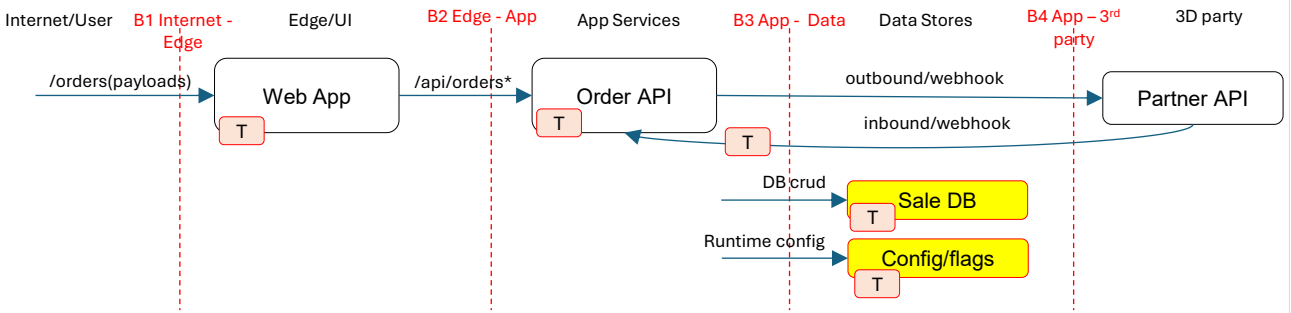
- **MFA** for login/sensitive actions; OTP/TOTP/WebAuthn; protect recovery flows.
  MFA cho đăng nhập/hành động nhạy cảm; bảo vệ luồng khôi phục.
- **mTLS & strong service identity**; client cert, SPKI pinning, rotation; avoid shared secrets.
  mTLS & danh tính dịch vụ; pin SPKI, xoay vòng; tránh secret dùng chung.
- **Anti-replay**; nonce + timestamp, JWT exp/jti, one-time tokens, channel binding (SameSite/token binding).
  Chống phát lại; nonce + timestamp, token một lần, ràng buộc kênh.
- **Session locking/rotation** on sensitive changes/MFA; short TTL; bind to UA.
  Khoá/đổi phiên sau thay đổi nhạy cảm/MFA; TTL ngắn; ràng buộc vào UA.
- **Rate limiting & backoff** on /auth and webhooks (per-IP/ASN/account).
  Giới hạn tần suất tại /auth & webhook (theo IP/ASN/tài khoản).
- **Observability**: auth logs, mTLS cert subject, webhook signature pass/fail; idempotent retries.
  Quan sát được: log xác thực, subject cert mTLS, chữ ký webhook (đỗ/trượt); retry bất biến.

STRIDE – T: Tampering

DEFINITION & OBJECTIVE

- **Tampering** = unauthorized modification of *data/code/config* in transit or at rest → breaks **Integrity**.
  Sửa đổi trái phép đối với *dữ liệu/mã/cấu hình* khi truyền hoặc lưu → vi phạm tính toàn vẹn.
- Focus: ensure changes are **authentic, intentional, and traceable**.
  Trọng tâm: mọi thay đổi phải **xác thực, có chủ đích, truy vết được**.
- **Typical Scenarios**
  - ✓ **API payload manipulation** at EP `/order`, `/transfer` across B1/B2.
    Chỉnh sửa nội dung yêu cầu (JSON/headers) tại entry point.
  - ✓ **File/content tampering**: uploaded files, stored objects, log records.
    Sửa tệp tải lên, đối tượng lưu trữ, hoặc log.
  - ✓ **Config/feature-flag drift** in prod; unauthorized edits to runtime config.
    Sai lệch cấu hình/feature flag; chỉnh cấu hình chạy không qua kiểm soát.
  - ✓ **DB record tampering** across B3 App→DB (e.g., order amounts, roles, limits).
    Sửa dữ liệu trong DB (giá trị đơn, vai trò, hạn mức).
  - ✓ **Partner/webhook payload** altered at B4 App↔3rd-Party.
    Payload webhook bị thay đổi trên đường hoặc bị giả mạo.
- **Risks & Impact**
  - ✓ **Corrupted state**: financial loss, fraud, policy violations.
    Trạng thái sai lệch: thất thoát, gian lận, vi phạm chính sách.
  - ✓ **Durable integrity loss** if persisted to DB/logs/backups.
    Mất toàn vẹn bền vững nếu đã ghi vào DB/log/backup.

