

Default Cache Rule 預設策略優化

Executive Summary

問題：目前 Default Cache Rule 的預設快取策略，尚未提供更貼近 Web App / 靜態資源場景的 TTL 檔位，亦缺少 follow origin 能力，導致使用者在常見靜態資源與既有源站快取控制場景下，需額外配置規則才能達到合理快取效果。

目標：建置一套更標準化的 Default Cache Rule 預設策略，支援固定 TTL 檔位與 follow origin，並提供一致、可預期的快取行為。

適用範圍：GMA、CN2 Apollo、CN2 Jupiter、CN2 Poseidon、CDNetworks、Wangsu、Baishan、Tencent、CloudFront、Akamai、Akamai DD、Akamai AMD、Aliyun、Chunghwa、VNCDN、BytePlus CDN。

【修改】關鍵需求：提供固定 TTL 選項（2h / 1d / 7d / 28d）、預設 TTL 為 1d、支援 follow origin 且預設為 off；非自建節點不強制統一「有效 Cache-Control」與 fallback 行為；自建節點需參照 RFC 9111 評估 core-nginx 行為；並針對指定副檔名提供預設規則。本次新增預設副檔名為 .mp3 / .avif / .ipa / .apk / .bmp / .tiff。

【修改】成功指標：Default Cache Rule 在指定副檔名上可正確生效；follow origin 預設值正確；新增域名預設策略正確；既有業務相容性正常；.mp3、.avif、.ipa、.apk、.bmp、.tiff 可命中預設副檔名規則；非自建節點不破壞各 Provider 既有快取行為；自建節點需完成 core-nginx 行為評估後補充驗收標準。

本文件描述 Default Cache Rule 預設策略優化之產品需求，目標為提升預設快取行為的合理性與可用性，並降低使用者額外配置複雜規則的成本。

1. 文件資訊

項目	說明
文件名稱	Default Cache Rule 預設策略優化
專案代號	Cache Rule
版本	【修改】v1.2
文件狀態	Draft
相關 Jira	

涉及系統	CDN / Cache Rule / Console
最後更新日期	2026/05/05

2. 專案背景

目前 Default Cache Rule 需提供更符合實際 Web App / 靜態資源場景的預設快取能力。

在現況中，若使用者僅依賴預設 Cache Rule，常見問題如下：

預設策略需要更合理的 TTL 檔位，以兼顧更新頻率與命中率

源站已設定 Cache-Control 時，平台缺少 follow origin 能力

動靜態混合場景中，若預設規則不夠合理，可能造成命中率不足或回源偏高

因此，本專案將建置一套更標準化的 Default Cache Rule 預設策略，使使用者在不額外設定大量自定義規則的前提下，也能獲得合理、可預期的快取行為。

【新增】本次 follow origin 能力需區分自建節點與非自建節點：

非自建節點：因各 CDN Provider 對 HTTP 快取相關 header 的解讀與支援能力不完全一致，主要包含 Cache-Control、Expires，以及 Cache-Control directives，如 s-maxage、max-age、no-cache、no-store、private 等，本期不強制統一行為。

自建節點：需評估 core-nginx 代碼倉，確認其對 HTTP 快取相關 header 與 Cache-Control directives 的實際處理行為，並參照 RFC 9111 HTTP Caching 標準確認差異與產品預期。

3. 專案目標

3.1 核心目標

【修改】建立一套 Default Cache Rule 預設策略，提供固定 TTL 檔位與 follow origin 能力，提升 Web App 常見靜態資源的預設快取效果，並降低使用者額外配置規則的成本。

【新增】follow origin 本次預設值為 off，避免源站 header 未整理時產生非預期快取行為。

3.2 具體目標

支援固定 TTL 檔位：2h / 1d / 7d / 28d

預設 TTL 為 1d

支援 follow origin

【新增】follow origin 預設值為 off

【修改】非自建節點 follow origin 與 fallback 行為以各 CDN Provider 實際能力與既有行為為準，本期不強制統一

【新增】自建節點需評估 core-nginx 對 HTTP 快取相關 header 與 Cache-Control directives 的處理邏輯，並參照 RFC 9111 HTTP Caching 標準產出差異比對結果

