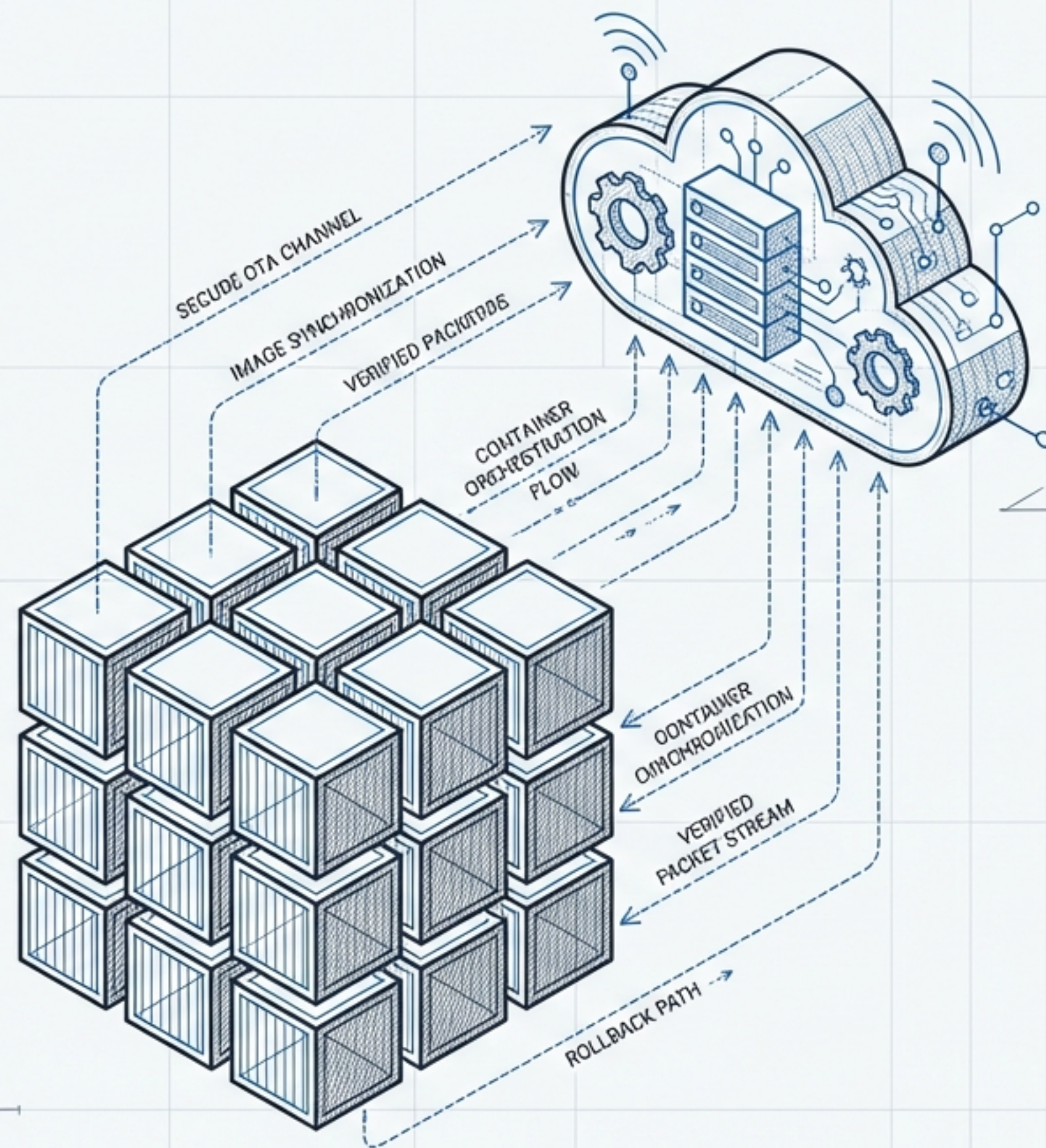


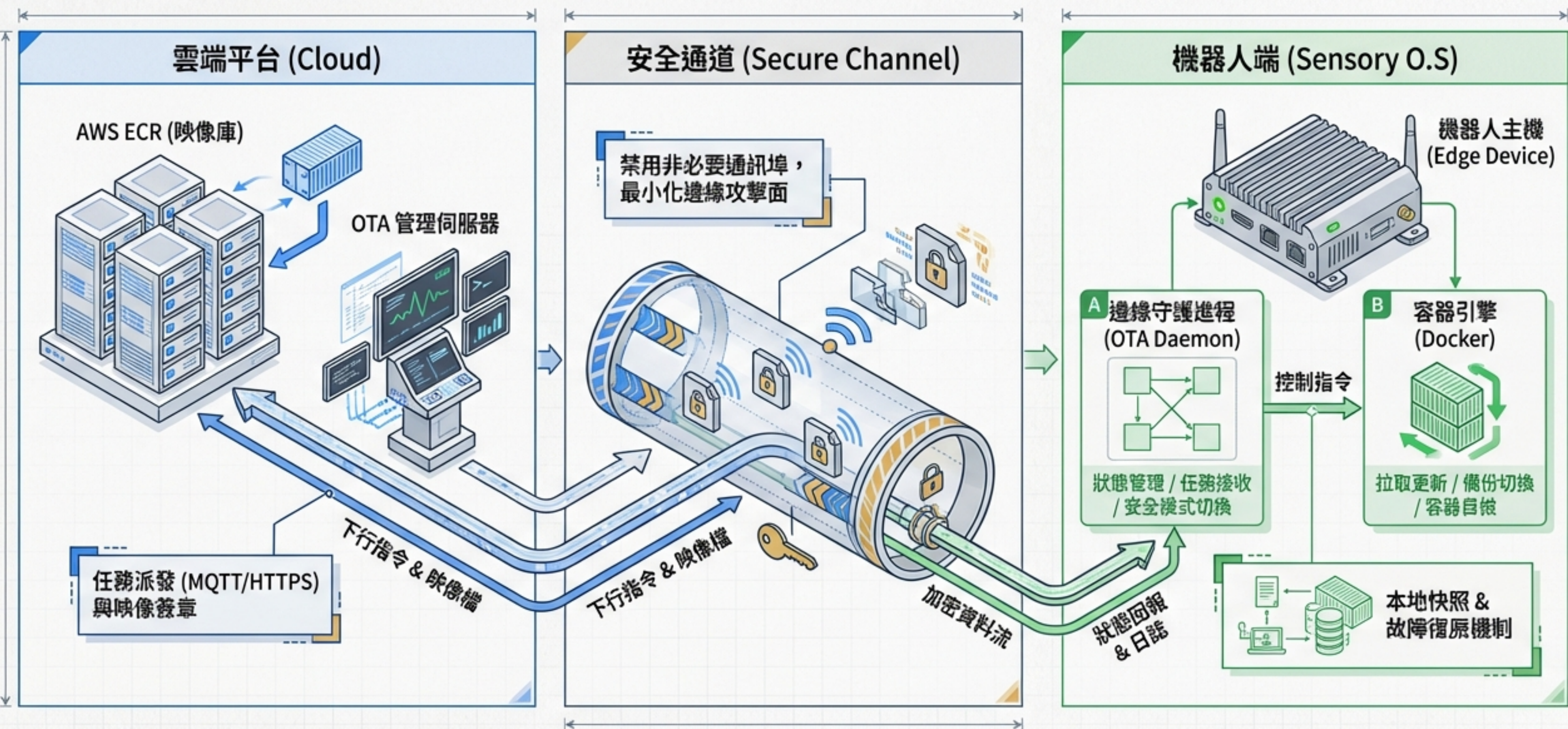
Sensory 0.S OTA 系統架構藍圖

基於 Docker 容器的高可靠無線
更新與回滾實作指南

Target_Audience: R&D / DevOps / System Architects
System_Target: Sensory 0.S Edge Devices
Core_Tech: Docker, AWS ECR, MQTT