ILLUSTRATION



MITIGATIONS — MAP TO Q3 LEVERS

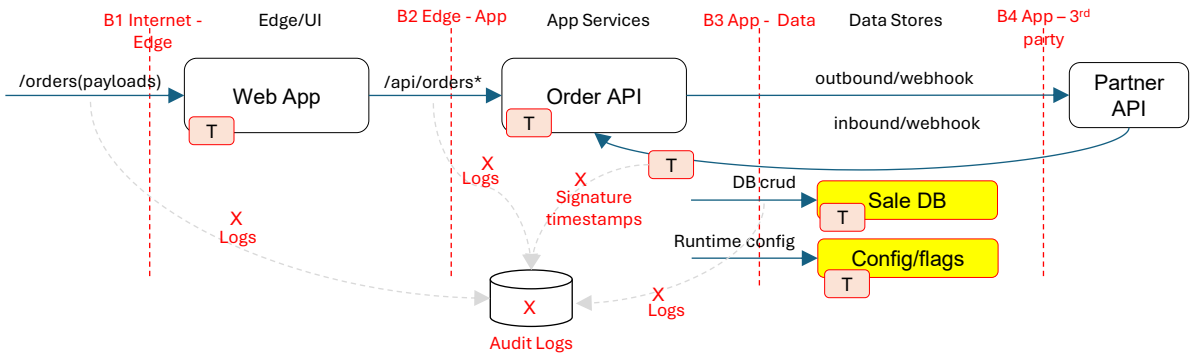
- **MAC/HMAC** for API bodies & webhooks; canonicalize then compute; reject on mismatch.
  MAC/HMAC cho payload API/webhook; chuẩn hoá trước khi tính; lệch là từ chối.
- **Digital signatures** (request/message/artifact); verify key ownership & timestamps.
  Chữ ký số cho yêu cầu/thông điệp/artefact; kiểm khoá & thời gian.
- **Checksums & content hashing** for files/objects; compare on download/ingest.
  Checksum/băm nội dung cho tệp/đối tượng; so sánh khi tải/xử lý.
- **Immutable infrastructure & artefacts**: image digests, signed releases, no ad-hoc prod edits.
  Hạ tầng & artefact bất biến: bám digest, phát hành có chữ ký, cấm sửa tay trên prod.
- **DB integrity controls**: constraints, append-only/event sourcing, signed/audited logs.
  Kiểm soát toàn vẹn DB: ràng buộc, ghi-bổ-sung, log ký/audit.
- **Config governance**: IaC, code-review/2-person rule, change history, least-privilege on writes.
  Quản trị cấu hình: hạ tầng-như-mã, review 2 người, lịch sử thay đổi, tối thiểu quyền ghi.

STRIDE – R: Repudiation

DEFINITION & OBJECTIVE

- **Repudiation** = an actor can deny performing an action because the system lacks trustworthy evidence (logging/audit).
  Chối bỏ = tác nhân có thể phủ nhận hành động do hệ thống thiếu bằng chứng tin cậy (log/audit).
- **Objective**: ensure accountability with tamper-evident logs & verifiable provenance.
  Mục tiêu: bảo đảm trách nhiệm giải trình bằng log chống sửa & nguồn gốc có thể xác minh.
- AAA mapping: Repudiation ↔ Auditing/Accountability.
  Ánh xạ AAA: Chối bỏ ↔ Ghi vết/Trách nhiệm.
- **Typical Scenarios (EP/Boundaries) / Tình huống (EP/B)**
  - ✓ **Missing or weak logs** at EP1 /auth/login, EP2 /orders across B1/B2 ⇒ cannot attribute actions.
    Thiếu/yếu log tại /auth, /orders qua ranh B1/B2 ⇒ không quy trách nhiệm.
  - ✓ **Shared accounts** for admins / service users across B2/B3 ⇒ who did what?
    Tài khoản dùng chung (admin/dịch vụ) qua B2/B3 ⇒ không biết ai làm gì.
  - ✓ **Webhook/3rd-party callbacks** over B4 without signatures/timestamps ⇒ source cannot be proven.
    Webhook/Callback bên thứ ba qua B4 không có chữ ký/dấu thời gian ⇒ không chứng minh được nguồn.
  - ✓ **Log tampering** in transit/store (no integrity, no WORM) ⇒ audit cannot be trusted.
    Bị sửa log trên đường truyền/lưu trữ (không đảm bảo toàn vẹn, không WORM) ⇒ audit mất tin cậy.
- **Risks & Impact**
  - ✓ **No accountability**: disputes cannot be resolved; incident RCA stalls.
    Mất trách nhiệm giải trình: tranh chấp không xử lý; điều tra sự cố bế tắc.
  - ✓ **Compliance & forensics** weakened; fraud cannot be proven/contained.
    Tuân thủ & giám định suy yếu; không chứng minh/khống chế được gian lận.

ILLUSTRATION



MITIGATIONS — MAP TO Q3 LEVERS

- **Immutability**: append-only/WORM, write-once buckets, restricted admin path.
  Bất biến: chỉ-ghi/ghi 1-đọc nhiều, bucket write-once, hạn chế đường quản trị.
- **Cryptographic integrity**: per-event signatures, hash-chains, merkle/periodic sealing.
  Toàn vẹn số: ký từng sự kiện, chuỗi băm, niêm phong định kỳ.
- **Provenance**: include sub/client_id, mTLS peer, request-id, trace_id, timestamp with synchronized clocks.
  Nguồn gốc: ghi chủ thể, danh tính mTLS, request-id/trace-id, dấu thời gian (đồng bộ giờ).
- **Accountability controls**: unique accounts (no sharing), RBAC/SoD for log access, audit of admin actions.
  Kiểm soát trách nhiệm: tài khoản riêng (không dùng chung), RBAC/SoD cho truy cập log, audit hành động admin.
- **Signed inbound**: require HMAC/PKI signatures + anti-replay for webhooks/3rd-party callbacks.
  Ký số inbound: bắt buộc chữ ký HMAC/PKI + chống phát lại cho webhook/callback.
- **Retention & privacy**: retention policy, PII minimization/redaction, access reviews.
  Lưu trữ & riêng tư: chính sách lưu trữ, giảm/ẩn PII trong log, rà soát quyền.

