OPEN-SOURCE SOFTWARE POLICY

CHÍNH SÁCH SỬ DỤNG PHẦN MỀM MÃ NGUỒN MỞ

Know Your License • Scan Everything • Attribute Correctly

Tổ chức / Organization	[Tên tổ chức]
Chủ sở hữu / Document Owner	Head of Engineering (phối hợp Pháp chế / Legal cho các vấn đề giấy phép)
Phê duyệt / Approved by	Engineering Governance Council
Ngày hiệu lực / Effective Date	[DD/MM/YYYY]
Phiên bản / Version	1.0
Chu kỳ rà soát / Review Cycle	Hằng năm, hoặc ngay khi có thay đổi đáng kể về khung pháp lý giấy phép hoặc chuẩn SBOM được ngành áp dụng rộng rãi
Tài liệu cấp trên / Parent Documents	Engineering Constitution v1.0 (Tầng 1); Software Development Governance Policy v1.0 (Tầng 2)
Phân loại / Classification	INTERNAL

SAMPLE DOCUMENT

A. Kiểm soát tài liệu

DOCUMENT CONTROL

Phiên bản	Ngày	Người soạn	Mô tả thay đổi
1.0	[DD/MM/YYYY]	[Tên / Vai trò]	Ban hành lần đầu

B. Mục lục nội dung

CONTENTS

- 1. Mục đích
- 2. Phạm vi áp dụng
- 3. Liên kết với Engineering Constitution
- 4. Định nghĩa
- 5. Yêu cầu bắt buộc
- 6. Vai trò và trách nhiệm
- 7. Đo lường tuân thủ
- 8. Cơ chế ngoại lệ
- 9. Không tuân thủ
- 10. Chủ sở hữu tài liệu và chu kỳ rà soát
- 11. Tài liệu liên quan
- 12. Phê duyệt

1. Mục đích

PURPOSE

1. Cho phép tổ chức khai thác đầy đủ lợi ích của phần mềm mã nguồn mở (tốc độ phát triển, chi phí, chất lượng từ cộng đồng) trong khi kiểm soát được rủi ro pháp lý về giấy phép, rủi ro bảo mật trong chuỗi cung ứng phần mềm, và rủi ro vận hành khi phụ thuộc vào thư viện không còn được bảo trì.
2. Mở rộng chi tiết yêu cầu đã nêu ngắn gọn tại Software Development Governance Policy (SDGP), Điều 5.10 — Chuỗi cung ứng phần mềm và mã nguồn mở — thành các quy định cụ thể, khả thi để áp dụng hằng ngày trong đội ngũ kỹ thuật.
3. Bảo đảm mọi sản phẩm phần mềm của tổ chức có thể chứng minh được (traceable) thành phần mã nguồn mở mà nó sử dụng, tình trạng giấy phép và tình trạng lỗ hổng bảo mật tại bất kỳ thời điểm nào.

Chính sách này bổ sung cho SDGP, không thay thế — mọi yêu cầu chung của SDGP (SDLC, vai trò, đo lường tuân thủ, cơ chế ngoại lệ, không tuân thủ) vẫn áp dụng đầy đủ.

2. Phạm vi áp dụng

SCOPE

- Mọi thành phần phần mềm mã nguồn mở (open-source component/dependency) được đưa vào sản phẩm, dịch vụ hoặc hạ tầng nội bộ của tổ chức, dù được khai báo trực tiếp hay là dependency gián tiếp (transitive).
- Áp dụng cho toàn bộ vòng đời sử dụng: lựa chọn, đưa vào dự án, giám sát trong suốt thời gian sử dụng, đến khi gỡ bỏ.
- Áp dụng cho nhân sự nội bộ, nhà thầu và đối tác thuê ngoài phát triển phần mềm cho tổ chức.
- Bao gồm cả trường hợp đóng góp ngược (outbound contribution) — nhân viên đóng góp mã nguồn của tổ chức ra dự án mã nguồn mở công cộng.
- Không áp dụng cho phần mềm thương mại có giấy phép độc quyền (proprietary/commercial licensed) — các phần mềm đó thuộc phạm vi quản lý nhà cung cấp/bên thứ ba tại SDGP Điều 5.11.