僅針對預設支援副檔名套用 Default Cache Rule

【修改】預設副檔名模板本次新增 .mp3 / .avif / .ipa / .apk / .bmp / .tiff

提升動靜態混合場景下的命中率與預設策略可用性

4. 問題定義

4.1 預設策略彈性不足

若 Default Cache Rule 未提供合理的 TTL 檔位，使用者仍需自行建立自定義規則，增加使用門檻。

4.2 缺乏 follow origin 能力

【修改】當源站已正確配置 Cache-Control 時，平台需提供 follow origin 能力，使使用者可選擇沿用源站快取控制。

【新增】產品決策：follow origin 預設為 off。使用者需明確開啟 follow origin 後，才套用 follow origin 行為。

4.3 動靜態混合場景命中率不足

【修改】在 cocos 或其他混合型 Web App 業務中，若預設規則不夠合理，容易出現靜態資源未充分命中或不必要回源等情況。

【新增】用語澄清：本文件中不再使用 mixed_content_optimization 表示此快取策略優化。mixed_content_optimization 本次確認指的是 HTTPS 請求載入 HTTP 內容的 mixed content 問題，與 Default Cache Rule 預設快取策略無直接關係，本期不處理。

5. 現況策略與目標策略

5.1 現況策略

目前 Default Cache Rule 尚未形成一套完整的固定 TTL + follow origin + 預設副檔名模板設計。

現況風險如下：

預設策略不足以覆蓋 Web App 常見靜態資源場景

源站已提供 Cache-Control 時，平台無法直接沿用

【新增】不同 Provider 對 HTTP 快取相關 header 的解讀與支援能力不同，主要包含 Cache-Control、Expires，以及 Cache-Control directives，如 s-maxage、max-age、no-cache、no-store、private 等。若未說明清楚，可能導致產品承諾與實際行為不一致。

5.2 目標策略

Default Cache Rule 將調整為：

支援固定 TTL：2h / 1d / 7d / 28d

預設 TTL：1d

新增 follow origin

【新增】follow origin 預設值為 off

【修改】非自建節點 follow origin / fallback 行為以各 CDN Provider 實際能力與既有行為為準

【新增】自建節點需評估 core-nginx 行為，並參照 RFC 9111 HTTP Caching 標準產出差異比對結果

【修改】預設支援副檔名：css / js / png / jpg / jpeg / gif / svg / ico / webp / avif / bmp / tiff / woff / woff2 / ttf / otf / webm / mp4 / zip / mp3 / ipa / apk

【修改】本次新增 .mp3 / .avif / .ipa / .apk / .bmp / .tiff 至預設模板

5.3 目標策略特性

固定 TTL 檔位可降低配置複雜度

【修改】follow origin 可承接既有源站快取控制，但預設為 off，降低非預期行為風險

可對常見 Web App 靜態資源提供合理預設

可提升使用者預設配置的有效性

【新增】非自建節點不強制要求所有 Provider 的 HTTP 快取相關 header 與 Cache-Control directives 行為一致

6. 適用範圍

設計原則

新 Default Cache Rule 以「標準 TTL、支援 follow origin、預設可用」為核心原則，兼顧快取效果與系統可維護性。

CDN	驗收範圍
GMA / CN2 Apollo / CN2 Jupiter / CN2 Poseidon / CDNetworks / Wangsu / Baishan / Tencent / CloudFront / Akamai / Akamai DD / Akamai AMD / Aliyun / Chunghwa / VNCDN / BytePlus CDN	Default Cache Rule

【新增】Provider 行為限制說明：

雖然上述 Provider 在需求範圍內，但不同 Provider 對以下能力可能存在差異：

是否支援 Default Cache

是否支援 follow origin / honor origin

是否支援 fallback 到平台 TTL

是否能完整表達 HTTP 快取相關 header 與 Cache-Control directives 的語義，主要包含 Cache-Control、Expires、s-maxage、max-age、no-cache、no-store、private 等

是否已有明確下發鏈路與 contract test

若 Provider 不支援相關能力，產品與控制台不得靜默宣稱完整成功。

7. 範圍說明

7.1 In Scope

Default Cache Rule TTL 檔位優化

Default Cache Rule 新增 follow origin

【新增】follow origin 預設值為 off

【修改】非自建節點 follow origin 行為以各 Provider 既有能力為準，不強制統一 HTTP 快取相關 header 與 Cache-Control directives 判定

