OPENCHAIN 課程

自由開源軟體訓練課程簡報–搭配 OpenChain 規範書 1.1 版

本簡報依美國法律進行說明。不同的司法管轄區域可能會有不同的法律要求。
此點在使用簡報作爲合規訓練專案的一部分時，應被考量。

本簡報內容並未包含法律建議

OpenChain 課程是什麼?

- OpenChain項目，協助確認及分享自由開源軟體合規專案的核心構成要素。
- OpenChain項目的核心爲其**規範書**。其確認並發布一個自由開源軟體合規專案，所應滿足的核心要項。
- 本OpenChain**課程**，透過提供自由可取得的訓練素材，來支持規範書。
- 本簡報協助商業公司滿足規範書 1.2 項的要求。其亦可被用於一般合規訓練上。

取得更多資訊：https://www.openchainproject.org

內容

1. 什麼是智慧財產？
2. 介紹自由開源軟體授權
3. 介紹自由開源軟體合規
4. 自由開源軟體審核的關鍵軟體觀念
5. 進行自由開源軟體審核
6. 端對端的合規管理（流程範例）
7. 避開合規陷阱
8. 開發者準則

自由開源軟體政策

- <<此待補充的空白項，用以指示何處可以找到自由開源軟體政策書(依OpenChain規範書 1.1 版第1.1.1項要求)>>

- 你可透過Linux Foundation開源合規專案取得一份自由開源軟體政策書的範本 https://www.linux.com/publications/generic-foss-policy

章節一

什麼是智慧財產？

什麼是「智慧財產」？

- 著作權：保護作者具創作性之原始著作
  - 保護表達(不及於其後之思想)
  - 涵蓋軟體、書籍，及其他相類作品
- 專利：具新穎性及非屬顯著已知狀態的實用發明
  - 抑制壟斷以鼓勵創新
- 營業秘密：保護具價值的保密資訊
- 商標：保護用來辨識產品來源的標章（文字、圖形記號、標語、顏色，等等。）
  - 保護消費者與品牌；避免消費者混淆與品牌淡化

本章節將聚焦在著作權與專利，

該領域與自由開源軟體合規最為相關。

軟體中的著作權概念

- 基本規則：著作權保護具創作性的作品
- 著作權一般適用於文學作品，例如書籍、電影、圖片、音樂、地圖
- 軟體受到著作權保護
  - 軟體被著作權保護的部分並非功能（這部份是被專利保護的）而是表達（實作細節中的創作性）
  - 包括二進位執行碼及源碼皆受到保護
- 著作權利人只對他/她創作的作品有控制地位，不及於他人的獨立創作。
- 未經作者同意的複製可能導致侵權行爲

著作權中與軟體最相關的權利態樣

- 重製軟體的權利 - 製作副本
- 創作「衍生著作」的權利 - 進行修改
  - 此處衍生著作一詞引用自美國著作權法
  - 其爲「專有名詞」，意指依法令的特定涵義來解釋，而非依一般字典定義。
  - 一般來說，其指的是基於一個原著作來創作的新作品，過程中有足夠的創作性加入，而讓新作品表現爲另一具創作性的作品，而非僅爲原著作的副本。
- 散布的權利
  - 散布一般被視爲，在二進位執行檔或是源碼的形式下，提供軟體一部分的副本給其他的實體。（在你公司或是組織外的個人或是組織）

注意：對何者構成「衍生著作」或「散布」的解釋，在自由開源軟體社群與自由開源軟體法律圈中仍有爭議

軟體中的專利概念

- 專利保護功能性 - 這可以包含操作的方法，例如電腦程式
  - 不保護抽象思想、自然法則
- 爲於特定地區取得專利，必須在該特定的司法管轄領域提出專利申請。倘該專利被核可，專利擁有者有權利可以阻止他人實施該專利的功能，不論該功能是否爲獨立創作。
- 其他希望利用該項技術者，或會洽詢專利授權（該授權可能授與使用、製造、使製造、銷售、提供銷售，及輸入該技術的權利）
- 即使其他人獨立地創作相同的發明，仍可能導致侵權行爲

授權

- 「授權」是著作權或專利擁有者，授與同意或權利給其他人的方法
- 授權可以被限定在：
  - 被允許的使用類型（商業性/非商業性、散布、衍生著作/製造、使製造、生產）
  - 專屬或非專屬的條件
  - 地理範圍
  - 永久的或限定期間
- 授權可以是附條件的授與，意指僅有在遵循某些義務性要求才會獲得授權
  - 例如，提供姓名標示，或給予互惠性授權
- 可能包含與保證、賠償、支援、升級、維護相關的契約條件

檢測你的了解程度

OPENCHAIN

- 著作權法保護何種素材？
- 對軟體最重要的著作權權利是什麼？
- 軟體能夠是專利的標的嗎？
- 專利給予專利擁有者什麼權利？
- 如果你獨立開發了你的軟體，你有可能會需要第三方的著作權授權或是專利授權嗎？

章節二

介紹自由開源軟體授權

自由開源軟體授權

- 自由開源軟體授權依定義來說，是讓源碼能被容許修改及再次散布的條款
- 自由開源軟體授權，可能會附隨姓名標示、保留著作權聲明，或是提供書面源碼索取文件有關的條件
- 一組較普及的授權是由開放原始碼促進會(OSI)依據其開放源碼定義(OSD)核準的列表。完整的OSI核準授權列表可參照右列連結

  http://www.opensource.org/licenses/

寬鬆式的自由開源軟體授權

- 寬鬆式的自由開源軟體授權 – 此一詞彙用來描繪最低限制程度的自由開源軟體授權
- 例如：BSD-3-Clause
  - 3款BSD授權條款是寬鬆式自由開源軟體授權的一則著例，其允許源碼或目的碼形式不受限制，基於任何目的之再行發布，只要其著作權聲明以及授權條款裡的免責聲明有被維持
  - 該授權當中有一個條款，限制了在衍生著作中，若未得到特別允許的話，不得使用貢獻者的名字背書
- 其他例子：MIT、Apache-2.0

授權互惠性 & Copyleft 授權條款

- 某些授權條款要求當衍生著作（或相同檔案的軟體、同一個軟體程式，或依其他界線劃分的範圍）被散布時，必須以原著作相同的條款進行散布
- 此被稱爲 “copyleft” 或、「授權互惠性」的效應
- 以GPL-2.0的授權互惠性爲例：

  你必須讓任何你散布或發布的作品，其全部或一部含有 GPL-2.0 原生程式，或爲 GPL-2.0 程式之衍生，或前述原生衍生的任何一部分，採，…，本授權之條款來進行散布或發布。

- 帶有互惠性或稱Copyleft條文的授權條款，包括所有版本的GPL、LGPL、AGPL、MPL，以及CDDL。

私有授權或閉源軟體

- 私有軟體授權（或稱商業授權、或稱終端使用者授權協議），對軟體的使用、修改，及／或散布具有限制性條件
- 私有授權對每一個提供者都是獨特的–有幾個提供者就有幾種私有授權的變異，故每個私有授權都應該被個別進行評估
- 即使自由開源軟體及私有授權都是基於智慧財產，以給予該財產的授權允許，然自由開源軟體的開發者常使用「私有(proprietary)」這個字詞，來形容商業性的非自由開源軟體授權。

其他非自由開源軟體的授權情境

- 免費軟體(Freeware)－採私有授權以免費或非常低價進行散布的軟體
  - 源碼可能或不能提供，創作衍生作品通常是被限制的
  - 免費軟體通常會提供完整功能（沒有被上鎖的特別功能），且可以永久使用（不受使用天數限制）
  - 免費軟體授權通常會對重製、散布和製作衍生作品，以及使用目的（個人、商業用、學術用，...等）施加限制。
- 共享軟體(Shareware)－就試用基礎提供給使用者的私有軟體，限定期間、免費但限制功能或限制特別功能
  - 共享軟體的目的是提供潛在購買者使用此軟體的機會，並在付費購買授權或完整版前先評估其實用性
  - 多數公司對共享軟體持懷疑態度，因爲共享軟體供應商，常會在共享軟體免費流傳在組織內部後，向這些公司要求索取高額的授權費用。

其他非自由開源軟體的授權情境

- 「非商業性」–某些授權具有多數自由開源軟體授權的特性，但卻限制僅供非商業使用(例如，CC 姓名標示-非商業性 授權條款 / CC BY-NC)
  - 自由開源軟體依定義，便不能限制軟體的使用範疇
  - 商業使用即爲一種使用範疇，故對商業的任何限制即阻卻該授權成爲自由開源軟體授權

公眾領域

- **公眾領域**一詞是指不被法律保護的軟體，因此公眾不需取得授權即可使用
- 開發者或會在他們的軟體附上公眾領域宣告
  - 例如，「在此軟體中的所有程式碼及文件，已被作者貢獻至公眾領域。」
  - 公眾領域宣告不等於自由開源軟體授權
- 公眾領域宣告讓開發者得嘗試放棄或消除軟體中，任何或有的智慧財產權，以明示其可不受任何限制的被使用，然宣告的可執行性於自由開源軟體社群裡仍有爭議，且其法律上的有效性亦因司法管轄區域的不同而有所差異
- 公眾領域宣告常會附帶其他條文，例如免責條款；在這種情形，此軟體通常會被視爲依授權條款提供，而非處於公眾領域

授權相容性

- 授權相容性是確保授權條款不衝突的程序
- 若有一個授權條款要求你做一件事，但另一個授權條款卻禁止你做那件事，而倘若將兩個軟體模組合併將導致其結合須置於單一授權條款的義務，那這兩個授權條款就是*互相衝突且不相容*的。
  - GPL-2.0 及 EPL-1.0 皆將其義務性規定延伸至被散布的「衍生著作」
  - 若有一 GPL-2.0 模組與一個 EPL-1.0 模組被結合在一起，此合併的模組也被散布了，那麼該模組必須：
    - (依照 GPL-2.0) 僅依 GPL-2.0 來被散布，並且
    - (依照 EPL-1.0) 僅依 EPL-1.0 來被散布。
    - 散布者無法同時滿足上列兩個條件，故此模組也許不能被散布。
    - 此爲一個*授權不相容(license incompatibility)*的例子。

「衍生著作」的定義在自由開源軟體社群中有不同看法，且其法律釋義亦隨司法管轄區域不同而有可能變化。

聲明

聲明，例如檔案標頭上的註解文字，通常會提供作者及授權資訊。自由開源軟體授權條款也可能會要求在源碼或文件裡，或併隨源碼或文件放置聲明，以表彰作者(姓名標示)，或清楚指示該軟體包括修改部件。

- **著作權聲明 –** 置於作品副本裡，告訴世界其著作權歸屬狀態的標示。例如： Copyright © A. Person (2016)
- **授權聲明 –** 說明及顯示產品中自由開源軟體的授權條款與條件的聲明。
- **姓名標示聲明 –**於產品釋出時，顯示產品中自由開源軟體的原始作者及／或其贊助者的聲明。
- **修改聲明 –** 指出你對源碼裡哪一個檔案已作修改的聲明，例如在檔案最上方加入你的著作權聲明即屬之。