3. Liên kết với Engineering Constitution

ALIGNMENT WITH THE CONSTITUTION

Yêu cầu trong Policy	Nguyên tắc Constitution tương ứng
5.1 Phân loại giấy phép mã nguồn mở	Điều 2 — Security and Privacy by Design (rủi ro pháp lý cần được thiết kế kiểm soát từ đầu)
5.2 Quét thành phần tự động (SCA) trong CI/CD	Điều 5, Điều 2 — Everything as Code / Security by Design
5.3 SBOM cho mỗi bản phát hành	Điều 6, Điều 8 — Observable and Operable / Evidence over Opinion
5.4 Vá lỗ hổng dependency theo SLA	Điều 2 — Security and Privacy by Design

5.5 Kiểm soát thư viện không rõ nguồn gốc/rủi ro cao	Điều 2, Điều 9 — Security by Design / Reuse có trách nhiệm
5.6 Đóng góp ngược có phê duyệt	Điều 4 — Segregation of Duties
5.7 Attribution và tuân thủ giấy phép khi phân phối	Điều 8 — Evidence over Opinion

4. Định nghĩa

DEFINITIONS

- **Phần mềm mã nguồn mở (Open-Source Software):** phần mềm được phân phối kèm mã nguồn theo một giấy phép cho phép sử dụng, sao chép, sửa đổi và phân phối lại, tuân theo các điều kiện do giấy phép đó quy định.
- **Giấy phép permissive:** nhóm giấy phép mã nguồn mở cho phép sử dụng, sửa đổi, phân phối lại — kể cả trong sản phẩm đóng nguồn/thương mại — với rất ít ràng buộc, thường chỉ yêu cầu giữ lại thông báo bản quyền và văn bản giấy phép gốc (ví dụ MIT, Apache License 2.0, BSD).
- **Giấy phép copyleft:** nhóm giấy phép mã nguồn mở yêu cầu tác phẩm phái sinh (derivative work) khi phân phối ra ngoài cũng phải được cấp phép lại theo cùng điều khoản và công bố mã nguồn tương ứng ("chia sẻ ngược"); mức độ ràng buộc khác nhau theo từng giấy phép cụ thể (ví dụ GPL, LGPL, AGPL).
- **Copyleft mạng (Network Copyleft):** đặc điểm của một số giấy phép copyleft — điển hình là AGPL — trong đó nghĩa vụ công bố mã nguồn được kích hoạt ngay cả khi phần mềm chỉ được cung cấp dưới dạng dịch vụ qua mạng (ví dụ SaaS, API) mà không phân phối file nhị phân hay mã nguồn trực tiếp cho người dùng.
- **Dependency trực tiếp / gián tiếp (transitive):** dependency trực tiếp là thư viện được khai báo trực tiếp trong dự án; dependency gián tiếp là thư viện mà các dependency trực tiếp đó phụ thuộc vào, thường không được đội ngũ phát triển chủ động lựa chọn hay nhận biết đầy đủ.
- **SCA (Software Composition Analysis):** kỹ thuật/công cụ quét tự động để phát hiện thành phần mã nguồn mở, giấy phép và lỗ hổng bảo mật đã biết trong một dự án phần mềm, bao gồm cả dependency gián tiếp.
- **SBOM (Software Bill of Materials):** danh mục có cấu trúc, máy đọc được, liệt kê đầy đủ các thành phần phần mềm — bao gồm mã nguồn mở — cấu thành nên một sản phẩm phần mềm tại một phiên bản/thời điểm phát hành cụ thể.
- **VEX (Vulnerability Exploitability eXchange):** tài liệu đi kèm SBOM, công bố tình trạng một lỗ hổng đã biết trong một thành phần có thực sự có thể bị khai thác trong bối cảnh sử dụng cụ thể của sản phẩm hay không (ví dụ "Not Affected"), giúp giảm số lượng cảnh báo cần xử lý thủ công.
- **Đóng góp ngược (Outbound Contribution):** hành vi gửi mã nguồn, bản vá hoặc tài liệu kỹ thuật do nhân viên tạo ra trong quá trình làm việc cho tổ chức tới một dự án mã nguồn mở công cộng bên ngoài.