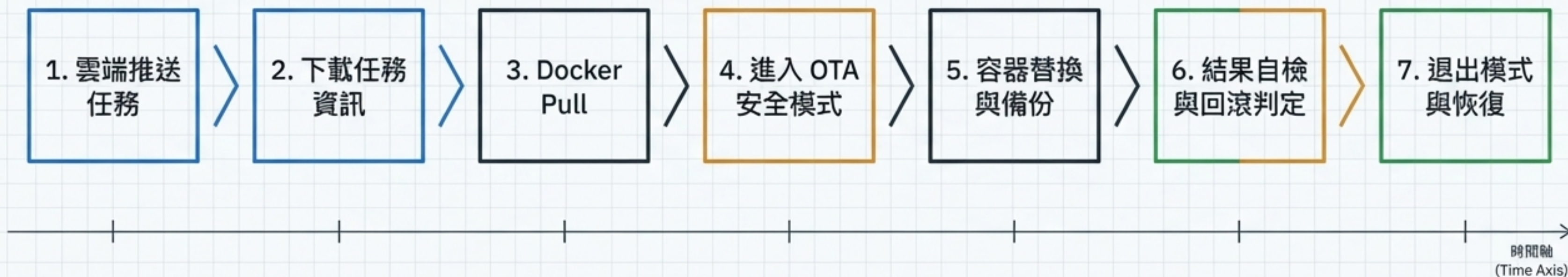
雲端至邊緣的三大核心架構支柱



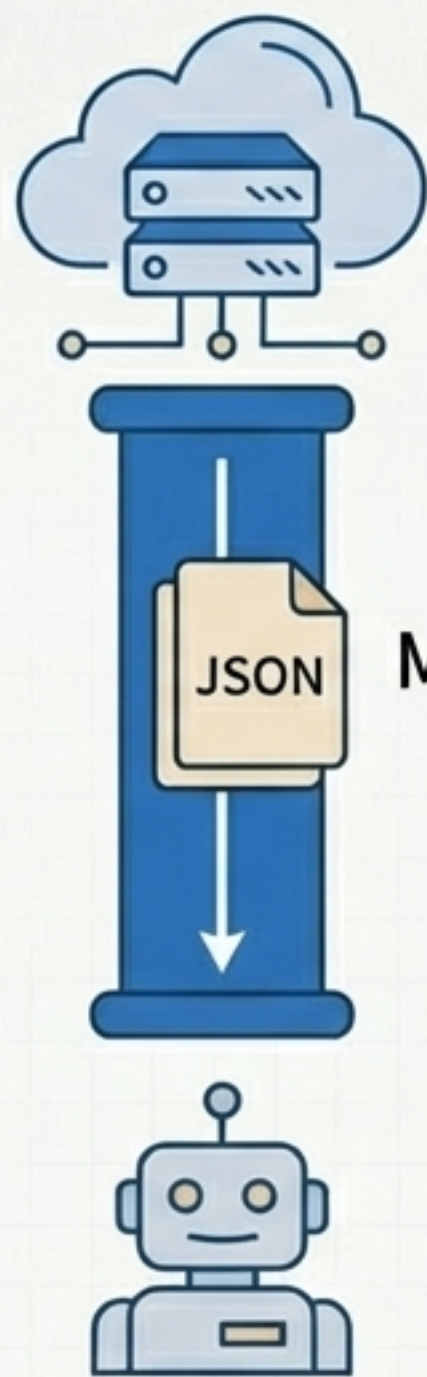
放棄傳統更新機制的技術決策考量

	傳統做法 (Traditional)	Sensory O.S 方案
版本控制 (Version Control)	<p>使用 Tag 標籤 (易發生同名覆蓋， 導致拉取舊版)</p> 	<p>✓ Digest (SHA256) → 確保拉取目標具備全球唯一性 與絕對正確性</p> 
更新包格式 (Payload Format)	<p>差分包 Diff (解包容易出錯， 環境依賴度高)</p> 	<p>✓ 完整映像 + Layer Caching → 確保更新一致性，利用 Docker 快取免下載重複依賴</p> 
容錯與復原 (Fault Tolerance)	<p>失敗需重新刷機 或重新下載舊版 (耗時且風險高)</p> 	<p>✓ 本地映像標記備份 (Tag Backup) → 支援秒級斷線回滾， 無需依賴網路即可恢復</p> 