多重授權

- 多重授權指的是，將軟體散布同時置於二組或更多不同的條款與條件下進行實作。
  - 例如，若軟體是採「雙重授權」，則著作權利人是將二組授權條款的選擇權交給每一個軟體的收受者
- 注意：此不應與授權人要求你必須遵從多於一組的所有授權條款之狀況混淆

檢測你的了解程度

- 什麼是自由開源軟體授權？
- 寬鬆式的自由開源軟體授權，典型的義務性要求有哪些？
- 試列出一些寬鬆式的自由開源軟體授權條款。
- 授權互惠性意指什麼？
- 試列出一些 copyleft 類型的授權條款。
- 使用依 copyleft 條款授權的程式碼時，什麼是需要一併被散布的？
- 免費軟體及共享軟體是否會被視爲自由開源軟體？
- 什麼是多重授權？
- 在自由開源軟體的聲明裡你可能找到什麼資訊，以及這些聲明能被如何利用？

章節三

介紹自由開源軟體合規

自由開源軟體合規的目標

- **了解對你的義務性要求。** 你應有一套能辨識及追蹤，你的軟體現存哪些自由開源軟體元件之流程
- **滿足授權條款的義務性規定。**你的流程應要能夠處理，因你組織的商業實作而帶來的自由開源軟體授權義務性規定。

哪些合規義務性規定必須被滿足？

依據使用到的自由開源軟體授權條款，你的合規義務性規定或許包括：

- **姓名標示與聲明。** 你也許需要提供或保留著作權聲明及授權文字到源碼，及/或產品的文件，或使用者操作介面裡，好讓下游使用者得知軟體的來源，及在該授權條款下賦予他們的權利。你也許需要提供與修改紀錄有關的聲明，或授權文件的完整副本。
- **源碼的提供。**你也許需要提供該自由開源軟體本身、你所作的修改、供結合或連結的軟體，以及控制建制流程的腳本之程式源碼。
- **互惠性。** 你也許需要採與管理該自由開源軟體元件完全相同的授權條款，來維護其修改版本或衍生著作。
- **其他條款。**該自由開源軟體授權條款或會限制其著作權利人姓名或商標之使用，也許會要求修改版本使用不同的名稱來避免混淆，或在違反此要求時終止授權。

自由開源軟體合規爭議：散布

- 對外部組織散播素材
  - 應用程式被下載到使用者的機器或行動裝置
  - JavaScript、網路服務客戶端，或其他程式碼被下載到使用者的機器
- 對於某些自由開源軟體授權條款來說，透過網絡存取可視爲觸發點。
  - 某些授權條款對觸發點的定義，包含對在伺服器上運行的軟體提供存取(例如：若該軟體被修改過的話 – 所有 Affero GPL 版本皆作如此定義)，或是「使用者透過電腦網路遠端與其互動」這種情境

自由開源軟體合規爭議：修改

- 對於當前既存的程式進行變動（例如：增加、刪除檔案裡的程式碼，將元件結合在一起）
- 依某些自由開源軟體授權條款，修改也許會在散布時帶來額外的義務性要求，例如：
  - 提供修改聲明
  - 提供伴隨的程式源碼
  - 依管理自由開源軟體元件的同份授權條款來授權該修改

自由開源軟體合規專案

已於自由開源軟體合規上取得成功的組織，會建立他們自己的自由開源軟體合規專案(包含政策、流程、訓練，及和工具)，來達到下列目的：

1. 便利自由開源軟體於其產品(商業性或其他)裡的採用效率
2. 尊重自由開源軟體開發者/權利人的權利，及遵守其授權義務性規定
3. 貢獻並參與自由開源軟體社群

導入合規實作

準備好企劃流程及足夠的人力資源來應對：

- 辨識所有內部及外部軟體的出處及授權
- 在開發流程裡追踪自由開源軟體
- 進行自由開源軟體審核及辨識其授權義務性規定
- 在產品發送時實現授權義務性規定
- 監管自由開源軟體合規專案、建立政策，及合規決策
- 內部訓練

合規的好處

健全的自由開源軟體合規專案帶來的好處包括：

- 對自由開源軟體的好處及其如何對你的組織產生影響，增加認識
- 對使用自由開源軟體的成本及風險，增加認識
- 對可用的自由開源軟體方案，增加知識
- 減低及管理侵權風險、增加對自由開源軟體開發者/權利人授權決策的尊重
- 與自由開源軟體社群及組織培養良好關係

檢測你的了解程度

- 自由開源軟體合規意指什麼？
- 自由開源軟體合規專案的兩個主要目標是什麼？
- 條列並說明自由開源軟體合規專案的重要商業實作
- 自由開源軟體合規專案的好處爲何？

章節四

自由開源軟體審核的關鍵軟體觀念

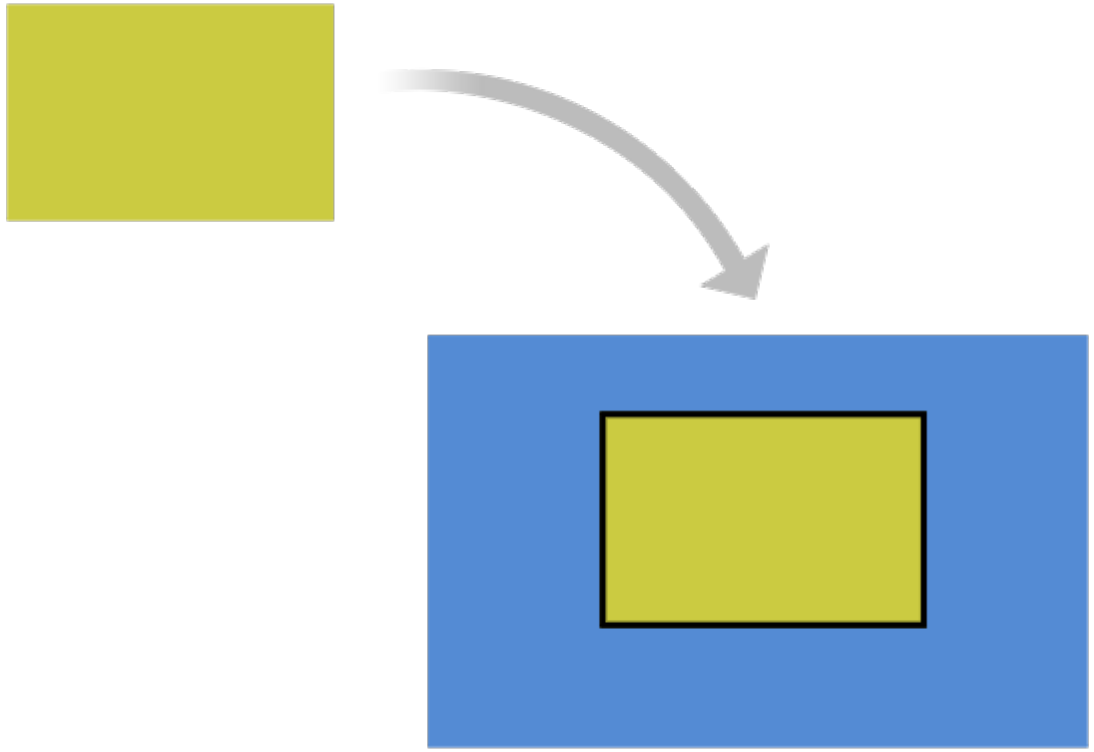
你想要如何使用自由開源軟體元件？

常見的使用情境包括：

- 合併(Incorporation)
- 連結(Linking)
- 修改(Modification)
- 轉變(Translation)

合併

開發人員可能會重製部份的自由開源軟體元件，到你的軟體產品之中。



相關的字詞包括：

- 整合(Integrating)
- 融合(Merging)
- 貼上(Pasting)
- 改用(Adapting)
- 嵌入(Inserting)

連結

開發人員可能會連結或加入自由開源軟體授權元件，與你的軟體產品一起運作。

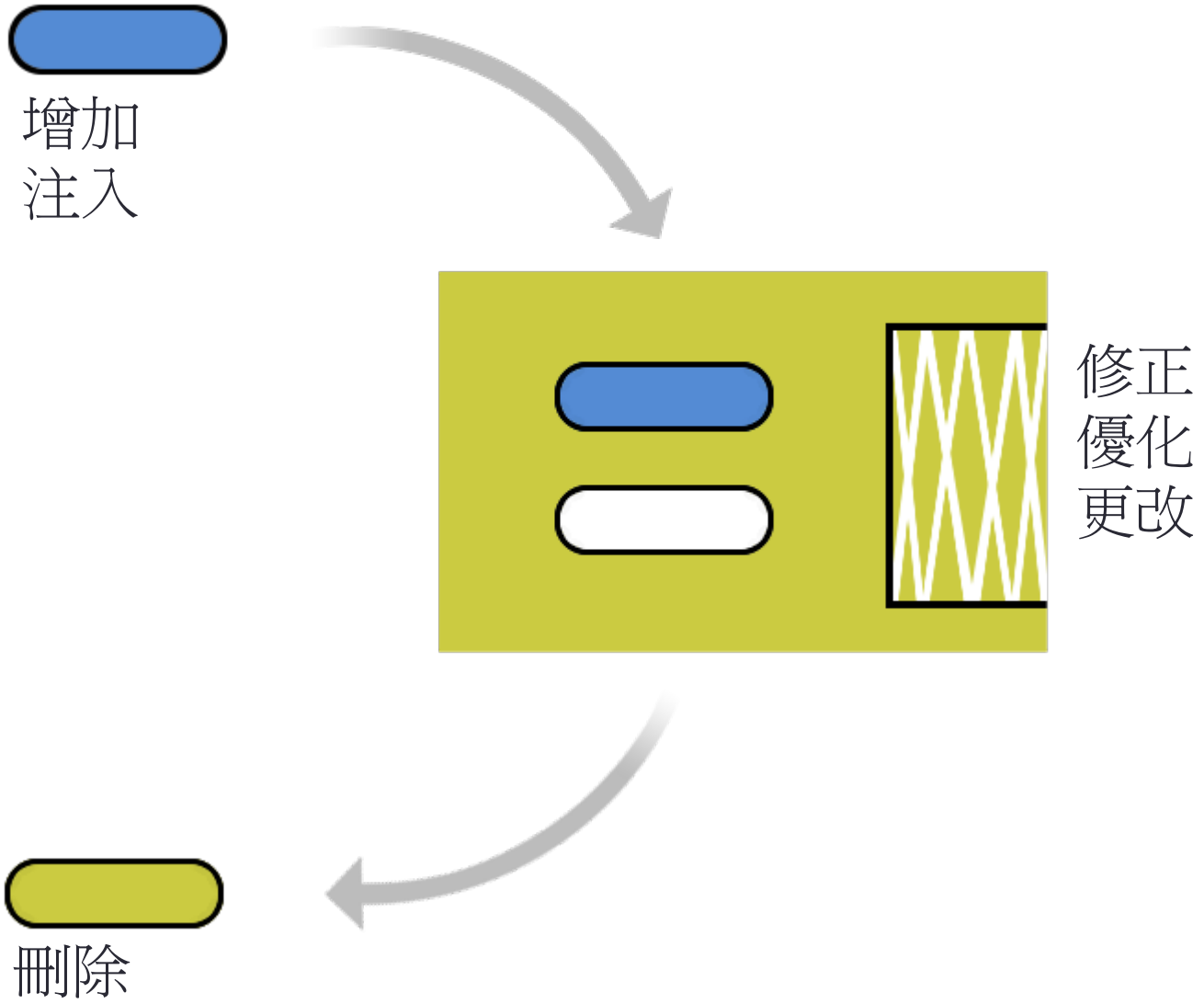
相關的字詞包括：

- 靜態/動態連結(Static/Dynamic Linking)
- 配對(Pairing)
- 結合(Combining)
- 利用(Utilizing)
- 打包(Packaging)
- 建立相依性(Creating interdependency)