【新增】自建節點 core-nginx HTTP 快取相關 header 與 Cache-Control directives 行為評估

預設支援副檔名清單定義

【修改】.mp3 / .avif / .ipa / .apk / .bmp / .tiff 加入本次預設副檔名模板

動靜態混合場景的預設策略優化

控制台配置設計

新增域名預設套用新 Default Cache Rule

既有域名維持原設定，僅於使用者主動修改時套用新邏輯

驗收與測試方案

7.2 Out of Scope

任意 TTL 自定義組合

進階條件式 Cache Rule 引擎重構

Purge / Refresh 機制優化

API 設計與接口細節

- 【新增】mixed content 修復

- 【新增】HTTPS 頁面載入 HTTP 內容的處理

- 【新增】HTTP-to-HTTPS resource rewrite

- 【新增】瀏覽器 mixed content warning 消除

- 【新增】對所有第三方 CDN Provider 的 HTTP 快取相關 header 與 Cache-Control directives 行為做一致性抽象

8. 假設條件

各 CDN 皆可支援基礎 TTL 設定

- 【修改】各 CDN 是否支援 follow origin / honor origin，需依 Provider 實際能力確認

平台控制台可承載 TTL 與 follow origin 配置項

現有觀測平台可接收命中率、回源率與規則命中結果

- 【新增】非自建節點不假設所有 Provider 對 HTTP 快取相關 header 與 Cache-Control directives 行為一致

- 【新增】自建節點需透過 core-nginx 代碼倉確認實際行為，並參照 RFC 9111 HTTP Caching 標準產出差異比對結果

9. 使用者故事

9.1 平台使用者

身為平台使用者，我希望在不額外建立複雜自定義規則的前提下，即可透過 Default Cache Rule 獲得合理的 Web App 預設快取效果。

【修改】身為平台使用者，我希望若源站已設定 Cache-Control，平台可以提供 follow origin 選項，讓我可以選擇是否沿用源站設定。

【新增】身為平台使用者，我希望 follow origin 預設不要自動開啟，避免源站 header 未整理好時產生非預期快取行為。

9.2 維運與產品團隊

身為維運人員，我希望新的 Default Cache Rule 能讓預設行為更清楚且可預期，降低排查成本。

身為產品團隊，我希望新 Default Cache Rule 能提升預設規則的可用性，降低使用者上手門檻。

【新增】身為維運人員，我希望 Provider 不支援某項能力時，系統能明確回報或提示，而不是靜默忽略。

10. 技術規格

10.1 TTL 檔位

Default Cache Rule 支援以下 TTL：

2h

1d

7d

28d

預設值為：

1d

【新增】補充：此限制僅適用於 Default Cache Rule，不影響普通 Cache Rule TTL。

10.2 Follow Origin

Default Cache Rule 需支援 follow origin 開關：

follow origin = off：依平台原本設定的 TTL 執行

follow origin = on：優先遵循源站或 Provider 所支援的快取控制行為

【修改】若源站無有效 Cache-Control：非自建節點依各 Provider 實際能力與既有行為處理；自建節點待 core-nginx 行為確認後，再定義 fallback 條件

【新增】follow origin 預設值為 off。

10.3 Fallback

【修改】原 fallback 語義調整如下：

原 PRD 中「當 follow origin 開啟但源站未返回有效 Cache-Control 時，退回原本設定 TTL」的方向保留，但本期需區分自建節點與非自建節點處理。

10.3.1 非自建節點 / 第三方 CDN Provider

【新增】因各 CDN Provider 對 HTTP 快取相關 header 的解讀與支援能力不完全一致，主要包含 Cache-Control、Expires，以及 Cache-Control directives，如 s-maxage、max-age、no-cache、no-store、private 等，本期不強制統一「有效 Cache-Control」與 fallback TTL 行為。

非自建節點 follow origin 與 fallback 行為以各 Provider 實際支援能力與既有行為為準。

若 Provider 無法表達「follow origin + fallback TTL」，產品側需透過能力限制、禁配、warning 或 skipped 明細方式呈現，不應靜默宣稱完整成功。

10.3.2 自建節點

【新增】自建節點需評估 core-nginx 代碼倉，確認以下 HTTP 快取相關 header 與 Cache-Control directives 的實際處理行為：