一次 OTA 更新的生命週期 (The 7-Step Workflow)



建立加密通訊與精確的任務載荷



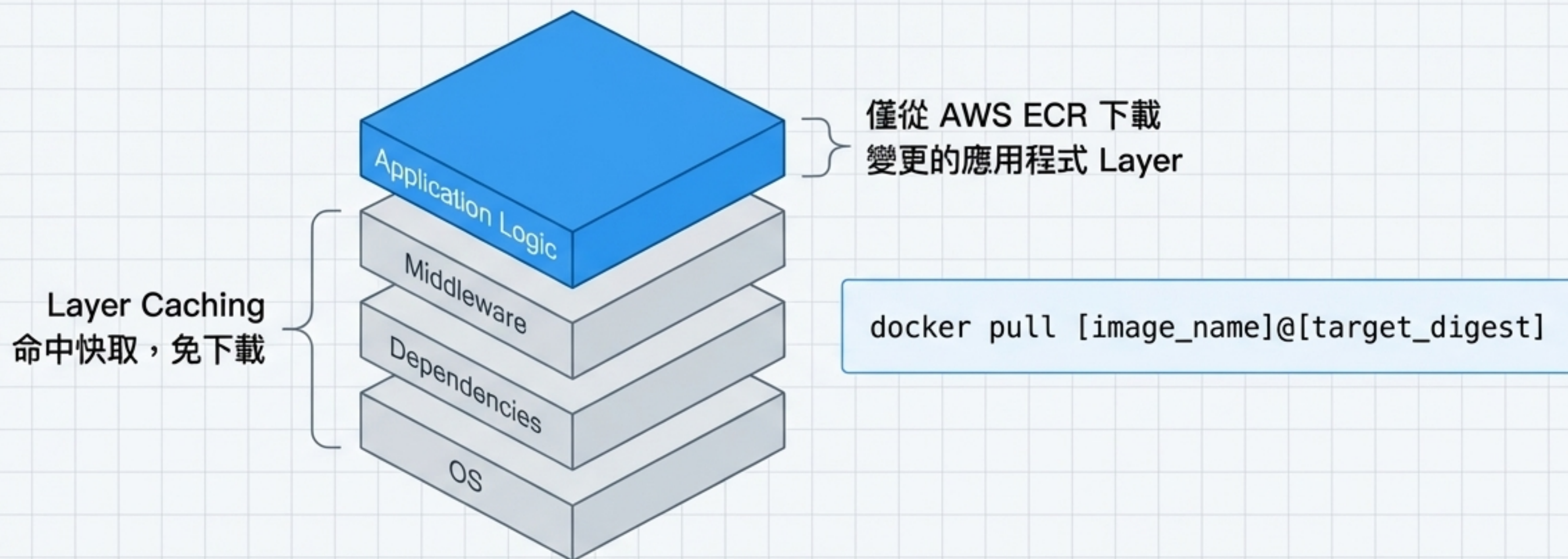
傳輸層安全

邊緣端僅開放必要之連線，阻斷未授權的本機/外部干擾，確保雙向加密通訊。

```
{  
  "task_id": "ota_req_9942",  
  "image_name": "sensory-app",  
  "target_digest": "sha256:7b3a1d...]"  
}
```