修改

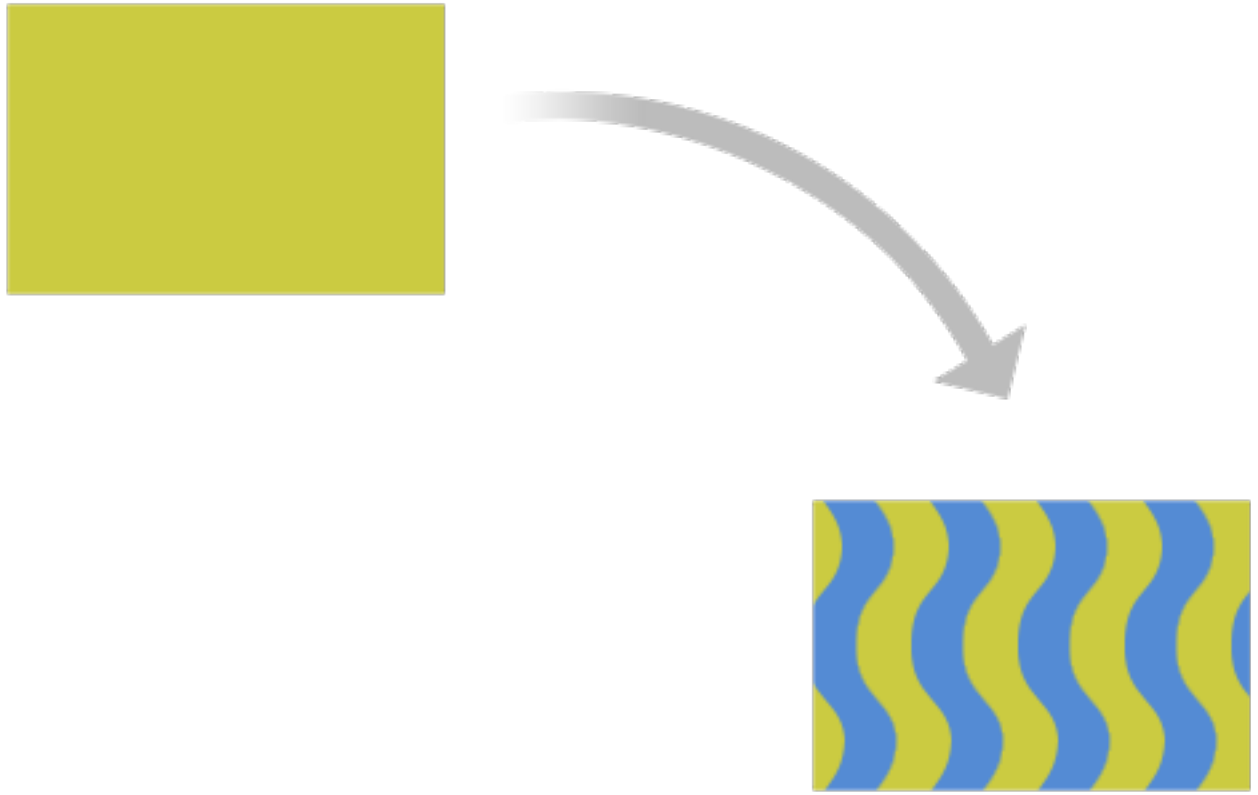
開發人員可能會對自由開源軟體元件進行變動，包括：

- 增加/注入新的程式碼到自由開源軟體元件裡
- 對自由開源軟體元件進行修正、優化，或更改
- 刪除或移除程式碼



轉變

開發者可能會轉化程式碼的狀態



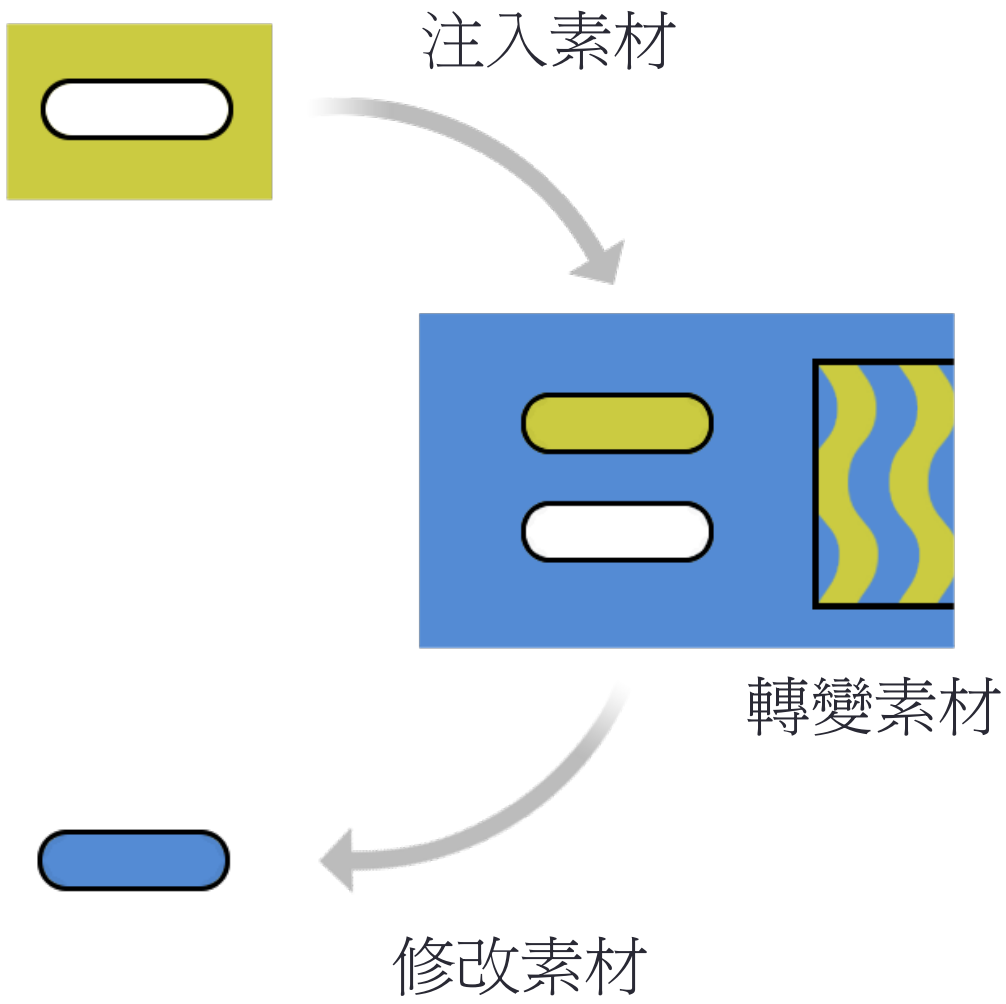
例子包括：

- 將中文翻譯成英文
- 將C++轉變爲Java
- 編譯成二進位執行檔

開發工具

開發工具可能會在幕後執行某些操作行爲。

例如，開發工具可能會將其自身的部份程式碼注入至輸出成果。



自由開源軟體元件如何被散布？

- 誰會收受到這些軟體？
  - 顧客/合作夥伴
  - 社群項目
  - 在商業團體範圍內的另一個法人(這可能會被視爲散布)
- 傳遞的形式是什麼？
  - 以程式源碼傳遞
  - 以二進位執行檔傳遞
  - 預載到硬體裡

檢測你的了解程度

- 合併(Incorporation)是什麼？
- 連結(Linking)是什麼？
- 修改(Modification)是什麼？
- 轉變(Translation)是什麼？
- 評估散布的重要要素是什麼？

章節五

進行自由開源軟體審核

自由開源軟體審核

- 在專案及產品管理與工程師，已就推薦的自由開源軟體元件進行可用性及品質的審核後，使用該選定元件牽涉到的權利與義務關係之審核，應被啓動。
- 蒐集相關資訊
- *自由開源軟體審核*流程，是自由開源軟體合規專案的關鍵元素。透過此流程，公司得以分析其採用的自由開源軟體，並理解其權利與義務關係。
- 自由開源軟體審核流程，包含以下幾個步驟：
  - 蒐集相關資訊
  - 分析並理解授權的義務性規定
  - 提供與公司政策與商業目標相合的使用指導

啓動自由開源軟體審核

任何在公司裡職司與自由開源軟體有關的人，都應該能夠啓動自由開源軟體審核，包括專案或產品管理人員、工程師，以及法務。

注意：此流程通常會在，基於自由開源軟體的新軟體被工程師或外部承包商選用時啓動。

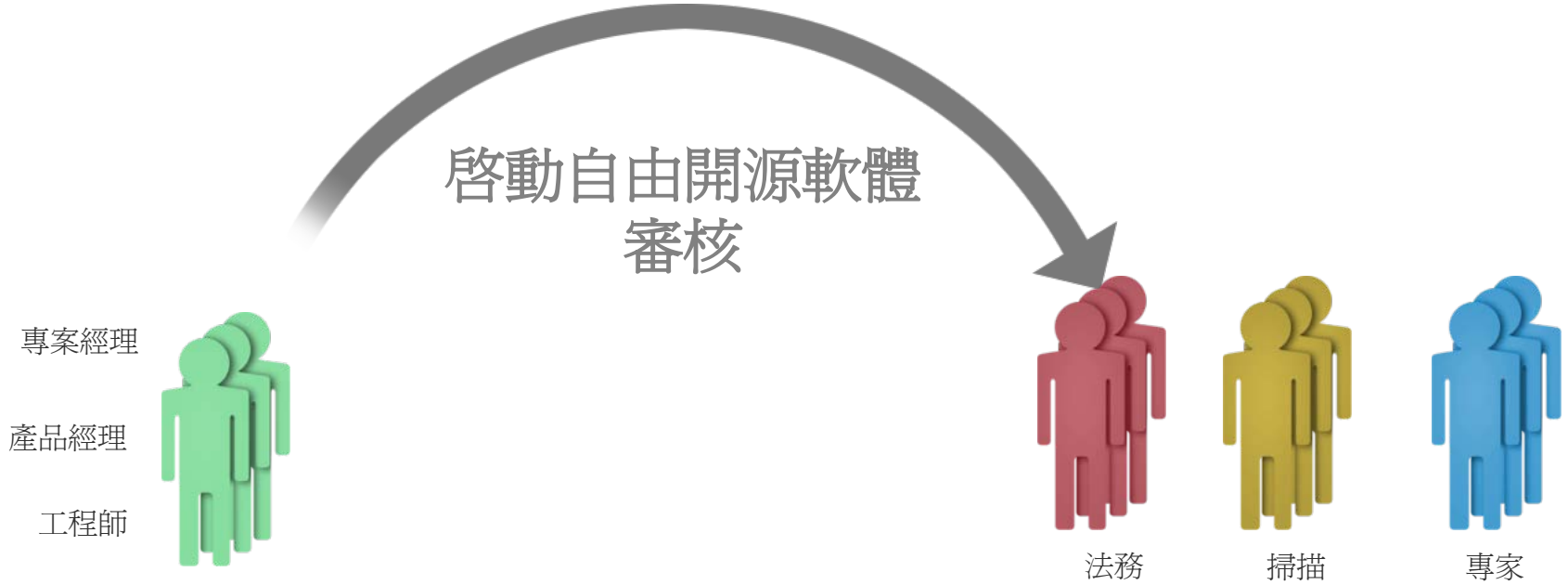
你需要蒐集哪些資訊？

在分析自由開源軟體使用時，你需要蒐集關於自由開源軟體元件辨識資訊，它的來源，及能被如何使用。這些資訊可能包括：

- 套件名稱(Package name)
- 與套件相關的社群狀態(活動、各種成員狀況、回應程度)
- 版本(Version)
- 下載或源碼網址(URL)
- 著作權利人
- 授權條款及授權條款網址
- 姓名標示及其他聲明和其網址
- 對於修改部分的描述