FPT Education
FPT UNIVERSITY

STRIDE – I: Information Disclosure

DEFINITION & OBJECTIVE

- **Information Disclosure** = exposing data to unauthorized parties (content or *metadata*).
  Rò rỉ thông tin = dữ liệu lộ ra cho đối tượng không được phép (nội dung hoặc *metadata*).
- **Targets**: PII/PHI/financials, keys/secrets, configs, schemas, logs, and metadata (filenames, sizes, timings).
  Mục tiêu: PII/PHI/tài chính, khóa/bí mật, cấu hình, lược đồ, log, và metadata (tên file, kích thước, thời điểm).
- CIA mapping: I ↔ Confidentiality .
  Ánh xạ CIA: I ↔ Bảo mật dữ liệu.
- **Typical Scenarios**
  - ✓ **Unsecured channels**: missing TLS/HSTS, internal plaintext links; mixed-content.
    Kênh không an toàn: thiếu TLS/HSTS, liên kết nội bộ không mã hoá; nội dung trộn.
  - ✓ **At-rest leaks**: DB/files/backups/snapshots not encrypted; public object storage.
    Rò rỉ khi lưu trữ: DB/file/backup/snapshot không mã hoá; storage công khai.
  - ✓ **Verbose outputs**: error pages, debug endpoints, logs/metrics/traces contain PII, tokens.
    Đầu ra quá chi tiết: trang lỗi, endpoint debug, log/metrics/traces chứa PII, token.
  - ✓ **Side-channels**: response timing/size, cache/probing, enumeration via metadata.
    Kênh phụ: thời gian/kích thước phản hồi, cache/probing, liệt kê qua metadata.
- **Risks & Impact**
  - ✓ **Privacy & regulatory exposure** (PII/PHI/fin); **key compromise** leads to broader breaches.
    Rủi ro riêng tư & pháp lý (PII/PHI/tài chính); lộ khóa kéo theo xâm phạm diện rộng.
  - ✓ **Intelligence for attackers** via metadata & errors; enables chaining to S/T/E/D threats.
    Thông tin tình báo cho kẻ tấn công qua metadata & lỗi; tạo tiền đề kết hợp với S/T/E/D.

MITIGATIONS — MAP TO Q3 LEVERS

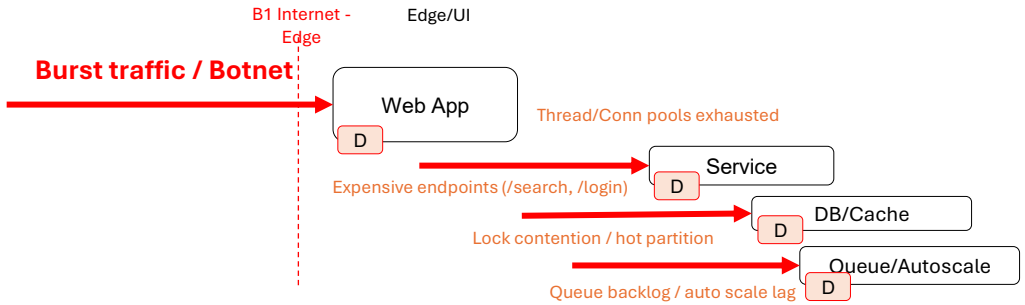
- **TLS everywhere**: TLS 1.2+/1.3, HSTS, no mixed-content; internal TLS, cert rotation.
  Mã hoá khi truyền: TLS 1.2+/1.3, HSTS, tránh nội dung trộn; TLS nội bộ, xoay vòng chứng chỉ.
- **Encrypt at rest**: DB TDE/disk, object storage SSE-KMS, envelope encryption; key rotation & access reviews.
  Mã hoá khi lưu: TDE/đĩa, SSE-KMS, envelope; xoay khóa & rà soát quyền.
- **Least information**: redact PII in logs, minimal error messages, pre-signed URLs, no directory listing.
  Tối thiểu thông tin: ẩn PII trong log, thông báo lỗi tối giản, URL ký thời hạn, không liệt kê thư mục.
- **Access controls**: RBAC/ABAC to data & keys; split vault/KMS roles; monitor exfiltration (DLP).
  Kiểm soát truy cập: RBAC/ABAC cho dữ liệu & khóa; tách vai trò vault/KMS; giám sát rò rỉ (DLP).
- **Side-channel hardening**: constant-time crypto ops, padding/length-hiding, cache rules.
  Giảm kênh phụ: thao tác mật mã hằng thời gian, đệm/ẩn độ dài, quy tắc cache.
- **Data minimization**: collect only what you need; classify assets; purge stale data/keys.
  Giảm dữ liệu: chỉ thu thập cần thiết; phân loại tài sản; dọn dữ liệu/khóa cũ.