項目	需確認事項
Cache-Control	是否正確解析 response cache directives
Cache-Control: s-maxage=N	shared cache 情境下是否有對應處理
Cache-Control: max-age=N	是否作為 freshness lifetime 依據
Cache-Control: max-age=0	是否視為需立即重新驗證或過期
Cache-Control: no-store	是否禁止儲存
Cache-Control: no-cache	是否要求重新驗證後才能使用
Cache-Control: private	shared cache 是否避免儲存

Expires	在缺少適用 Cache-Control 指令時是否作為 freshness 依據
無 Cache-Control / 無 Expires	是否 fallback 到平台設定 TTL
非法 / 無法解析 header	是否 fallback 到平台設定 TTL

【新增】參考依據：自建節點需參照 RFC 9111 HTTP Caching 標準，確認 core-nginx 對 HTTP 快取相關 header 與 Cache-Control directives 的處理邏輯是否符合產品預期。RFC 9111 定義 HTTP caches，以及控制 cache 行為或標示可快取回應訊息的相關 header fields，作為本次自建節點行為評估的主要標準依據。

【新增】注意：PRD 不直接固化所有 RFC 細節為產品規格。實際自建節點行為需由技術評估 core-nginx 實作後，產出差異清單與建議產品口徑，再回填驗收標準。

10.4 預設副檔名

【修改】Default Cache Rule 預設支援以下副檔名：

css / js / png / jpg / jpeg / gif / svg / ico / webp / avif / bmp / tiff / woff / woff2 / ttf / otf / webm / mp4 / zip / mp3 / ipa / apk

【修改】產品決策：本次新增 .mp3 / .avif / .ipa / .apk / .bmp / .tiff 至 Default Cache Rule V2 預設副檔名模板。

10.5 配置項

配置項	規格要求
ttl	2h / 1d / 7d / 28d
default ttl	1d
follow_origin	on / off
follow_origin default	【新增】off
fallback	【修改】非自建節點以 Provider 行為為準；自建節點待 core-nginx 評估後確認
extensions	【修改】固定預設集合：css / js / png / jpg / jpeg / gif / svg / ico / webp / avif / bmp / tiff / woff / woff2

	/ ttf / otf / webm / mp4 / zip / mp3 / ipa / apk
mixed_content_optimization	【修改】本期不納入；此詞指 mixed content 問題，非本次快取策略欄位

11. 非功能需求

11.1 可用性

Default Cache Rule 應可穩定生效，且不因 TTL 或 follow origin 設計導致明顯配置異常。

【新增】follow origin 預設為 off，以降低源站 header 未整理時造成非預期快取行為的風險。

11.2 效能

Default Cache Rule 不應對正常流量造成明顯性能退化，亦不應因 follow origin 引入不可接受的額外處理延遲。

11.3 可維運性

副檔名與策略映射需簡潔可維護，避免預設策略過度複雜。

【新增】Provider 不支援相關能力時，需明確呈現 skipped / unsupported / warning，不應靜默忽略。

11.4 可觀測性

【修改】需具備足夠觀測能力，以驗證 TTL 生效、follow origin 配置、規則命中、命中率與回源率變化。

【新增】由於非自建節點 fallback 行為不強制統一，本期不要求所有 Provider 都提供一致的 fallback 次數指標。自建節點 fallback 指標需待 core-nginx 行為確認後再定義。