- 相依性清單(List of dependencies)
- 在產品中的用途
- 第一個包含此套件的發布產品
- 程式源碼被維護的位置
- 在之前的其他脈絡是否已被同意使用
- 是否是由外部承包商處取得
- 開發團隊的接觸點
- 著作權聲明、姓名標示，供應商修改部分的程式源碼(若授權義務性要求須被滿足的話)

自由開源軟體審核團隊



自由開源軟體審核團隊，包括支援、指導、協調及審核自由開源軟體使用的公司代表們。這些代表或會包含：

- 辨識與評估授權義務性規定的法務小組
- 支援源碼掃描及工具輔助，以協助辨識與追蹤自由開源軟體使用的掃描小組
- 與企業利益、商業授權、出口規範等等部門共工，而可能會被自由開源軟體使用影響到的工程專家群

分析自由開源軟體的使用提議

自由開源軟體審核團隊，在提供議題指導之前，應先評估已蒐集資訊。這可能包括掃描程式源碼，以確認資訊的正確性。

自由開源軟體審核團隊應考量：

- 程式碼及其相關資訊是否完整、一致，並且準確？
- 聲稱的授權是否與程式碼檔案裡顯示的一致？
- 該授權是否容許與軟體裡的其他元件一起使用？

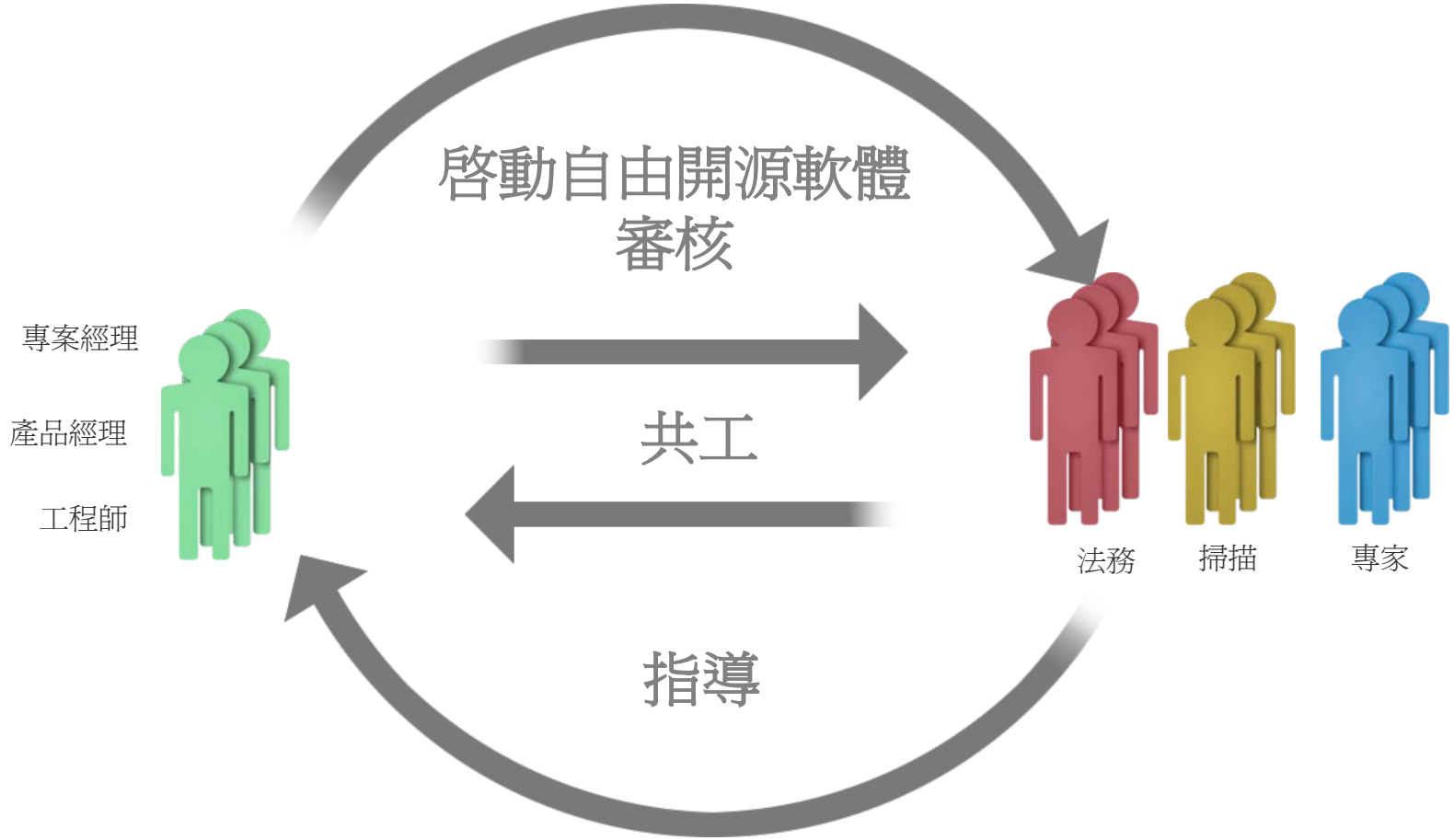
程式源碼掃描工具

OPENCHAIN

- 有許多不同的自動化開放源碼掃描工具。
- 這些方案皆呼應到特定的需求 – 也因爲這個原因 – 並無單一方案能解決所有可能的挑戰
- 公司選擇與他們特定市場領域與產品最相合的方案
- 許多公司兼採自動化工具及人工審核
- 自由供取用的開放源碼掃描工具 FOSSology 是一個優良範例，這個項目是由 Linux Foundation 所主持：

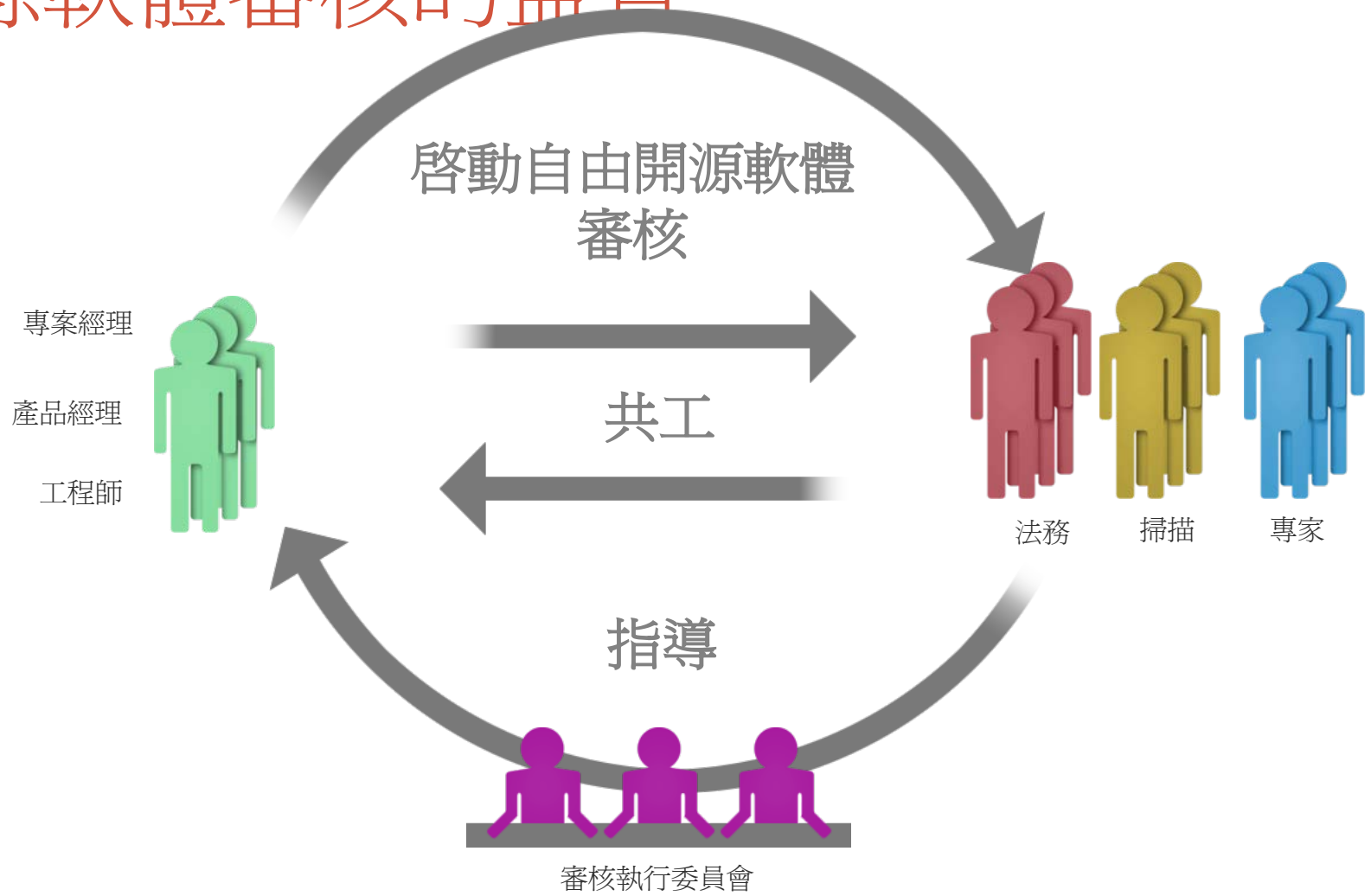
  https://www.fossology.org

透過自由開源軟體審核進行共工



自由開源軟體審核流程跨越學科領域，包括工程、商務，以及法務團隊。它必須保持互動性，以確保所有那些團隊皆對議題有正確理解，並能建立明確的共享性指導文件。

自由開源軟體審核的監督



自由開源軟體審核的流程，應該要有執行性的監督，以解決歧見，並核可最重要的決策。

檢測你的了解程度

- 自由開源軟體審核的目的爲何？
- 若你要使用自由開源軟體元件，第一個應採行的行動爲何？
- 如果你對使用自由開源軟體有疑問，應該怎麼做？
- 爲了自由開源軟體審核，你可能需要蒐集哪些種類的資訊？
- 什麼資訊可以協助辨識軟體是被誰授權的？
- 當自由開源軟體元件是從外部承包商而來，哪些額外資訊對於審核它是重要的？
- 在自由開源軟體審核裡，可以採行哪些步驟，來評估所蒐集資訊的品質？