STRIDE – D: Denial of Service

DEFINITION & OBJECTIVE

- **Denial of Service** = making a system/service **unavailable** by exhausting resources or blocking progress.
  Từ chối dịch vụ = làm hệ thống không phục vụ bằng cách cạn kiệt tài nguyên hoặc cản trở tiến trình.
- **Objective**: protect **Availability** via capacity, isolation, and graceful degradation under stress.
  Mục tiêu: bảo vệ Khả dụng bằng dung lượng, cách ly và suy giảm có kiểm soát khi quá tải.
- **Typical Scenarios**
  - ✓ **Burst traffic & bots** at B1 (e.g., /search, /login) → CPU, I/O spikes; cache stampede.
    Lưu lượng đột biến & bot tại B1 (ví dụ /search, /login) → tăng đột biến CPU/I-O; dồn tải cache.
  - ✓ **Resource exhaustion**: thread/connection pools, memory/CPU, per-tenant quotas.
    Cạn tài nguyên: pool luồng/kết nối, RAM/CPU, hạn ngạch theo tenant.
  - ✓ **Lock contention & hot partitions** in DB/kv; synchronized sections bottleneck.
    Tranh chấp khoá & phân vùng nóng trong DB/kv; đoạn mã đồng bộ gây nghẽn.
  - ✓ **Amplification**: aggressive retries without backoff; fan-out calls; N+1 queries; chatty protocols.
    Khuếch đại: retry không backoff; gọi phân nhánh; N+1 query; giao thức "lắm lời".
  - ✓ **State exhaustion**: session store/queue backlog; long-lived websockets.
    Cạn trạng thái: kho phiên/hàng đợi dồn ứ; websocket kéo dài.
- **Risks & Impact**
  - ✓ **SLA/SLO breaches**, cascading failures, incident response load.
    Vượt ngưỡng SLA/SLO, lỗi dây chuyền, tăng tải ứng cứu.
  - ✓ **Business loss**: downtime, abandoned carts, penalties.
    Mất mát kinh doanh: gián đoạn, bỏ giỏ hàng, phạt vi phạm.

ILLUSTRATION



MITIGATIONS — MAP TO Q3 LEVERS

- **Rate limiting & quotas**: token/leaky bucket; per-IP/API-key/account; 429 with Retry-After.
  Giới hạn tần suất & hạn ngạch: token/leaky bucket; theo IP/API-key/tài khoản; 429 kèm Retry-After.
- **Backpressure & load shedding**: bounded queues, early drops, prioritize health-checks.
  Phản áp & bỏ bớt tải: hàng đợi có giới hạn, loại sớm, ưu tiên kiểm tra sống.
- **Timeouts, circuit breakers, bulkheads**: fail-fast, isolate resources to prevent cascade.
  Timeout, cầu dao, vách ngăn: fail-fast, cách ly tài nguyên tránh lan rộng.
- **Autoscale & capacity**: HPA/queue-based scaling; warm cache; CDN; precompute hot views.
  Tự mở rộng & dung lượng: scale theo HPA/hàng đợi; làm ấm cache; dùng CDN; tiền tính kết quả nóng.
- **Cache stampede control**: TTL jitter, single-flight, request coalescing.
  Chống dồn tải cache: xáo TTL, hợp nhất yêu cầu, chỉ một tiến trình xây cache.
- **Retries with jitter & budgets**: idempotency keys; client-side backoff; per-request time budgets.
  Retry có jitter & ngân sách thời gian: khoá bất biến; backoff phía client; ngân sách thời gian theo yêu cầu.