5. Yêu cầu bắt buộc

MANDATORY REQUIREMENTS

5.1 Phân loại giấy phép mã nguồn mở

1. Trước khi đưa một thành phần mã nguồn mở vào sản phẩm, đội ngũ phát triển **PHẢI (MUST)** xác định giấy phép của thành phần đó và phân loại vào một trong hai nhóm: permissive (ví dụ MIT, Apache-2.0, BSD) hoặc copyleft (ví dụ GPL, LGPL, AGPL và các biến thể tương tự).
2. Thành phần mang giấy phép permissive **ĐƯỢC PHÉP (MAY)** sử dụng tự do hơn trong sản phẩm thương mại/đóng nguồn, miễn tuân thủ nghĩa vụ attribution tại Điều 5.7.
3. Thành phần mang giấy phép copyleft, đặc biệt AGPL và GPL, **PHẢI (MUST)** được rà soát bởi Pháp chế (Legal) trước khi đưa vào bất kỳ sản phẩm thương mại nào, do nghĩa vụ "chia sẻ ngược" mã nguồn có thể ảnh hưởng trực tiếp đến mô hình kinh doanh và tài sản trí tuệ của tổ chức. Việc rà soát áp dụng bất kể hình thức liên kết (tĩnh, động) hay hình thức cung cấp dưới dạng dịch vụ qua mạng (xem khái niệm copyleft mạng tại Điều 4) — Legal quyết định phạm vi ảnh hưởng cụ thể theo từng giấy phép.
4. Giấy phép chưa được phân loại rõ ràng (không rõ nguồn gốc, giấy phép tùy chỉnh không theo danh sách được OSI công nhận) **PHẢI (MUST)** được xử lý như copyleft cho đến khi Legal xác nhận khác.
5. Khi một sản phẩm kết hợp nhiều thành phần mã nguồn mở có giấy phép khác nhau, đội ngũ phát triển **PHẢI (MUST)** đối chiếu với ma trận tương thích giấy phép (License Compatibility Matrix) do Legal ban hành trước khi phát hành, để bảo đảm việc liên kết các thành phần với nhau không vi phạm điều khoản của bất kỳ giấy phép nào trong số đó.

5.2 Quét thành phần tự động (SCA)

1. Mọi dự án phần mềm **PHẢI (MUST)** tích hợp công cụ quét thành phần tự động (Software Composition Analysis - SCA) vào pipeline CI/CD, quét toàn bộ dependency kể cả dependency gián tiếp (transitive).
2. Việc quét **PHẢI (MUST)** được thực hiện tối thiểu tại thời điểm build và định kỳ theo lịch để phát hiện lỗ hổng mới công bố trên thành phần đã có sẵn trong dự án, không chỉ tại thời điểm thêm dependency mới.
3. Tại các nhánh phát triển thông thường, kết quả quét phát hiện vấn đề mới **NÊN (SHOULD)** chỉ tạo cảnh báo (warn) để không cản trở công việc hằng ngày. Tại cổng phát hành (release gate) trước khi lên production, pipeline **PHẢI (MUST)** chặn (block) khi phát hiện: giấy phép copyleft chưa qua rà soát Legal theo Điều 5.1, hoặc lỗ hổng ở mức nghiêm trọng theo Điều 5.4 — cho đến khi được xử lý hoặc có ngoại lệ hợp lệ theo Điều 8.
4. Security/AppSec **NÊN (SHOULD)** duy trì danh sách giấy phép/thư viện đã được phê duyệt trước (allowlist) để tự động thông qua các trường hợp phổ biến, giảm khối lượng rà soát thủ công lặp lại cho Legal.