章節六

端對端的合規管理（流程範例）

中小型公司查核清單的範例

OPENCHAIN

持續性的合規工作事項：

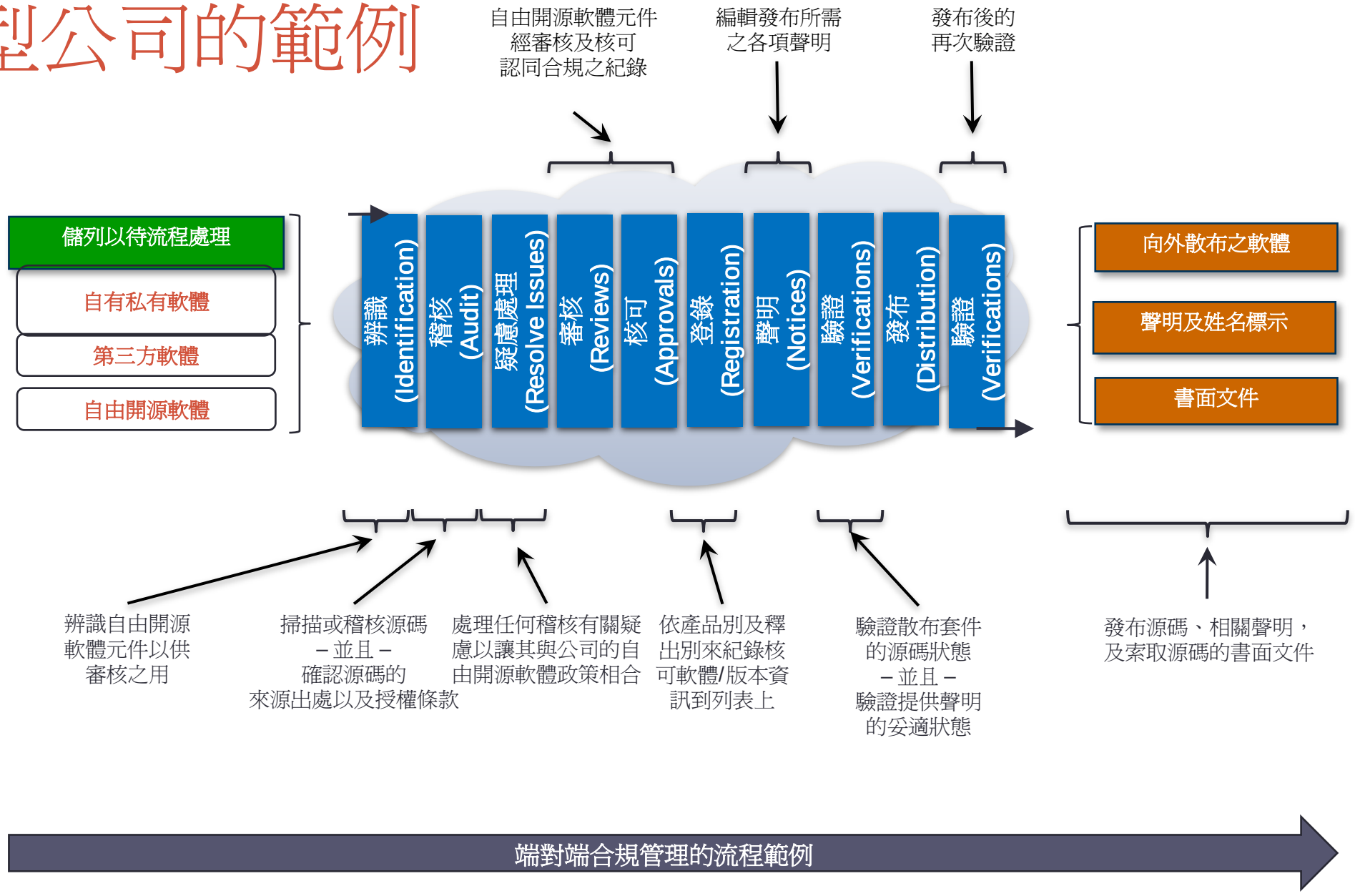
1. 在取得/開發的早期過程即發掘所有的自由開源軟體
2. 審核及批准所有使用到的自由開源軟體元件
3. 查驗滿足自由開源軟體義務性要求的必要資訊是否具足
4. 審核及批准任何對外部自由開源軟體項目的貢獻

需要的支援項目：

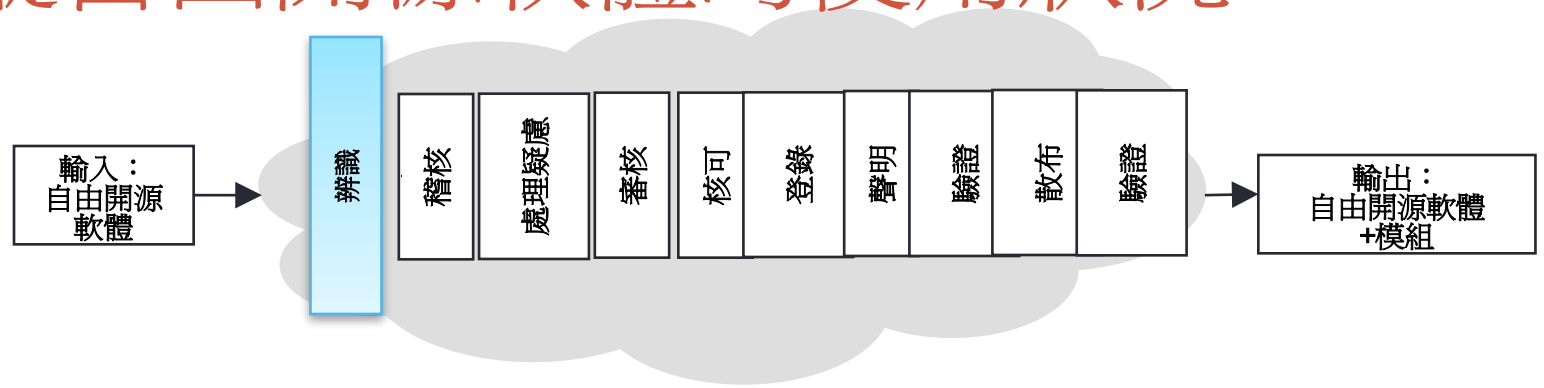
1. 確認勝任的合規工作人員，並就其職務責任指派清楚的界限
2. 採納到既存的企業管理流程裡，來支持自由開源軟體合規專案
3. 提供組織的自由開源軟體政策之訓練課程給所有人
4. 對所有自由開源軟體合規舉措進行歷程追蹤

你可在此取得這些項目的詳細查核清單: https://www.linuxfoundation.org/projects/opencompliance/self-assessment-compliance-checklist

企業型公司的範例


自由開源軟體元件
經審核及核可
認同合規之紀錄
編輯發布所需
之各項聲明
發布後的
再次驗證
儲列以待流程處理
自有私有軟體
第三方軟體
自由開源軟體
辨識 (Identification)
稽核 (Audit)
疑慮處理 (Resolve Issues)
審核 (Reviews)
核可 (Approvals)
登錄 (Registration)
聲明 (Notices)
驗證 (Verifications)
發布 (Distribution)
驗證 (Verifications)
向外散布之軟體
聲明及姓名標示
書面文件
辨識自由開源
軟體元件以供
審核之用
掃描或稽核源碼
– 並且 –
確認源碼的
來源出處以及授權條款
處理任何稽核有關疑
慮以讓其與公司的自
由開源軟體政策相合
依產品別及釋
出別來紀錄核
可軟體/版本資
訊到列表上
驗證散布套件
的源碼狀態
– 並且 –
驗證提供聲明
的妥適狀態
發布源碼、相關聲明，
及索取源碼的書面文件
端對端合規管理的流程範例

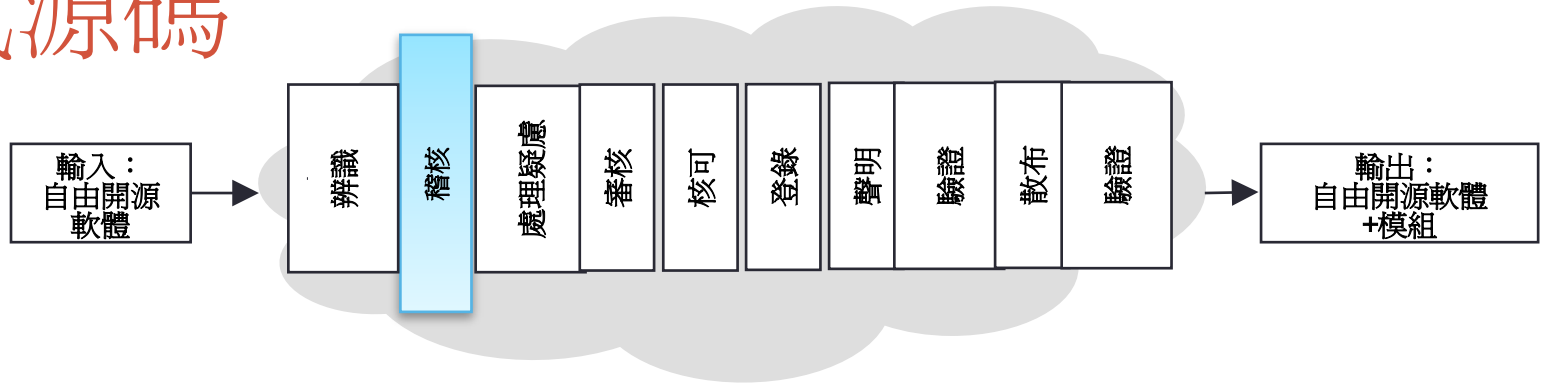
辨識及追蹤自由開源軟體的使用狀況



辨識自由開源軟體元件

- 步驟：
  - 來自工程師的輸入要求
  - 掃描軟體
  - 對第三方軟體的發現盡相當努力(Due Diligence)
  - 將人工辨識之新元件資訊加到知識庫
- 成果：
  - 對該自由開源軟體的合規紀錄被建立或更新
  - 依自由開源軟體政策所訂，窮盡或限定範圍內，對源碼審核之稽核將被要求。