STRIDE – E: Denial of Service

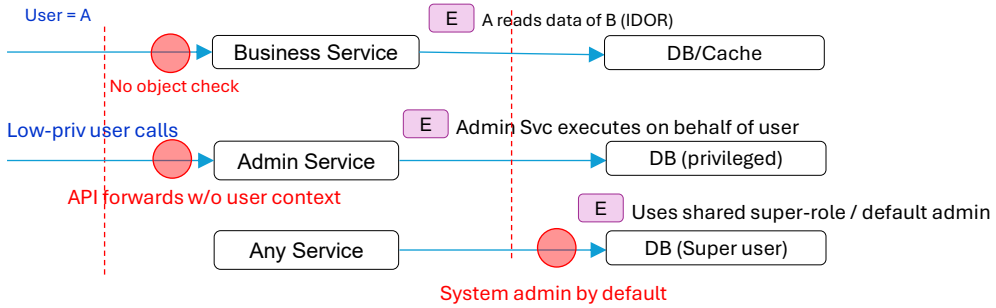
DEFINITION & OBJECTIVE

- **Elevation of Privilege** = gaining **more authority** than intended (role/scope/resource) by bypassing **authorization (AuthZ)** or abusing defaults/flows.
  Leo thang đặc quyền = đạt mức quyền cao hơn dự kiến (vai trò/phạm vi/tài nguyên) do lách ủy quyền hoặc lạm dụng mặc định/luồng.
- **Objective**: preserve **privilege boundaries** for users & services; every action must be explicitly authorized.
  Mục tiêu: giữ vững ranh giới đặc quyền cho người dùng & dịch vụ; mọi hành động phải được ủy quyền rõ ràng.
- **Typical Scenarios (mapped to EP/Boundaries)**
  - ✓ **Bypass AuthZ**: missing object-level checks (IDOR/BOLA) on EP2 /api/* across B2.
    Vượt ủy quyền: thiếu kiểm tra quyền theo đối tượng (IDOR/BOLA) tại API qua B2.
  - ✓ **Confused Deputy**: service with high privilege performs action for a low-privileged caller across B2→B3 (no caller intent binding).
    Deputy nhầm vai: dịch vụ đặc quyền cao hành động thay người gọi quyền thấp qua B2→B3 (không ràng buộc ý định người gọi).
  - ✓ **Insecure defaults**: default admin creds/roles, wide *:* IAM policies; DB runs as superuser on B3.
    Mặc định nguy hiểm: tài khoản/quyền admin mặc định, policy IAM quá rộng; DB chạy quyền siêu người dùng ở B3.
  - ✓ **Policy gaps**: allow-lists on B4 webhooks but no **AuthZ** per resource/action; partial checks only at login, not at use (no **complete mediation**).
    Lỗ hổng policy: có allow-list webhook nhưng thiếu ủy quyền theo tài nguyên/hành động; chỉ kiểm tra lúc đăng nhập, không kiểm ở thời điểm dùng (thiếu kiểm soát toàn phần).
- **Risks & Impact**

✓ **Unauthorized actions** (read/modify/delete, money movement, key access) → compromise of assets & integrity.
Hành động trái phép (đọc/sửa/xóa, chuyển tiền, truy cập khóa) → xâm phạm tài sản & tính toàn vẹn.

✓ **Blast radius** escalates when services use shared super-roles or long-lived broad tokens.
Phạm vi thiệt hại nới rộng khi dịch vụ dùng vai trò chung cấp cao hoặc token rộng, sống lâu.

ILLUSTRATION

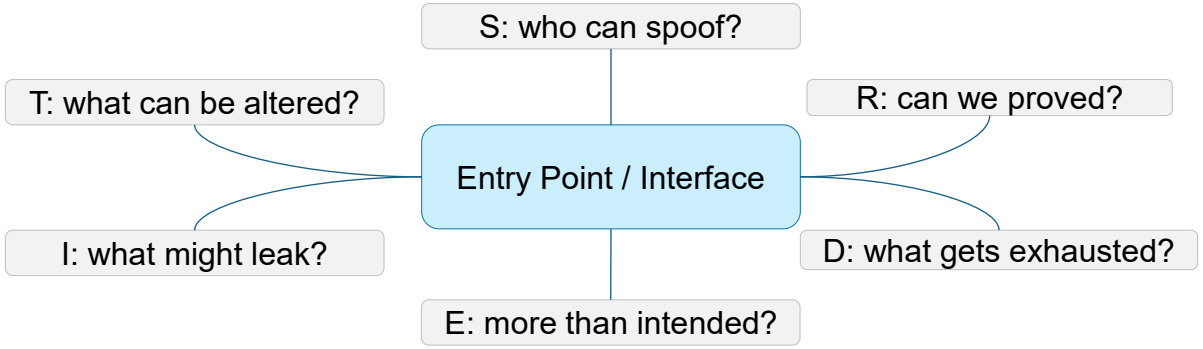


MITIGATIONS — MAP TO Q3 LEVERS

- **Least privilege**: fine-grained RBAC/ABAC, deny-by-default, resource-level checks; short-lived scoped tokens.
  Đặc quyền tối thiểu: RBAC/ABAC chi tiết, mặc định từ chối, kiểm theo tài nguyên; token có phạm vi & thời hạn ngắn.
- **Complete mediation**: enforce AuthZ at **every** API/resource call; validate subject/scope/resource on use.
  Kiểm soát toàn phần: ủy quyền ở mọi lần gọi; xác minh chủ thể/phạm vi/tài nguyên tại thời điểm dùng.
- **Separation of duties (SoD)**: split powerful actions (approve vs execute), dual control for money/keys.
  Phân tách nhiệm vụ: tách vai trò hành động mạnh (duyệt vs thực thi), kiểm soát kép cho tiền/khóa.
- **Service identity binding**: distinguish *user-as-caller* vs *service-as-caller*; propagate caller intent (delegation tokens).
  Ràng buộc danh tính dịch vụ: phân biệt người dùng gọi vs dịch vụ gọi; truyền ý định người gọi (delegation token).
- **Harden defaults**: remove default admin, principle of safe defaults, policy-as-code with reviews.
  Mặc định an toàn: bỏ tài khoản/quyền admin mặc định, áp dụng mặc định an toàn, policy-as-code có rà soát.
- **Drop privileges**: run DB/app with minimum roles; break glass with audit only.
  Giảm quyền: chạy DB/app bằng quyền tối thiểu; quyền khẩn cấp phải có audit.