嚴格約束：必須包含精確的 SHA256 Digest，嚴禁僅傳送 latest 標籤，防止版本漂移。

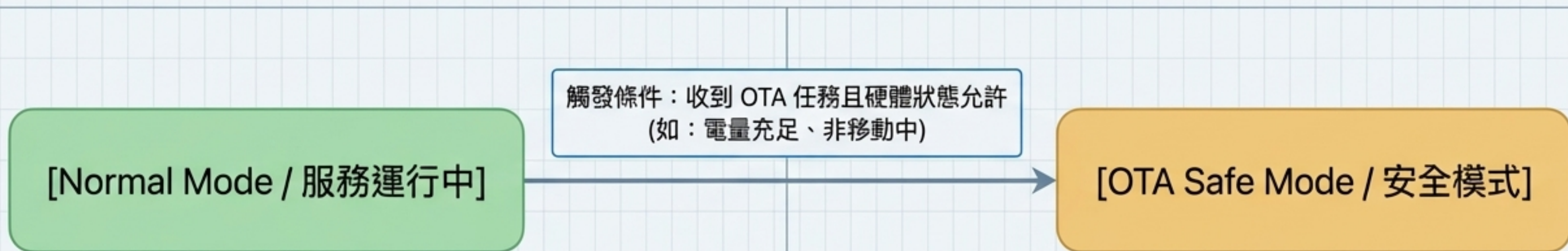
智能映像拉取策略與簽章驗證



映像來源信任 (Integrity & Trust)

整合 AWS Signer 或 Notation CLI。在拉取完成後、執行前，必須進行簽章驗證，確保映像未遭中間人篡改或污染。

狀態機轉移：進入 OTA 安全運行模式



安全停機 (Graceful Shutdown)

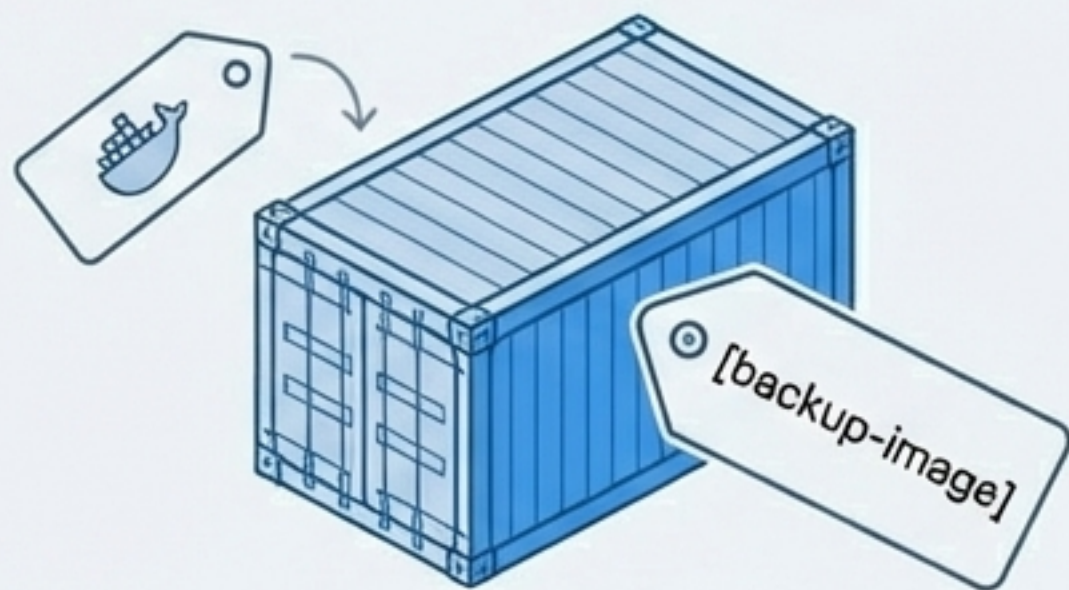
暫停機器人所有移動指令與非必要業務邏輯，釋放硬體控制權，確保系統處於靜止狀態。

資源隔離 (Resource Isolation)

確保 CPU/Memory 資源全數鎖定給 OTA 進程。防止更新時因系統 OOM (Out of Memory) 觸發 OOM Killer 強制中斷 Docker 容器。