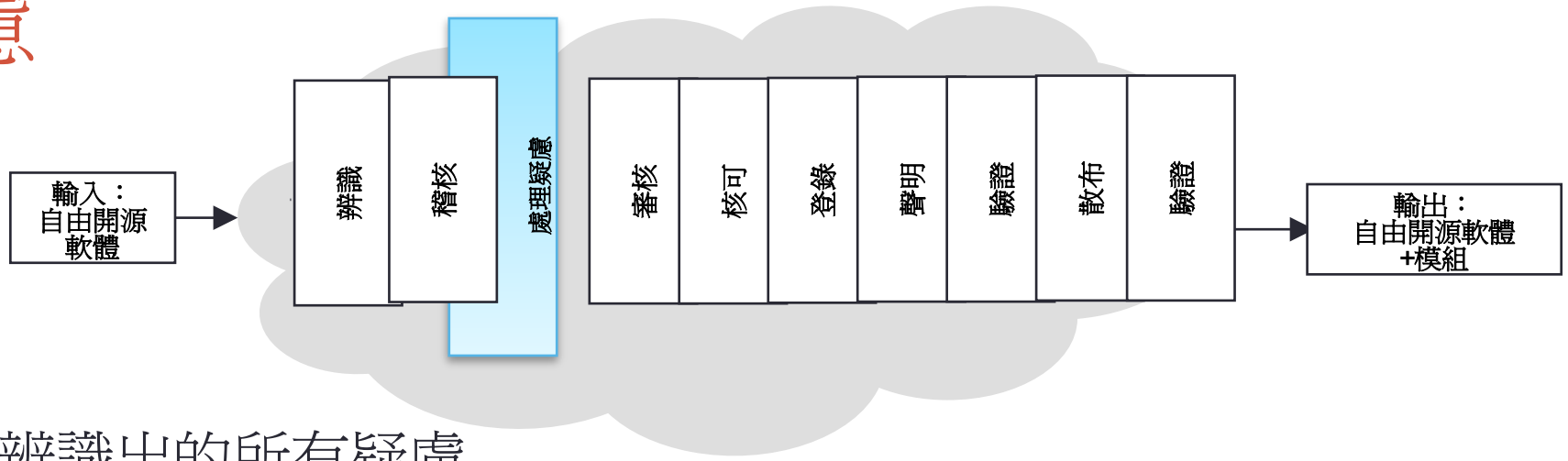
稽核程式源碼



辨識自由開源軟體授權條款

- 步驟：
  - 供稽核之程式源碼被辨識
  - 源碼或已使用軟體工具進行掃描
  - 稽核或掃描的「成果(Hits)」被審核及驗證，而得作爲該程式碼適宜的來源資訊
  - 稽核或掃描依該軟體的開發與釋出週期而被反覆操作
- 成果：
  - 稽核報告能用以辨識：
    1. 程式源碼的原始出處及其授權條款
    2. 有待處理的疑慮

處理疑慮

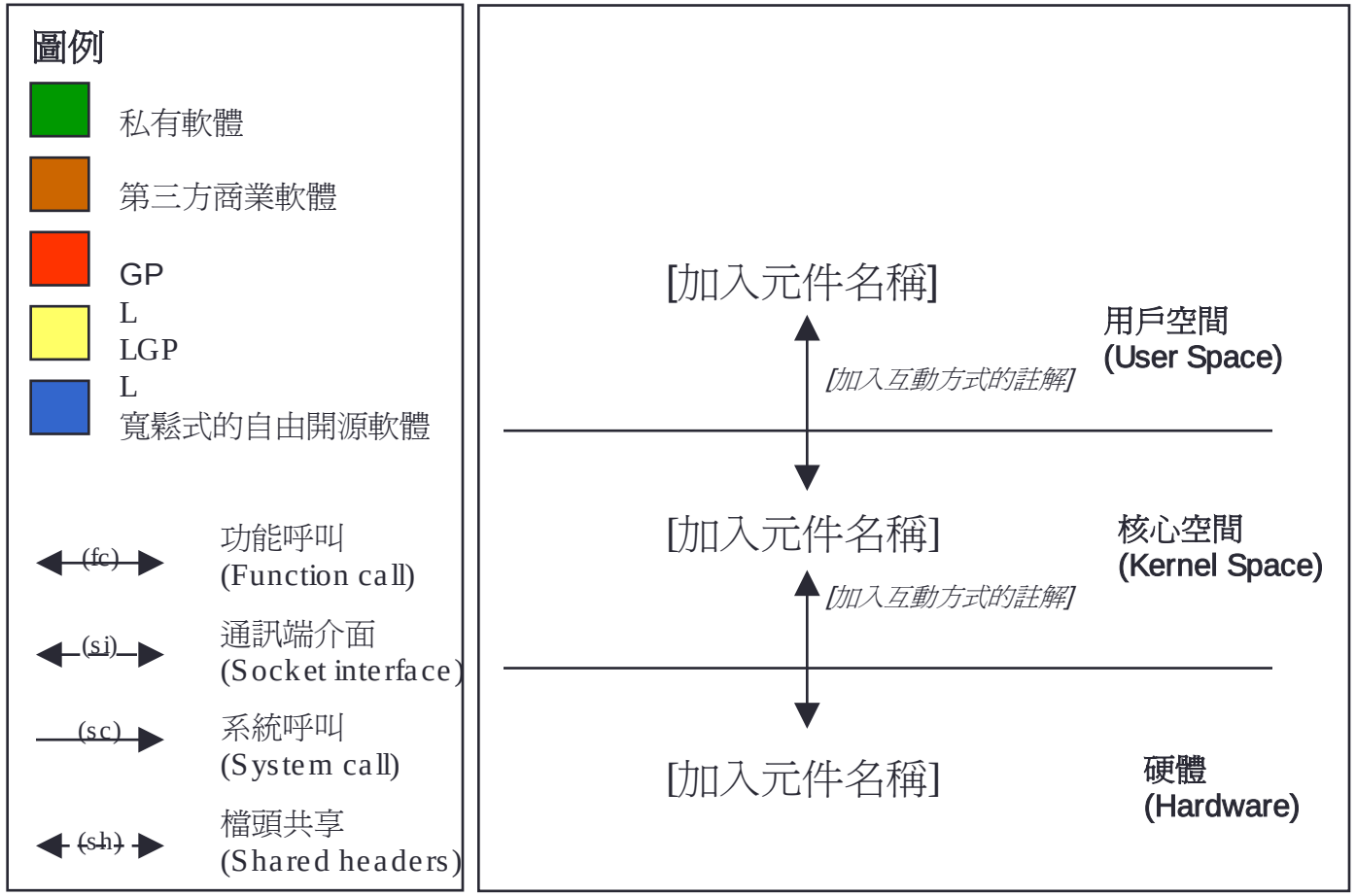


處理稽核過程辨識出的所有疑慮

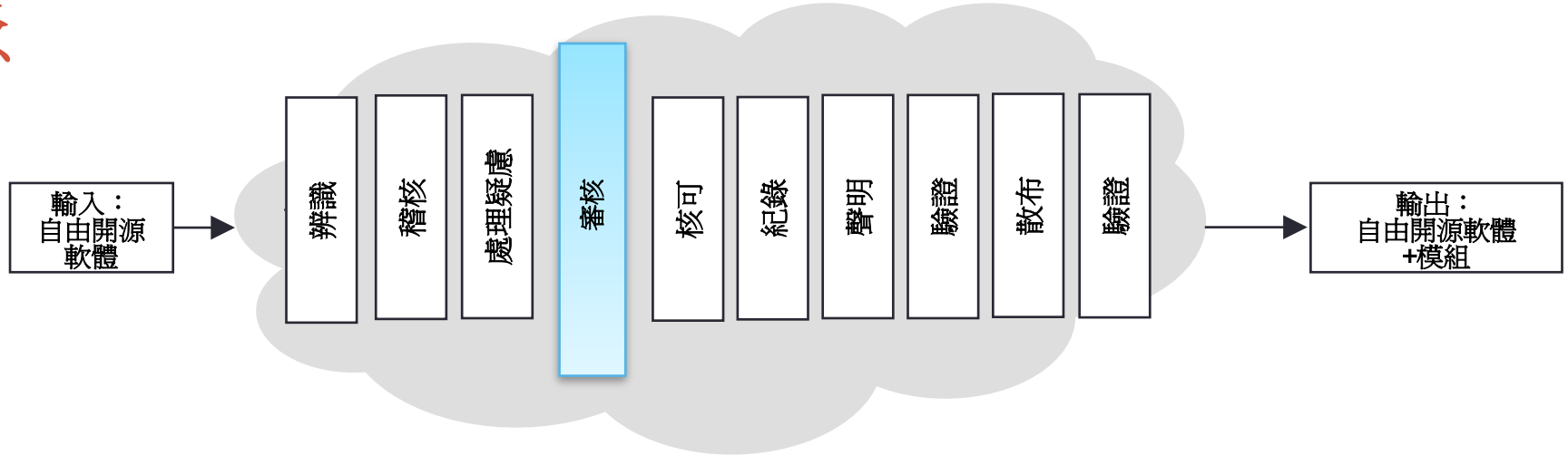
- 步驟：
  - 提供反饋予相應的工程師，以處理在稽核報告裡，與你的自由開源軟體政策衝突的疑慮
  - 該工程師接續就相關的程式源碼實施自由開源軟體審核(樣板可參閱下一張簡報)
- 成果：

  對報告裡每一個被標記的檔案作處理，及對任何標記授權衝突的狀況作處理

審核架構(樣板範例)


圖例
私有軟體
第三方商業軟體
GP
L
LGP
L
寬鬆式的自由開源軟體
(fc)
功能呼叫
(Function call)
(si)
通訊端介面
(Socket interface)
(sc)
系統呼叫
(System call)
(sh)
檔頭共享
(Shared headers)
[加入元件名稱]
[加入互動方式的註解]
用戶空間
(User Space)
[加入元件名稱]
核心空間
(Kernel Space)
[加入互動方式的註解]
[加入元件名稱]
硬體
(Hardware)

執行審核



審核已處理之疑慮以確認其與你的自由開源軟體政策相合

步驟：

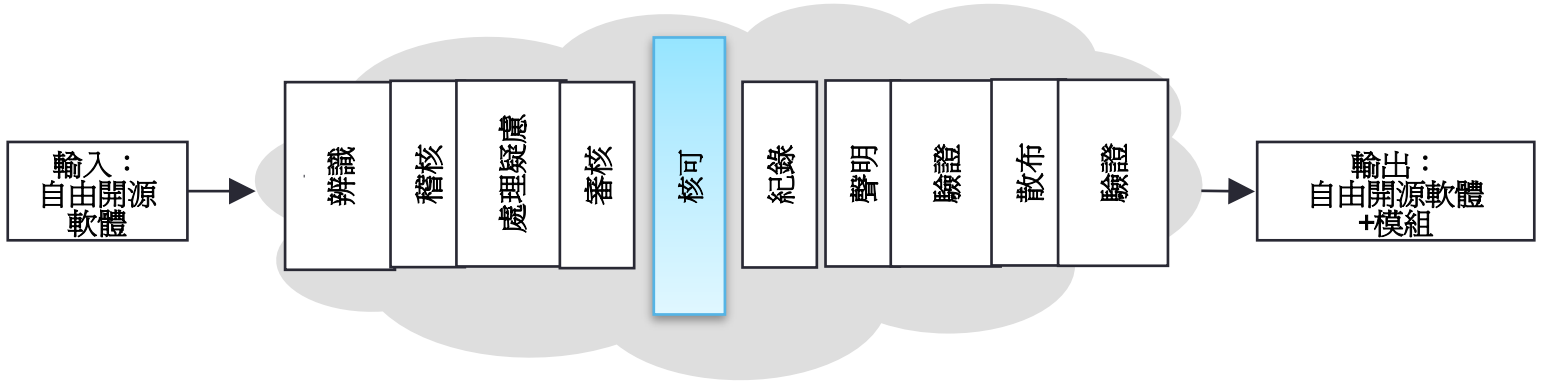
- 於審核工作人員裡，應包含適切對應的管理階層
- 依你的自由開源軟體政策爲參據來實施審核