STRIDE — Heuristics & Checklist

ENTRY POINT REVIEW



Don't forget
AuthZ + Immutable Audit

Outputs:
Threat List – Owners – $L \times I$

CHECKLIST – STRIDE PER ELEMENT

DFD Element	Likely STRIDE	Recommend to check
External Entity (Actor)	S	Identity proofing, MFA, anti-spoof
Process/Service	T, R, I, D, E	Input validation, authZ per action, logs, quotas
Data Flow	T, I, D	TLS, integrity (MAC), size/DoS controls
Data Store	T, R, I, D	At-rest encryption, immutability, access reviews
Trust Boundary	All (S/T/R/I/D/E)	Re-verify identities, sanitize, monitor

HOW TO USE STRIDE IN PRACTICE

- **Walk every entry point & boundary** with the six questions; capture threats, *controls*, and *owners*.
  Đi lần lượt từng điểm vào & ranh giới với 6 câu; ghi nhận đe dọa, *kiểm soát*, và *người chịu trách nhiệm*.
- **Cross-check Auditing & AuthZ** for each scenario: do we have immutable logs and *object-level* authorization at time-of-use?
  Đối chiếu Log & Ủy quyền cho từng kịch bản: có log bất biến và kiểm tra quyền theo đối tượng tại thời điểm sử dụng chưa?
- **Prioritize** with $L \times I$ and revisit after scope changes.
  Ưu tiên theo L×I và xem lại khi phạm vi thay đổi.

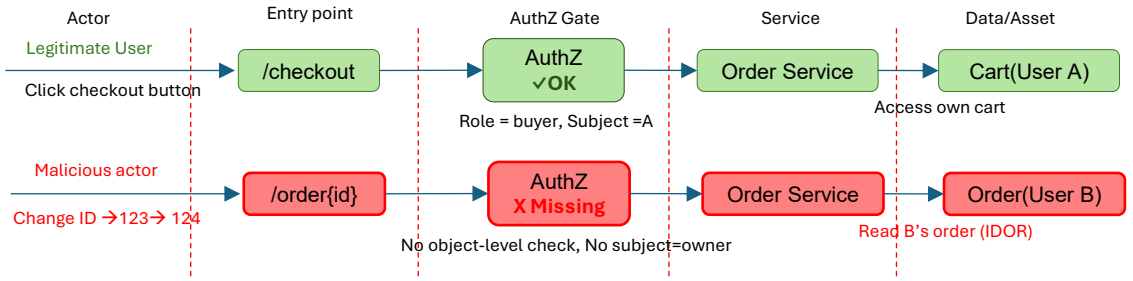
Tip: When short on time, apply "STRIDE per *interaction*" first (flows across boundaries), then go deeper per element.
Mẹo: Thiếu thời gian, hãy áp dụng "STRIDE theo *tương tác*" trước (luồng qua ranh giới), rồi đào sâu từng phần tử.