12. Rollout 計畫

12.1 上線策略

新的 Default Cache Rule 採新舊配置分流方式導入，以降低對既有業務的影響並確保可控性。

導入原則如下：

既有域名不主動變更

已存在之域名，維持原有 Cache Rule 行為與設定，不因本次 Default Cache Rule 優化而自動調整。

既有域名僅在使用者主動修改時套用新邏輯

若使用者主動編輯 Default Cache Rule 相關設定，則依最新規格套用新的 TTL / follow origin / fallback 行為。

【新增】待補充決策：仍需進一步定義以下行為是否算主動升級：

使用者打開頁面後未修改直接保存

使用者只修改 TTL

使用者只修改 follow origin

使用者點擊「恢復預設模板」

internal batch 工具顯式傳入 path

新增域名預設套用新策略

新建立之域名，Default Cache Rule 預設採用本次定義之新策略，包括：

預設 TTL = 1d

支援 TTL 選項：2h / 1d / 7d / 28d

【新增】follow origin 預設值 = off

【修改】預設支援副檔名套用新模板，新增 .mp3 / .avif / .ipa / .apk / .bmp / .tiff

12.2 階段性導入

可依以下順序規劃：

先完成控制台配置與預設模板設計

【修改】先驗證 TTL 與 follow origin 預設值

【修改】驗證 .mp3 / .avif / .ipa / .apk / .bmp / .tiff 命中 Default Cache Rule

先以新增域名套用新 Default Cache Rule

驗證新建域名在實際場景下的命中率與相容性

【新增】非自建節點驗證配置下發與 Provider 既有行為相容

【新增】自建節點完成 core-nginx 行為評估，並參照 RFC 9111 HTTP Caching 標準產出差異比對結果

既有域名維持原狀，待使用者主動調整時再套用新邏輯

13. 監控與告警

13.1 監控指標

Default Cache Rule 命中數

不同 TTL 檔位使用分佈

follow origin 啟用比例

【修改】fallback 觸發次數：非自建節點不要求一致；自建節點待 core-nginx 行為確認後再定義

Cache hit ratio

Origin fetch ratio

【新增】Provider skipped 次數

【新增】unsupported provider 次數

13.2 告警規則與閾值（佔位）

當 Default Cache Rule 生效率異常下降時觸發告警

【修改】當 follow origin 啟用後出現大量錯誤或回源異常時觸發告警

【新增】當 Provider skipped / unsupported 數量異常升高時觸發告警

當 Cache hit ratio 顯著下降時觸發告警

當 Origin fetch ratio 顯著上升時觸發告警

14. 功能需求

14.1 TTL 能力

Default Cache Rule 需支援 2h / 1d / 7d / 28d 四種固定 TTL 檔位，預設值為 1d。

14.2 Follow Origin

【修改】Default Cache Rule 需支援 follow origin 開關，開啟後優先依照源站或 Provider 所支援的快取控制行為處理。

【新增】follow origin 預設為 off。新增域名建立 Default Cache Rule 時，follow origin 應預設為 off。

14.3 Fallback

【修改】Fallback 行為需依節點類型區分：

非自建節點：因各 CDN Provider 行為不同，本期不強制統一 fallback 語義。

自建節點：需待 core-nginx 行為確認後，定義「有效 Cache-Control」與 fallback 條件。

14.4 預設副檔名匹配

預設規則僅對指定副檔名生效，並以小寫方式進行匹配。

【修改】預設副檔名包含：

css / js / png / jpg / jpeg / gif / svg / ico / webp / avif / bmp / tiff / woff / woff2 / ttf / otf / webm / mp4 / zip / mp3 / ipa / apk

【修改】本次新增 .mp3 / .avif / .ipa / .apk / .bmp / .tiff。

14.5 動靜態混合場景優化

【修改】針對混合型 Web App 場景，需提升靜態資源命中率並降低不必要回源；本節不應與 mixed_content_optimization 混淆。

【新增】mixed_content_optimization 指 HTTPS 請求載入 HTTP 內容的 mixed content 問題，本期不納入 Default Cache Rule 預設快取策略優化範圍。

15. 控制台配置

15.1 TTL

控制台需提供以下下拉選項：

2h

1d

7d

28d

預設選中值為：

1d

15.2 Follow Origin

控制台需提供 on / off 開關。

【新增】預設值為 off。

15.3 建議文案

TTL

2h：適合更新頻率較高的靜態資源

1d：適合一般 Web App 預設資源

7d：適合較穩定的靜態資源

28d：適合長期穩定且版本化發布的靜態資源

follow origin

【修改】開啟後優先遵循源站快取控制。實際行為可能因 CDN Provider 能力與既有行為不同而有所差異。新增域名預設為 off，使用者需明確開啟後才生效。

預設副檔名

【新增】Default Cache Rule 會針對常見靜態資源副檔名套用預設快取策略，包含圖片、字型、影音、壓縮檔、點陣圖、音訊與行動應用安裝檔等資源。本次預設模板新增 .mp3 / .avif / .ipa / .apk / .bmp / .tiff。