5.3 SBOM (Software Bill of Materials)

1. Mỗi bản phát hành sản phẩm ra production **PHẢI (MUST)** đi kèm một SBOM được tạo tự động, phản ánh đúng thành phần mã nguồn mở thực tế tại phiên bản đó.
2. SBOM **PHẢI (MUST)** được tạo theo một định dạng chuẩn được công nhận rộng rãi trong ngành — SPDX (dự án của Linux Foundation) hoặc CycloneDX (dự án của OWASP) — để bảo đảm khả năng trao đổi, đối chiếu tự động với đối tác khi cần.

3. Nội dung tối thiểu của SBOM **PHẢI (MUST)** tuân theo hướng dẫn hiện hành của cơ quan quản lý được công nhận rộng rãi (hiện là CISA, kế thừa và thay thế hướng dẫn ban đầu của NTIA năm 2021), bao gồm tối thiểu: tên và phiên bản thành phần, nhà cung cấp/tác giả, mã băm (hash) thành phần, giấy phép, quan hệ phụ thuộc, cùng thông tin về công cụ và bối cảnh đã dùng để tạo ra SBOM đó.
4. Đối với hệ thống được phân loại rủi ro cao hoặc phân phối cho khách hàng bên ngoài, tổ chức **NÊN (SHOULD)** đi kèm SBOM một tài liệu VEX (theo Điều 4) để công bố rõ lỗ hổng nào trong thành phần thực sự ảnh hưởng đến sản phẩm, giảm số lượng cảnh báo cần xử lý thủ công không cần thiết.
5. SBOM **PHẢI (MUST)** được lưu trữ có phiên bản (versioned) cùng bản phát hành tương ứng, phục vụ tra cứu khi có lỗ hổng mới được công bố ảnh hưởng đến thành phần đã phát hành trước đó, và **PHẢI (MUST)** có thể cung cấp cho khách hàng/đối tác khi hợp đồng hoặc quy định pháp luật hiện hành yêu cầu.

5.4 Vá lỗ hổng dependency theo SLA

1. Lỗ hổng bảo mật được phát hiện trong dependency mã nguồn mở **PHẢI (MUST)** được xử lý theo cùng hạn mức thời gian (SLA) phân loại theo mức độ nghiêm trọng đã quy định tại Secure Software Development Policy, không có SLA riêng biệt chỉ vì lỗ hổng nằm ở thành phần bên ngoài thay vì mã tự viết.
2. Việc vá lỗ hổng **PHẢI (MUST)** được ưu tiên theo mức độ nghiêm trọng và khả năng khai thác thực tế trong bối cảnh triển khai của tổ chức (tham khảo VEX nếu có tại Điều 5.3.4), không chỉ theo điểm số kỹ thuật chung chung của lỗ hổng.
3. Trường hợp chưa có bản vá chính thức từ dự án nguồn mở, đội ngũ phụ trách **PHẢI (MUST)** áp dụng biện pháp giảm thiểu tạm thời (patch không chính thức, cấu hình chặn đường tấn công, cô lập thành phần) và ghi nhận rủi ro còn tồn đọng.
4. Với sản phẩm thuộc phạm vi các quy định về báo cáo lỗ hổng/sự cố an ninh mạng có hiệu lực bắt buộc tại thị trường đang phục vụ, đội phụ trách bảo mật **PHẢI (MUST)** tuân thủ thời hạn báo cáo bắt buộc theo quy định đó đối với lỗ hổng dependency đang bị khai thác trong thực tế, song song với SLA vá lỗi nội bộ nêu trên.