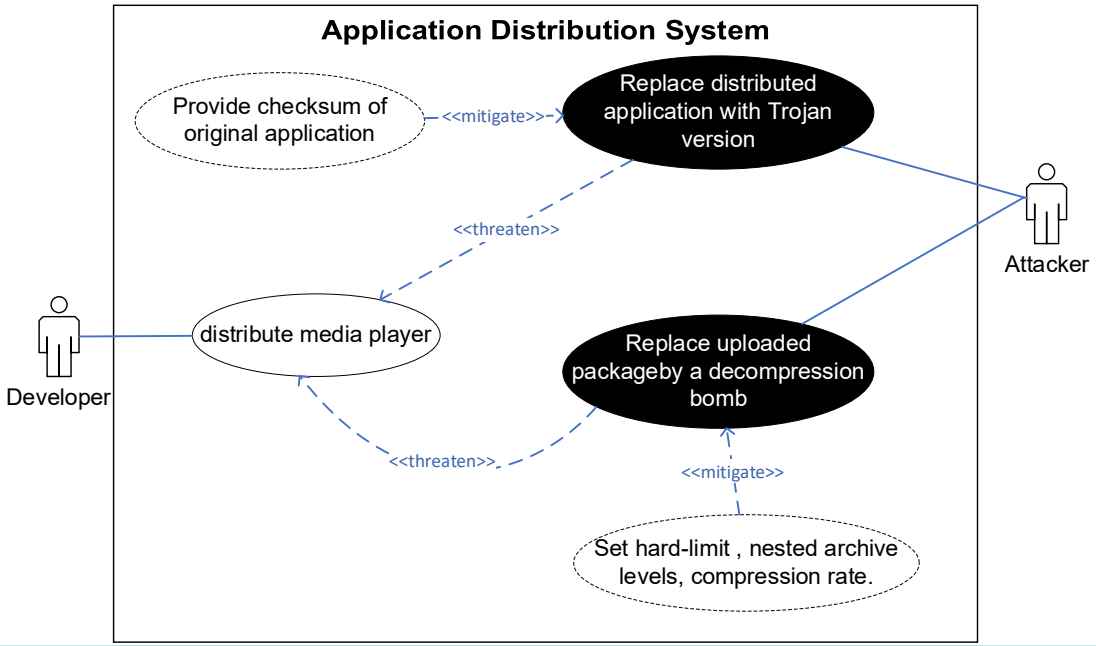
Misuse/Abuse Case - Concept

DEFINITION & WHY

- **Misuse/Abuse case** = intentional **misuse** of a system flow to cause harm or gain advantage.
  Tình huống lạm dụng/tấn công = cố ý khai thác luồng hệ thống để gây hại hoặc trục lợi.
- It flips the builder's view to the **attacker's perspective**; complements use cases to reveal threats early.
  Đảo góc nhìn người xây dựng sang **kẻ tấn công**; bổ sung cho use case để lộ đe dọa sớm.
- **How to write**
  - ✓ **Reverse user story**: "*As a malicious actor*, I want to *{action}* so that *{harm/benefit}*."
    User story đảo ngược: "*Với tư cách kẻ xấu*, tôi muốn *{hành động}* để *{gây hại/được lợi}*."
  - ✓ Include **scope & assets** (PII/keys/finance), **entry point**, **boundary crossing**, and **preconditions**.
    Ghi rõ phạm vi & tài sản, điểm vào, ranh giới, và điều kiện tiên quyết.
  - ✓ Add **detection criteria** (signals/thresholds), **owner**, and **mitigations** mapped to STRIDE.
    Thêm tiêu chí phát hiện (tín hiệu/ngưỡng), owner, và biện pháp gắn với STRIDE.
- **Examples**
  - ✓ Brute-force login (S/D): "As an attacker, I try many passwords to hijack accounts and lock capacity."
    Đoán mật khẩu (S/D): "Là kẻ tấn công, tôi thử nhiều mật khẩu để chiếm tài khoản và làm cạn tài nguyên."
  - ✓ IDOR/BOLA (E/I): "I change /orders/123 to view/modify others' orders."
    Truy cập sai đối tượng (E/I): "Tôi đổi /orders/123 để xem/sửa đơn của người khác."
  - ✓ Error leak (I/R): "I trigger verbose errors to exfiltrate secrets without leaving evidence."
    Rò rỉ qua lỗi (I/R): "Tôi gây lỗi chi tiết để hút bí mật mà khó bị truy vết."

VISUAL – USECASE VS MISUSECASE

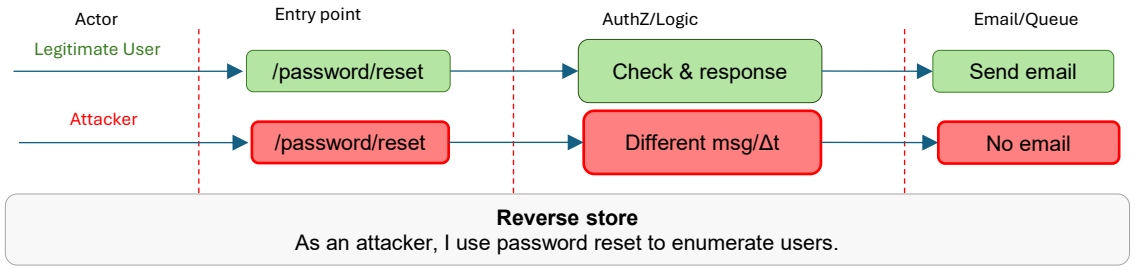




Tips: refer test-guide documents (eg: owasp top ten,…) to think like an attacker.
Mẹo: tham khảo tài liệu hướng dẫn kiểm tra (ví dụ: owasp top ten,…) để suy nghĩ như một kẻ tấn công.

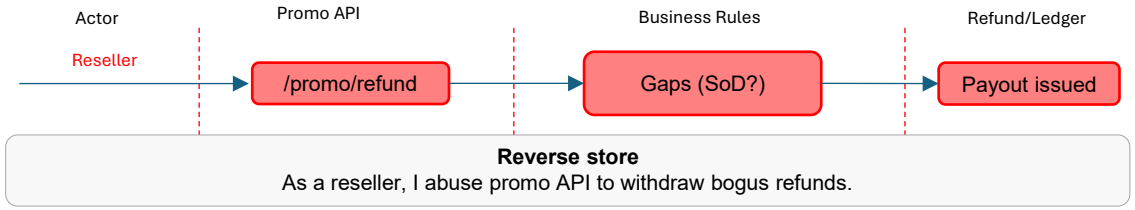
Misuse/Abuse Case Description

MISUSE CASE – ENUMERATE USERS



Field	Content
Actor	Anonymous attacker
Preconditions	Endpoint reveals existence via **distinct messages**, **timing (Δt)**, or **HTTP codes**
Steps	Probe emails/usernames → observe responses/timing
Entry & Boundary	POST /password/reset across B1/B2
Assets	User privacy list, account takeover
STRIDE	**I** (primary), **R/D** (secondary)
Detection	Pattern of many reset attempts per /24; Δt variance; unusual user-agent
Mitigations	Uniform messages/timing; per-user/IP budgets; CAPTCHA; email-only response; rate limit; anomaly alerts.
Owner	Identity/Frontend teams