容器替換時序與零時差本地備份

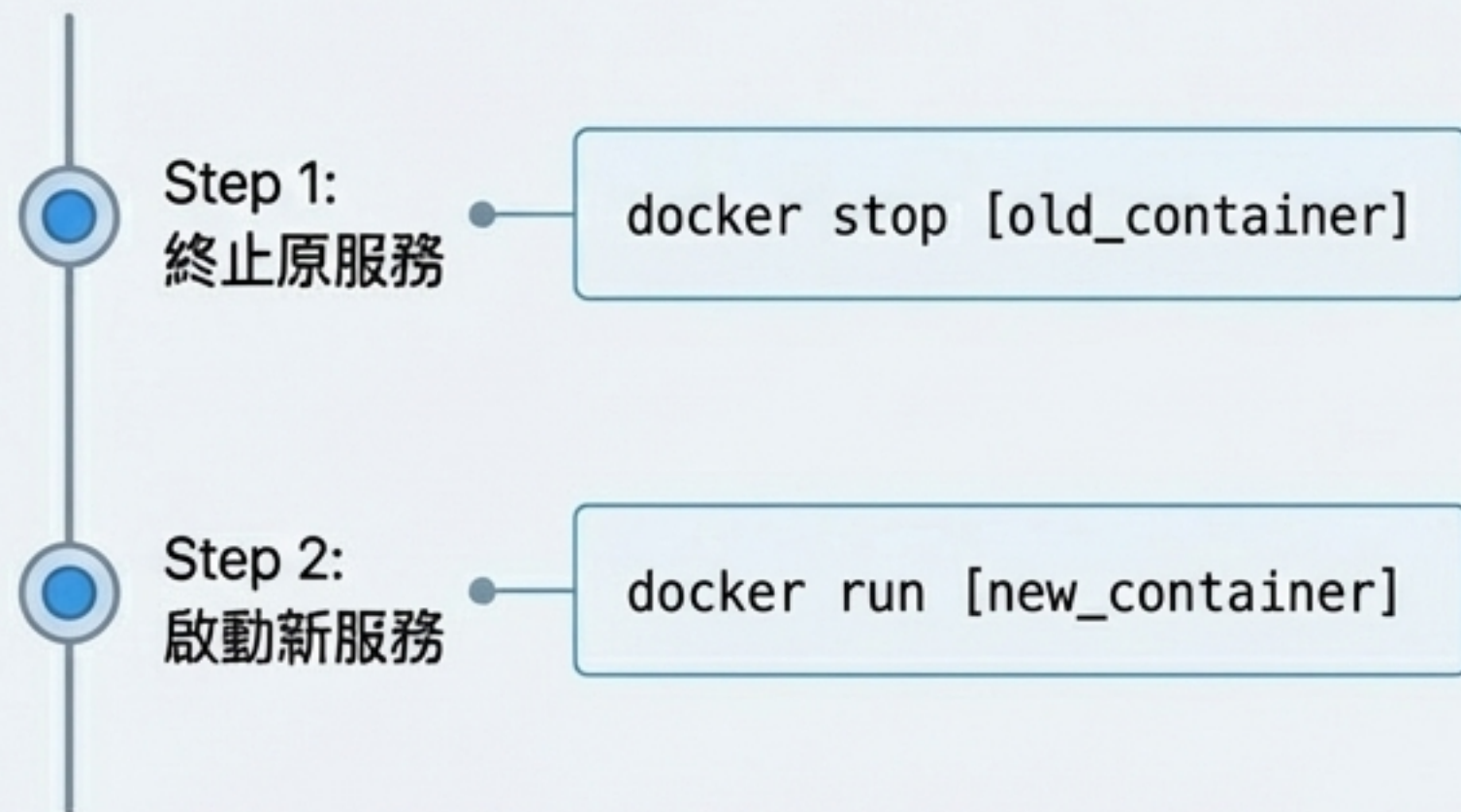
映像備份標記 (Tagging Strategy)



```
docker tag [current-image] [backup-image]
```