成果：

- 確保在稽核報告裡的軟體與自由開源軟體政策相合
- 準備往下一個步驟前(例如：核可前)，保存稽核報告的發現，並標註已處理的疑慮

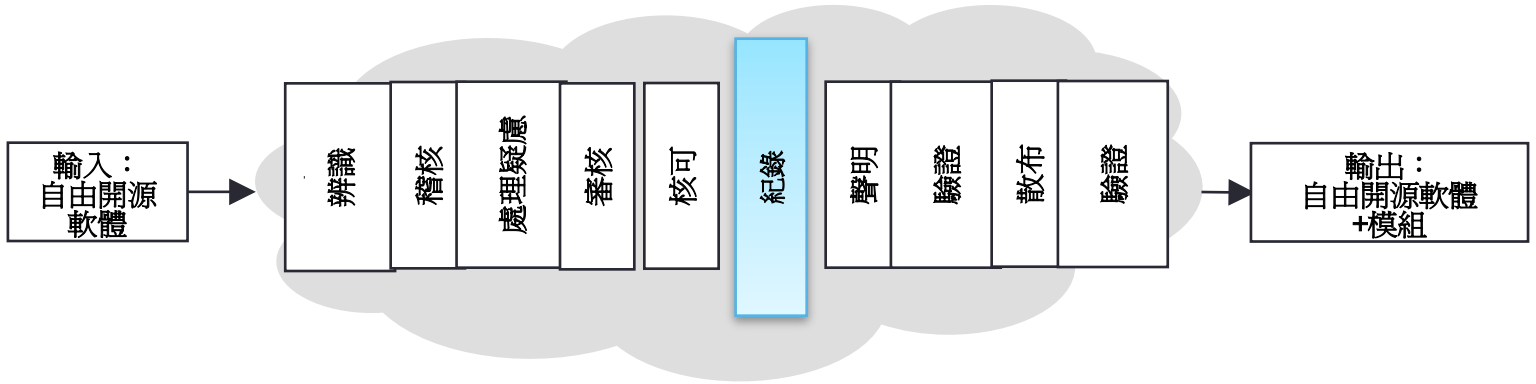
核可

- 根據上一個步驟的軟體稽核及審核結果，軟體可能被核可或可能不被核可使用
- 核可時，必須註明被核可的自由開源軟體版本、被核可元件的使用模式，以及其他依自由開源軟體授權條款應施行的義務性要求
- 核可須對應到適宜的行政管理階層來進行

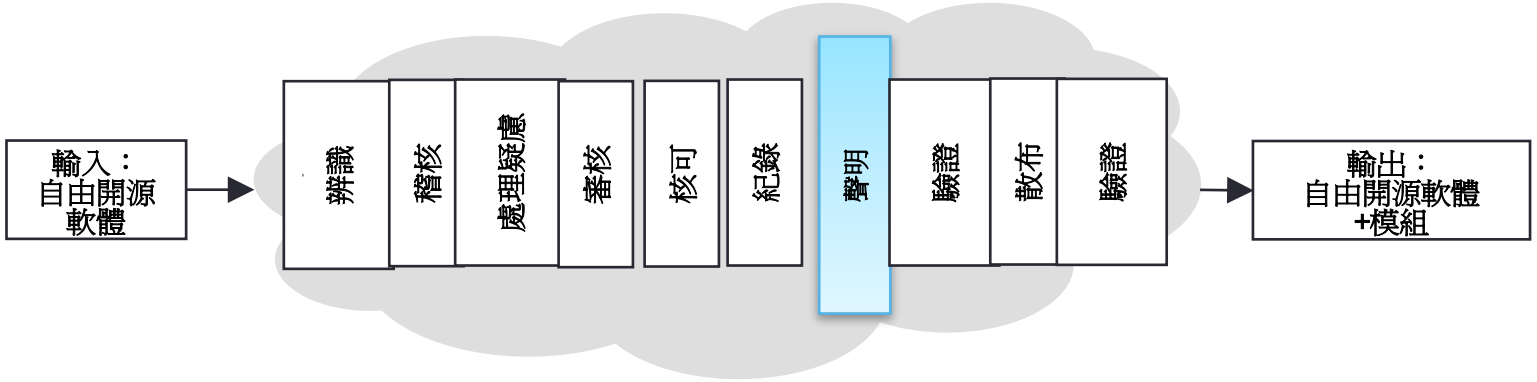


紀錄 / 核可追蹤

- 當一個自由開源軟體元件被核可在產品中使用時，其應被加入該產品的軟體清單
- 該項核可及核可的條件，必須被登記紀錄在可追蹤系統裡
- 若新版本的自由開源軟體元件或新的使用模式被提出時，該追蹤系統必須清楚顯示這需要一個新的核可



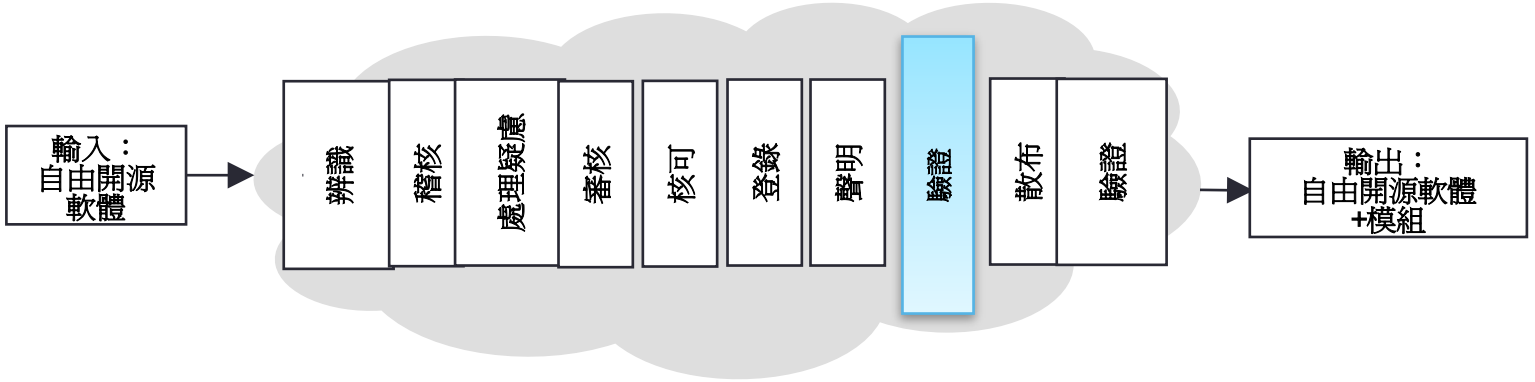
聲明



爲散布產品中任一自由開源軟體備妥適宜的聲明：

- 藉由完整著作權及姓名標示聲明之提供，來承認自由開源軟體的使用
- 通知產品的終極使用者，如何獲得自由開源軟體程式源碼的副本（當此要求適用時，例如 GPL 及 LGPL 即爲此種狀況）
- 應需求重製產品裡自由開源軟體程式碼之全部授權協議文件

散布前的驗證

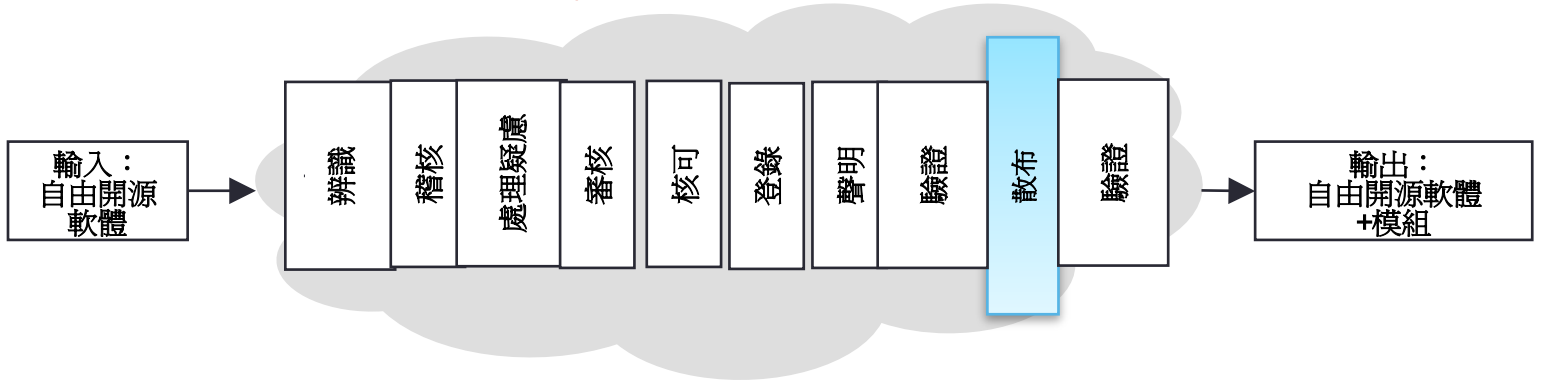


驗證散布的軟體已經過審核及核可

- 步驟：
  - 驗證預計散布的自由開源軟體套件已經過辨識及核可
  - 驗證已經過審核的程式源碼與販售產品裡相對應的二進位執行檔是符合的
  - 確保已被審核的源碼符合相對應的產品執行檔
  - 驗證所有相應的聲明已被列入，以告知終端使用者其索取已被辨識之自由開源軟體程式源碼的權利
  - 確保所有相關的聲明已被納入好讓終極用戶知道他們能獲取相關自由開源軟體的權益
  - 驗證合規於其他已被辨識的義務性要求

- 成果：
  - 使散布的套件僅會包含已經過審核及核可的軟體
  - 「供散布的合規稽證(Artifacts)」(依 OpenChain 規範書所定義)，包括被列入散布套件或其他投遞模式所相對應的聲明檔案

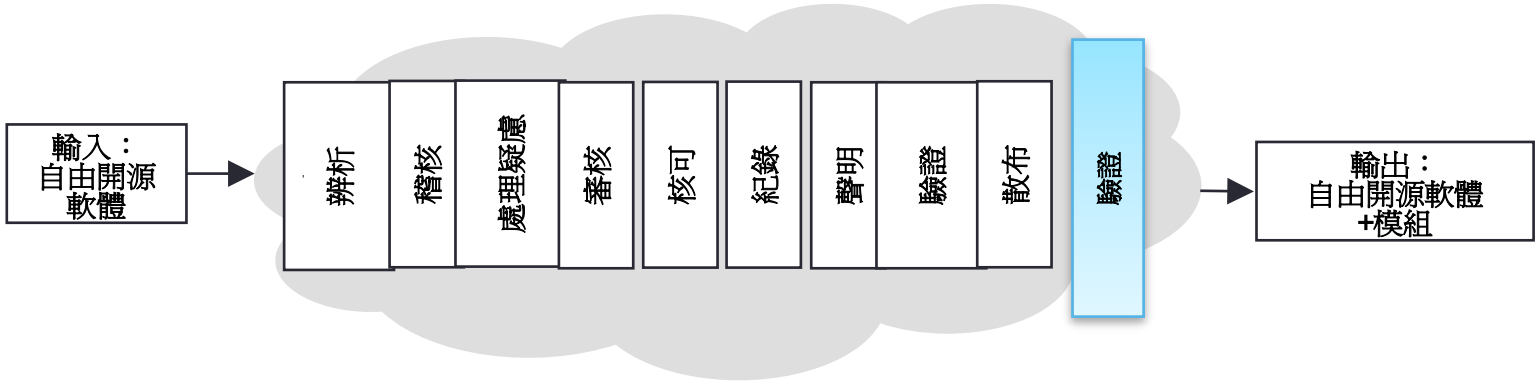
相應程式源碼的散布



依據要求提供相應的程式源碼

- 步驟：
  - 提供伴隨任何相關聯建置工具以及文件的對應程式源碼（例如，上傳到散布網站上或列入散布套件裡）
  - 此相應程式源碼應被辨識與標記，以和其產品及版本別對應
- 成果：
  - 提供相對應程式源碼的義務性規則被滿足