MISUSE CASE – WITHDRAW BOGUS REFUNDS



Field	Content
Actor	External reseller with valid API key
Preconditions	Promo API lacks eligibility check, per-merchant quotas, idempotency, or SoD (separation of duties)
Steps	Submit crafted refund → repeat/modify amount → replay
Entry & Boundary	POST /promo/refund across B2→B3
Assets	Funds/ledger integrity, promo budget
STRIDE	**E**, **T**, **R** (logging) • I (secondary)
Detection	Anomalies: burst refunds, same card/user across merchants, high variance amounts.
Mitigations	AuthZ per action/resource; eligibility & anti-abuse rules; per-merchant quotas; idempotency keys; 2-person approval (SoD); ledger invariants; audit.
Owner	Payments/Promo teams

Attach each abuse case to an entry point and a boundary crossing; reference the owner and plan mitigations in Q3.

Gắn mỗi tình huống vào điểm vào và ranh giới; chỉ định phụ trách và lên kế hoạch biện pháp ở Q3.

Keep detection criteria actionable with thresholds and dashboards; capture evidence for SDR.

Giữ tiêu chí phát hiện có thể hành động kèm ngưỡng & bảng theo dõi; lưu bằng chứng cho SDR.

Part 3
Lab work with Ushop Case-study

Summary

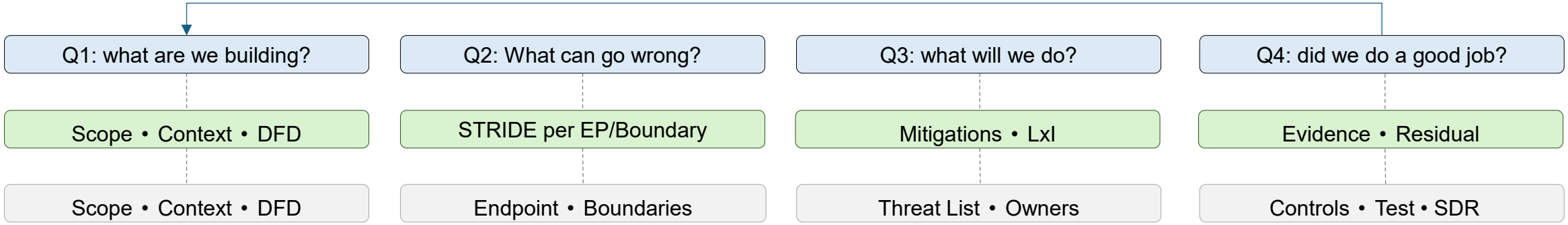
Summary - Threat Modeling

CORE IDEAS

- **Adversarial perspective**: stop thinking like builders; see **code + components + data flows** across **trust boundaries**.
  Góc nhìn đối kháng: nhìn hệ thống như mã + thành phần + luồng dữ liệu qua ranh giới tin cậy.
- **Threats are omnipresent** — manage, don't chase perfection; raise the bar iteratively.
  Đe doạ luôn hiện hữu — quản trị rủi ro theo từng bước tăng dần, không cầu toàn.
- **Focus on assets & attack surface**: prioritize high-value data/services/privileges; list entry points & interfaces.

Tập trung tài sản & bề mặt tấn công: ưu tiên dữ liệu/dịch vụ/quyền giá trị cao; liệt kê điểm vào & giao diện.

- **Four Questions (Q1→Q4)**



- **STRIDE for Threat Classification**
  - ✓ **Per entry point**: ask 6 questions (S/T/R/I/D/E).
    Mỗi điểm vào: hỏi đủ 6 câu STRIDE.
  - ✓ **Don't forget AuthZ & Auditing**: object-level checks at **time-of-use**; immutable logs for accountability.
    Đừng quên Ủy quyền & Ghi log: kiểm quyền theo đối tượng tại thời điểm dùng; log bất biến để truy cứu.
  - ✓ **Misuse/Abuse cases**: write reverse stories + detection criteria; map to STRIDE & owners.
    Tình huống lạm dụng: story đảo + tiêu chí phát hiện; ánh xạ STRIDE & gán phụ trách.

- **STRIDE ↔ CIA/AAA**

STRIDE	Primary	Also relates
S — Spoofing	AuthN	AuthZ, Audit
T — Tampering	Integrity	Audit
R — Repudiation	Auditability	Integrity
I — Info Disclosure	Confidentiality	Privacy, Metadata
D — DoS	Availability	Resilience
E — Elevation	Authorization	Least Privilege