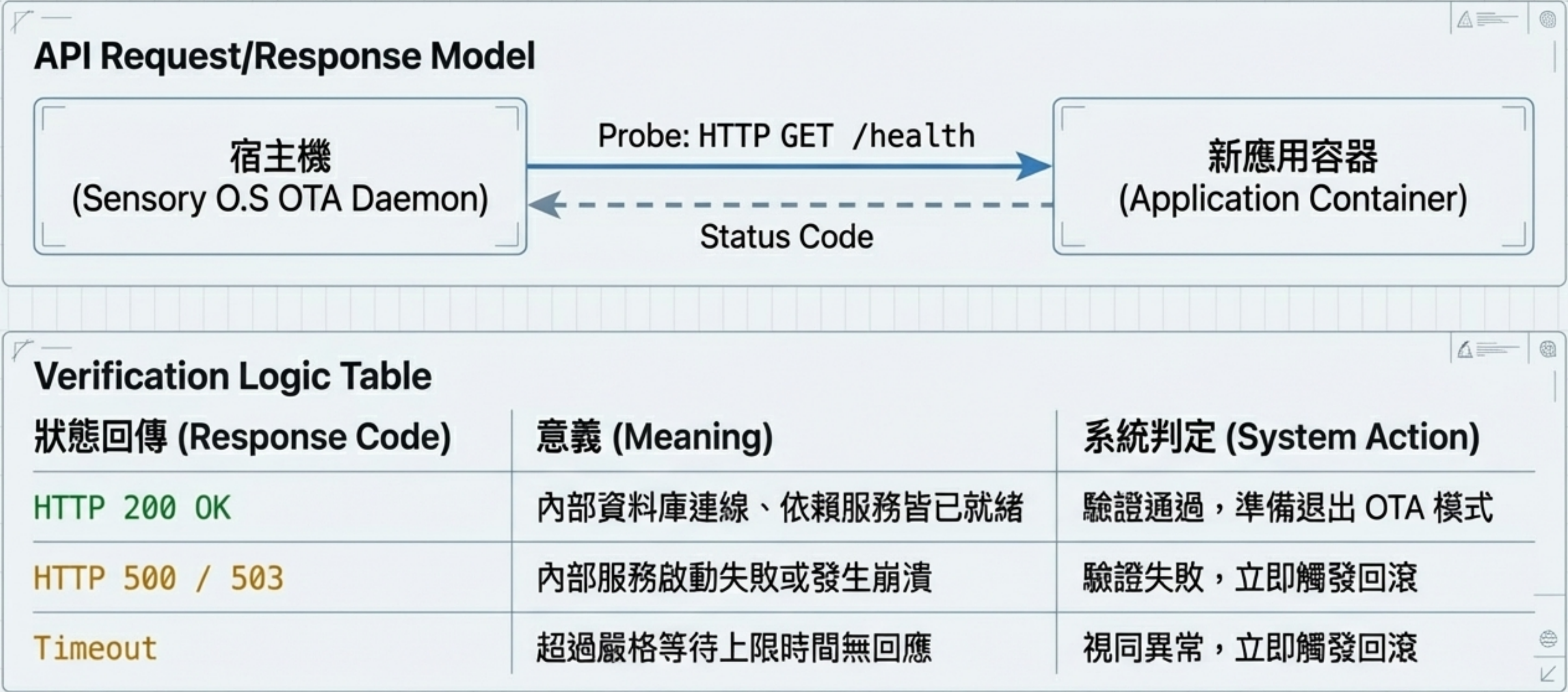
建立本地備份標籤。必須設定防護機制，確保該備份標籤在 OTA 期間絕對不被 docker system prune 意外清理。

啟停時序 (Stop & Start Sequence)