5.5 Thư viện không rõ nguồn gốc, không còn bảo trì hoặc rủi ro cao

1. **KHÔNG ĐƯỢC PHÉP (MUST NOT)** đưa vào sản phẩm thư viện mã nguồn mở không rõ nguồn gốc — không xác định được tác giả/tổ chức phát hành, không có kho mã nguồn công khai có thể kiểm chứng.
2. Thư viện đã được đánh dấu archived/deprecated bởi chính dự án, hoặc không có bản phát hành mới trong thời gian dài bất thường so với vòng đời thông thường của loại thư viện đó, **PHẢI (MUST)** qua rà soát đặc biệt trước khi sử dụng hoặc tiếp tục sử dụng — ưu tiên tìm giải pháp thay thế đang được bảo trì tích cực.
3. Thư viện có lịch sử phát hành mỏng, số lượng người dùng/tải về thấp bất thường, hoặc không có dấu hiệu được cộng đồng/tổ chức uy tín kiểm chứng, **PHẢI (MUST)** qua rà soát bổ sung của đội bảo mật trước khi dùng trong hệ thống được phân loại rủi ro cao theo SDGP.
4. Yêu cầu xác minh dependency nêu tại Điều này áp dụng đồng thời với yêu cầu tại AI-Assisted Software Development Policy, Điều 5.5 — dependency do AI đề xuất càng cần được xác minh là có thật và đáp ứng các tiêu chí trên trước khi cài đặt, nhằm phòng tránh rủi ro "ảo giác gói phần mềm" (slopsquatting).

5.6 Chính sách đóng góp ngược (Outbound Contribution)

1. Nhân viên muốn đóng góp mã nguồn, bản vá hoặc tài liệu kỹ thuật liên quan đến công việc tại tổ chức ra một dự án mã nguồn mở công cộng **PHẢI (MUST)** được phê duyệt trước bởi quản lý trực tiếp và bộ phận được ủy quyền (Legal hoặc Head of Engineering), nhằm bảo đảm không rò rỉ mã nguồn nội bộ, bí mật kinh doanh hoặc thông tin nhạy cảm khác.
2. Việc phê duyệt **PHẢI (MUST)** xem xét: nội dung đóng góp có chứa thông tin độc quyền của tổ chức hay không, và giấy phép của dự án nhận đóng góp có phù hợp với chiến lược tài sản trí tuệ của tổ chức hay không.
3. Đóng góp ngược đã được phê duyệt **NÊN (SHOULD)** được ghi nhận tập trung để tổ chức theo dõi được các đóng góp công khai mang tên hoặc liên quan đến tổ chức.

5.7 Attribution và tuân thủ giấy phép khi phân phối

1. Khi phân phối sản phẩm có chứa thành phần mã nguồn mở ra bên ngoài tổ chức (khách hàng, đối tác, công chúng), đội ngũ phát triển **PHẢI (MUST)** đính kèm đầy đủ thông báo bản quyền, văn bản giấy phép gốc và attribution theo đúng yêu cầu của từng giấy phép tương ứng.
2. Việc tổng hợp danh sách attribution **NÊN (SHOULD)** được tự động hóa từ dữ liệu SCA/SBOM tại Điều 5.2, 5.3 thay vì tổng hợp thủ công, để giảm rủi ro sai sót hoặc thiếu sót khi số lượng dependency lớn.
3. Việc không tuân thủ nghĩa vụ attribution là vi phạm nghĩa vụ pháp lý của giấy phép gốc, không chỉ là vấn đề nội bộ. Trường hợp phát hiện thiếu sót hoặc vi phạm giấy phép sau khi sản phẩm đã được phân phối, đội phụ trách **PHẢI (MUST)** phối hợp Legal khắc phục khẩn cấp — có thể bao gồm cập nhật attribution, phát hành lại bản vá, hoặc thay thế thành phần vi phạm — và báo cáo theo cơ chế tại Điều 9.

> **Không lặp lại nội dung của SDGP:** SDLC, vai trò/trách nhiệm chung, đo lường tuân thủ chung, cơ chế ngoại lệ và không tuân thủ đã được quy định tại SDGP. Policy này chỉ mở rộng phần chuỗi cung ứng phần mềm/mã nguồn mở (SDGP Điều 5.10); khi có xung đột diễn giải với phần chung, SDGP được ưu tiên áp dụng.