【新增】若音訊、圖片、點陣圖或安裝檔資源更新頻繁或非版本化，建議使用者自行調整 Cache Rule。

16. 驗收標準

16.1 CDN 覆蓋驗收

CDN	驗收範圍
GMA / CN2 Apollo / CN2 Jupiter / CN2 Poseidon / CDNNetworks / Wangsu / Baishan / Tencent / CloudFront / Akamai / Akamai DD / Akamai AMD / Aliyun / Chunghwa / VNCDN / BytePlus CDN	Default Cache Rule

【新增】由於 Provider 能力可能不同，驗收需標示：

完整支持

支援 TTL，但不支援 follow origin

支援 follow origin，但不支援一致 fallback

不支援 Default Cache

本期暫緩或待確認 Provider

16.2 功能驗收條件

編號	驗收項	測試方法	期望結果
AC-2.1	TTL 生效	切換 2h / 1d / 7d / 28d 配置	按配置快取
AC-2.2	預設值正確	使用預設配置建立或啟用規則	預設 TTL 為 1d
AC-2.3	【修改】follow origin 預設值與可配置	新增域名建立 Default Cache Rule；開啟 / 關閉測試	【新增】follow origin 預設為 off；配置可保存，行為依節點 / Provider 能力執行
AC-2.4	【修改】fallback 行為	follow origin 開啟且源站 header 不完整	【修改】非自建節點依 Provider 行為；自建節點待 core-nginx 評估後驗收
AC-2.5	命中率提升	對比優化前後	命中率提升
AC-2.6	相容性	現有業務測試	無異常
AC-2.7	CDN 覆蓋完整	逐項驗證 CDN	所列 CDN 完成能力標示與驗證
AC-2.8	預設副檔名生效	對預設副檔名發請求驗證	規則正確命中

AC-2.9	【新增】.mp3 / .avif / .ipa / .apk / .bmp / .tiff 生效	對 .mp3、.avif、.ipa、.apk、.bmp、.tiff 資源發請求驗證	上述副檔名皆正確命中 Default Cache Rule
AC-2.10	【新增】mixed content 排除	驗證本期範圍	不驗收 mixed content 修復相關能力

16.3 驗收補充說明

預設配置下，TTL 應為 1d

【新增】新增域名預設 follow origin 應為 off

【修改】follow origin 開啟時，非自建節點行為以 Provider 能力與既有行為為準

【修改】源站未返回有效 Cache-Control 時是否 fallback，不在非自建節點做統一強制要求

【新增】自建節點需完成 core-nginx 行為確認後，再補充 fallback 驗收條件

預設副檔名應正確命中 Default Cache Rule

【修改】.mp3、.avif、.ipa、.apk、.bmp、.tiff 應正確命中 Default Cache Rule

既有域名不得因本次上線自動變更既有快取行為

新增域名應預設套用新 Default Cache Rule

既有域名在使用者主動修改後，才套用新邏輯

正常流量不得出現明顯性能下降或錯誤快取

【新增】本期不驗收 mixed content、HTTP-to-HTTPS rewrite、瀏覽器 mixed content warning 消除

17. 風險與限制

follow origin 若遇到源站 Cache-Control 不規範，可能產生非預期行為

【新增】因此 follow origin 預設為 off，需使用者明確開啟

【新增】不同 CDN Provider 對 HTTP 快取相關 header 與 Cache-Control directives 的解讀與支援能力不一致，可能導致同一配置在不同 Provider 上行為不同

【新增】非自建節點本期不強制統一 HTTP 快取相關 header 與 Cache-Control directives 行為

【修改】動靜態混合場景若策略過於激進，可能增加誤快取風險；新增 .mp3、.avif、.ipa、.apk、.bmp、.tiff 後，若使用者相關資源非版本化或更新頻繁，可能出現更新不即時風險

【新增】自建節點需依賴 core-nginx 評估結果。若 core-nginx 實際行為與 RFC 9111 HTTP Caching 標準或產品預期存在差異，需由技術與產品共同確認是否接受現況、補充限制說明，或另行排期調整