最後驗證



確認合規於授權條款的義務性規定

- 步驟：
  - 驗證相對應的程式源碼（若有的話）已經被正確地上傳或散布
  - 確保其他授權條款有被遵守
  - 驗證上傳或散布的程式源碼與經核可的版本是相對應的
  - 驗證所需聲明已被適當地發布與提供
  - 驗證其他被辨識出的義務性要求已達到
- 成果：
  - 驗證供散布的合規稽證(Artifacts)已被適切地提供

檢測你的了解程度

OPENCHAIN

- 在合規盡職工作(compliance due diligence)裡牽涉到哪些事項？(就我們的範例流程裡高項次的步驟作說明)
  - 辨識(Identification)
  - 稽核程式源碼(Audit source code)
  - 處理疑慮(Resolving issues)
  - 執行審核(Performing reviews)
  - 核可(Approvals)
  - 紀錄/追蹤核可(Registration/approval tracking)
  - 聲明(Notices)
  - 散布前的驗證(Pre-distribution verifications)
  - 相應程式源碼的散布(Accompanying source code distribution)
  - 驗證(Verification)
- 什麼是結構性審核要追求的？

章節七

避開合規陷阱

合規陷阱

這個章節將會描述一些潛在的合規陷阱，以在合規程序中避免之：

1. 智慧財產(IP)陷阱
2. 授權條款合規陷阱
3. 合規流程陷阱

智慧財產陷阱

類型 & 描述	發現	避免
意外地將 Copyleft 自由開源軟體囊括進私有軟體或第三方程式碼： 這種類型的錯誤發生在開發過程，當工程師將自由開源軟體程式碼，添加到預定將採私有狀態之程式源碼，造成與自由開源軟體政策相牴觸之情況。	此類型之錯誤可以透過程式源碼的掃描或稽核，以找出與下列的可能相合： • 自由開源軟體程式源碼 • 著作權聲明 可以使用自動化程式源碼掃描工具來完成此目標	此類型的錯誤可透過以下方式避免： • 爲工程師人員提供合規疑慮、自由開源軟體授權條款差異，及列入自由開源軟體到私有程式源碼的隱憂之相關訓練 • 爲建置環境裡的所有程式源碼，定期執行程式源碼的掃描與稽核。

智慧財產陷阱

類型 & 描述	發現	避免
意外地將 Copyleft 自由開源軟體與私有軟體的程式源碼連結在一起： 這種類型的錯誤發生在將授權條款衝突或不相容的軟體連結的結果。連結產生的法律效果爲何，於自由開源軟體社群裡仍有爭議。	這種類型的錯誤可以使用相依性追蹤工具來發現，其可用來顯示不同軟體元件之間的連結性。	此類型的錯誤可透過以下方式避免： 1. 爲工程人員提供相關訓練，以避免連結到與你自由開源軟體政策有所牴觸的軟體元件，將能在這些法律風險上站穩腳步 2. 在你的建置環境上，持續地在執行相依性追蹤工具
透過修改程式源碼，將私有軟體程式碼包含到 copyleft 自由開源軟體裡	此類型之錯誤，可以透過稽核或掃描來辨識及分析你採用到自由開源軟體元件的程式源碼。	此類型的錯誤可透過以下方式避免： 1. 對工程人員提供訓練 2. 執行週期性的程式碼稽核

授權條款合規陷阱

類型 & 描述	避免
未能提供相應的程式源碼/適當的授權條款、姓名標示或聲明資訊	在產品上市前的產品釋出循環，使程式源碼擷取及發布一個核對清單項目(checklist item)，可避免此類型錯誤。
相應程式源碼提供不正確的版本	透過在合規流程裡添加驗證的步驟，確保二進位版本的相對應程式源碼被發布，可避免此類型錯誤。
對自由開源軟體元件修改部分未能提供相對應的程式源碼	透過在合規流程裡添加驗證的步驟，確保修改部分的程式源碼被發布，而非僅及於自由開源軟體元件的原始程式源碼，可避免此類型錯誤。

授權條款合規陷阱

類型 & 描述	避免
未對自由開源軟體程式源碼的修改進行標註： 未能依自由開源軟體授權條款的要求，去標註自由開源軟體程式源碼已經過變動。(或所提供與修改有關的資訊，在細節及清楚級別不充份，無法滿足授權條款)	此類型的錯誤可透過以下方式避免： 1. 於程式源碼散布前，將程式源碼的修改標記(markings)添加爲一個驗證步驟 2. 爲工程人員提供訓練，以確保其對將要釋出的所有自由開源軟體或私有軟體更新著作權標記(markings)或授權條款資訊

合規流程陷阱

描述	避免	預防
開發者未請求使用自由開源軟體的核可	就公司自由開源軟體政策及流程，提供工程人員訓練，能避免此類型的錯誤。	此類型的錯誤可透過以下方式預防： 1. 定期執行軟體平台的完整掃描以偵測任何「未經揭露」的自由開源軟體是不是被使用了 2. 就公司的自由開源軟體政策及流程，提供工程人員訓練 3. 將合規事宜列入職員的績效評估
未參與自由開源軟體訓練	確認將自由開源軟體訓練的完成，視爲職員專業養成計畫之一環，並將其完成視爲績效評估的一部份，能避免此類型的錯誤。	透過指示工程人員必須在特定日期前完成自由開源軟體訓練課程，能預防此類型的錯誤

合規流程陷阱

描述	避免	預防
未對程式源碼進行稽核	此類型的錯誤可透過以下方式避免： 1. 定期執行程式源碼的掃描/稽核 2. 確保稽核是開發流程反覆執行的里程碑	此類型的錯誤可透過以下方式預防： 1. 提供適當職員人力以免進度落後 2. 實施定期稽核
未對稽核發現進行處理（分析掃描工具或稽核所回報的「命中值(hits)」）	當稽核報告未終局結束時，不容許合規指派項目被標示已處理(例如關閉)，能避免此類型的錯誤。	在自由開源軟體合規流程中，實施未經核可則阻斷的機制，能預防此類型的錯誤
未在時限內取得自由開源軟體的審核	即使工程師仍未決定採用該自由開源軟體的程式源碼，仍及早開啓自由開源軟體審核的要求，能避免此類型的錯誤。	透過教育能預防此類錯誤

產品出貨前確保合規

- 公司必須在任何產品（以任何形式）出貨前確保合規的優先性
- 將合規順位提高能促進：
  - 在你的組織中更有效率的使用自由開源軟體
  - 與自由開源軟體社群及組織建立更好的關係

建立社群關係

當公司在商業產品中使用自由開源軟體時，最好要與自由開源軟體社群建立及維持良好關係；尤其是，你公司在使用及部署的自由開源軟體項目有關的特定社群。

此外，與自由開源軟體組織的良好關係，在被建議採取合規最佳方案時非常有用，當你經歷合規疑慮時也非常有幫助。

與軟體社群的良好關係，可能對雙向溝通也很有幫助：向上游推送(upstreaming)更新，以及從軟體開發者取得協助。

檢測你的了解程度

- 在自由開源軟體合規裡可能發生哪些類型的陷阱？
- 請舉一個智慧財產陷阱的例子。
- 請舉一個授權條款合規陷阱的例子。
- 請舉一個合規流程陷阱的例子。
- 提高合規優先序位有何好處？
- 和社群建立良好關係有何好處？

章節八

開發者準則

開發者準則

- 從質優並具良好支援的自由開源軟體社群選用程式碼
- 尋求指引
  - 就每一個你在使用的自由開源軟體元件皆取得正式的核可
  - 不將未經審核的程式碼登錄到任何內部的程式源碼庫(source tree)
  - 就自由開源軟體項目的外部貢獻取得正式的核可
- 保留既存的授權資訊
  - 不要從任何你所使用的自由開源軟體元件，移除或採任何方式妨礙既存的自由開源軟體著作權授權或其他授權資訊。所有於自由開源軟體元件裡的著作權及授權資訊都該被維持完整。
  - 除非依自由開源軟體授權條款的要求(例如，已經修改的版本需更換名稱)，不要去更動自由開源軟體元件的名稱。
- 應自由開源軟體審核流程所需來蒐集及保留自由開源軟體項目資訊

預見合規流程的需求

- 列入所需時間以在工作計畫依照已建立的自由開源軟體政策來進行
  - 依照開發人員指導書來使用自由開源軟體，特別是合併(incorporating)或連結(linking)自由開源軟體程式碼到私有或第三方程式源碼時，反之亦然。
  - 審核結構規劃，並避免混合受不相容自由開源軟體授權條款拘束的元件
- 永遠更新合規的驗證 – 對每個產品
  - 就不同產品(product-by-product)的基礎上驗證合規性：單單因爲自由開源軟體套件被核可使用在一個產品中，不必然代表它也會被核可使用在第二個產品裡。
- 及對每個自由開源軟體更新版本的升級
  - 確保同樣自由開源軟體元件的每一個新版本被審核並核可
  - 當你升級自由開源軟體套件的版本時，確定新版本的授權條款是與舊版本相同 (於版本升級之際授權條款的改變是可能發生的)
  - 若自由開源軟體項目的授權條款變更了，確定該合規紀錄被更新且新的授權條款不會製造衝突

將合規流程適用到所有的自由開源軟體元件

- 收受軟體
  - 採行步驟以了解從供應商處傳遞的軟體裡有什麼自由開源軟體
  - 就所有將被包含到你產品裡的軟體評估你的義務性要求
  - 永遠就你從軟體供應商處取得的程式源碼進行稽核，或者替代方案是，讓軟體供應商必須就任何你取得的程式源碼，遞交程式源碼稽核報告給你，作爲一項公司政策。

檢測你的了解程度

- 列舉一些開發者工作上採用自由開源軟體可以實施的一般準則。
- 你需要移除或修改自由開源軟體授權的檔頭資訊嗎？
- 列舉一些在合規流程裡的重要步驟。
- 一個之前已經審核自由開源軟體元件的新版本能如何製造新的合規疑慮？
- 你應如何描述收受軟體會有哪些風險？

透過 Linux Foundation 維護並免費提供的「給開發者的合規基礎」來學習更多：

https://training.linuxfoundation.org/linux-courses/open-source-compliance-courses/ compliance-basics-for-developers