6. Vai trò và trách nhiệm

ROLES & RESPONSIBILITIES

- **Head of Engineering (Document Owner):** ban hành, cập nhật Policy; quyết định cuối cùng về ngoại lệ khi không có sự đồng thuận giữa Engineering và Legal.
- **Pháp chế (Legal):** rà soát và quyết định việc sử dụng giấy phép copyleft trong sản phẩm thương mại (Điều 5.1); ban hành và cập nhật ma trận tương thích giấy phép (Điều 5.1.5); phê duyệt đóng góp ngược cùng quản lý trực tiếp (Điều 5.6); xử lý sự cố tuân thủ giấy phép khi phân phối (Điều 5.7).
- **Security/AppSec:** vận hành công cụ SCA và duy trì allowlist (Điều 5.2); xác định mức độ nghiêm trọng và SLA vá lỗ hổng dependency, theo dõi nghĩa vụ báo cáo bắt buộc theo thị trường (Điều 5.4); rà soát bổ sung thư viện rủi ro cao cho hệ thống rủi ro cao (Điều 5.5).
- **Engineering Teams:** lựa chọn, cập nhật, thay thế dependency; bảo đảm SBOM chính xác cho bản phát hành mình phụ trách; tuân thủ quy trình phê duyệt trước khi đóng góp ngược.

7. Đo lường tuân thủ

COMPLIANCE MEASUREMENT

- Tỷ lệ dự án có tích hợp SCA trong CI/CD trên tổng số dự án đang hoạt động.
- Tỷ lệ bản phát hành production có SBOM hợp lệ đi kèm, đạt yêu cầu nội dung tối thiểu tại Điều 5.3.3.
- Thời gian trung bình vá lỗ hổng dependency theo từng mức độ nghiêm trọng, đối chiếu với SLA đã cam kết.
- Số lượng thành phần copyleft chưa qua rà soát Legal được phát hiện trong sản phẩm đang chạy production.
- Số lượng đóng góp ngược được phê duyệt đúng quy trình so với số phát hiện ngoài quy trình (nếu có).

8. Cơ chế ngoại lệ

EXCEPTIONS

Áp dụng theo đúng cơ chế và giới hạn tại Engineering Constitution, Điều 8 và SDGP, Điều 8. Ngoại lệ liên quan đến giấy phép copyleft trong sản phẩm thương mại bắt buộc có ý kiến bằng văn bản của Legal trước khi phê duyệt.

9. Không tuân thủ

NON-COMPLIANCE

Áp dụng theo đúng nguyên tắc tại SDGP, Điều 9: báo cáo và khắc phục cho vi phạm thông thường; chỉ xem xét kỷ luật lao động — qua HR/Pháp chế, theo Nội quy lao động — đối với hành vi cố ý (ví dụ: cố tình đưa giấy phép copyleft vào sản phẩm thương mại mà không qua rà soát Legal, hoặc đóng góp ngược mã nguồn nội bộ ra ngoài mà không xin phê duyệt).

10. Chủ sở hữu tài liệu và chu kỳ rà soát

DOCUMENT OWNERSHIP & REVIEW CYCLE

Document Owner: Head of Engineering, phối hợp Pháp chế (Legal) cho các vấn đề giấy phép. Chu kỳ rà soát: hằng năm, hoặc ngay khi có thay đổi đáng kể về khung pháp lý giấy phép mã nguồn mở hoặc chuẩn SBOM được ngành áp dụng rộng rãi.

11. Tài liệu liên quan

RELATED DOCUMENTS

- Engineering Constitution v1.0, Software Development Governance Policy v1.0 (Tầng 1-2)
- Secure Software Development Policy, AI-Assisted Software Development Policy (Tầng 2)
- Vulnerability Management Standard, Secure Coding Standard (Tầng 3)
- Danh mục thư viện mã nguồn mở đã phê duyệt, Ma trận tương thích giấy phép, Quy trình phê duyệt đóng góp ngược (Tầng 4/5)

Mọi dòng mã mã nguồn mở đi kèm một giấy phép, và mọi giấy phép đi kèm một nghĩa vụ;
không dependency nào — kể cả gián tiếp — được phép là ẩn số trong sản phẩm của chúng ta;
mọi bản phát hành đều biết rõ nó được tạo thành từ những gì;
và không ai đóng góp mã của tổ chức ra thế giới bên ngoài mà không có sự chấp thuận.

12. Phê duyệt

APPROVAL

Vai trò	Họ tên	Chữ ký	Ngày
Chair, Engineering Governance Council			
Chief Technology Officer			
Head of Engineering (Document Owner)			
Head of Legal / Pháp chế			