本版本不支援任意 TTL 自定義

【新增】本期不處理 mixed content 問題

18. 相依項目

各 CDN 支援基礎 TTL 設定與對應映射能力

【修改】各 CDN 是否支援 follow origin / honor origin，需依 Provider 能力確認

控制台支援 TTL 與 follow origin 配置項

觀測平台可接收命中率與回源率相關指標

【新增】控制台或內部工具需可呈現 unsupported provider / skippedPlatforms

【新增】自建節點需評估 core-nginx 對 HTTP 快取相關 header 與 Cache-Control directives 的處理行為，並參照 RFC 9111 產出差異比對結果

【新增】若需 Provider capability matrix，需後端提供能力配置來源

19. 開放問題追蹤

問題	Owner	Target Date
follow origin 預設是否維持為 off	[Add Owner]	[Add Date]
fallback 是否固定為退回原本設定 TTL	[Add Owner]	[Add Date]
28d 是否作為目前最大 TTL	[Add Owner]	[Add Date]

【新增】Provider 不支援 Default Cache / follow origin 時如何呈現	[Add Owner]	[Add Date]
【新增】BytePlus / GMA / AkamaiAMD 是否完整支持	[Add Owner]	[Add Date]
【新增】.mp3 / .avif / .ipa / .apk / .bmp / .tiff 是否加入本期預設模板	[Add Owner]	[Add Date]
【新增】mixed_content_optimization 是否納入本期	[Add Owner]	[Add Date]
【新增】既有域名主動升級邊界	[Add Owner]	[Add Date]
【新增】觀測資料來源	[Add Owner]	[Add Date]
【新增】前端 / 外部 API 交付邊界	[Add Owner]	[Add Date]

20. 里程碑建議

20.1 Phase 1 : 基礎設計

Default Cache Rule TTL 與 follow origin 設計確認

預設副檔名清單確認

【修改】.mp3、.avif、.ipa、.apk、.bmp、.tiff 預設模板確認

控制台配置項確認

【新增】mixed content 不納入本期確認

【新增】自建節點 core-nginx 行為評估與 RFC 9111 差異比對啟動

20.2 Phase 2 : 功能驗證

【修改】完成 TTL、follow origin 預設 off、fallback 邊界行為驗證

完成預設副檔名命中驗證

【新增】完成 .mp3、.avif、.ipa、.apk、.bmp、.tiff 命中驗證

完成動靜態混合場景測試

【新增】完成 unsupported provider / skippedPlatforms 情境驗證

20.3 Phase 3：驗收與上線

驗收完成

完成既有業務驗證

分階段導入新預設模板

【新增】確認本期不包含 mixed content 驗收項

21. 預期效益

讓 Default Cache Rule 更貼近 Web App 常見資源場景

降低使用者額外建立自定義規則的需求

提升預設快取命中率與整體可用性

【修改】讓源站已配置快取控制的業務可選擇沿用既有設定

【新增】透過 follow origin 預設 off，降低源站 header 未整理時造成非預期快取的風險

【修改】透過 .mp3、.avif、.ipa、.apk、.bmp、.tiff 支援，補足常見音訊、圖片、點陣圖與行動應用安裝檔等靜態資源場景

降低預設策略理解與維護成本

22. 附錄

22.1 關鍵字定義

名詞	定義
Default Cache Rule	平台提供的預設快取規則
TTL	快取有效時間
follow origin	【修改】優先依源站快取控制標頭處理；本次預設為 off

fallback	【修改】源站未返回可用快取標頭時退回平台配置 TTL 的行為；本期非自建節點不強制統一
Cache-Control	源站返回的 HTTP 快取控制 header
Cache-Control directives	【新增】Cache-Control header 內部用於控制快取行為的指令，例如 s-maxage、max-age、no-cache、no-store、private 等
Cache hit ratio	快取命中率
Origin fetch ratio	回源比例
自建節點	【新增】平台自有節點或自有 NGINX 鏈路
非自建節點	【新增】第三方 CDN Provider 或非平台自有快取節點
core-nginx	【新增】自建節點需評估的 NGINX 相關代碼倉
mixed content	【新增】HTTPS 頁面載入 HTTP 資源造成的瀏覽器安全問題；本期不處理
RFC 9111 RFC9111	【新增】HTTP Caching 標準文件，定義 HTTP caches，以及控制 cache 行為或標示可快取回應訊息的相關 header fields；本次作為自建節點 core-nginx 快取行為評估的主要標準參照