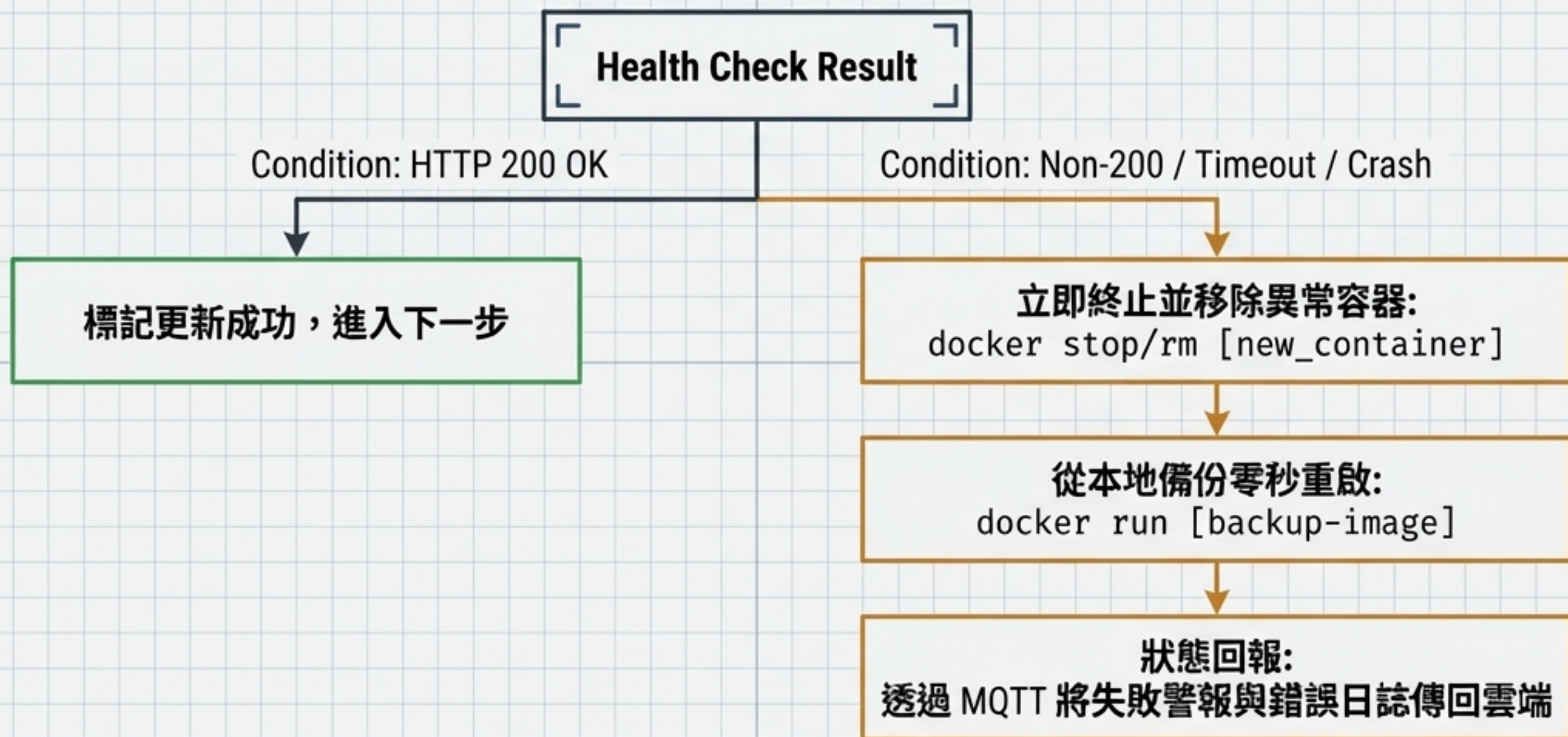


啟動時強制帶入最新下載的 Digest 版本，確保執行檔與預期完全一致。

結果自檢：容器端的自我健康檢查實作約定



故障轉移決策樹與秒級回滾機制



核心優勢：由於備份映像已完整存在於本地，回滾過程無需等待網路重新下載，實現斷線狀態下的秒級復原。

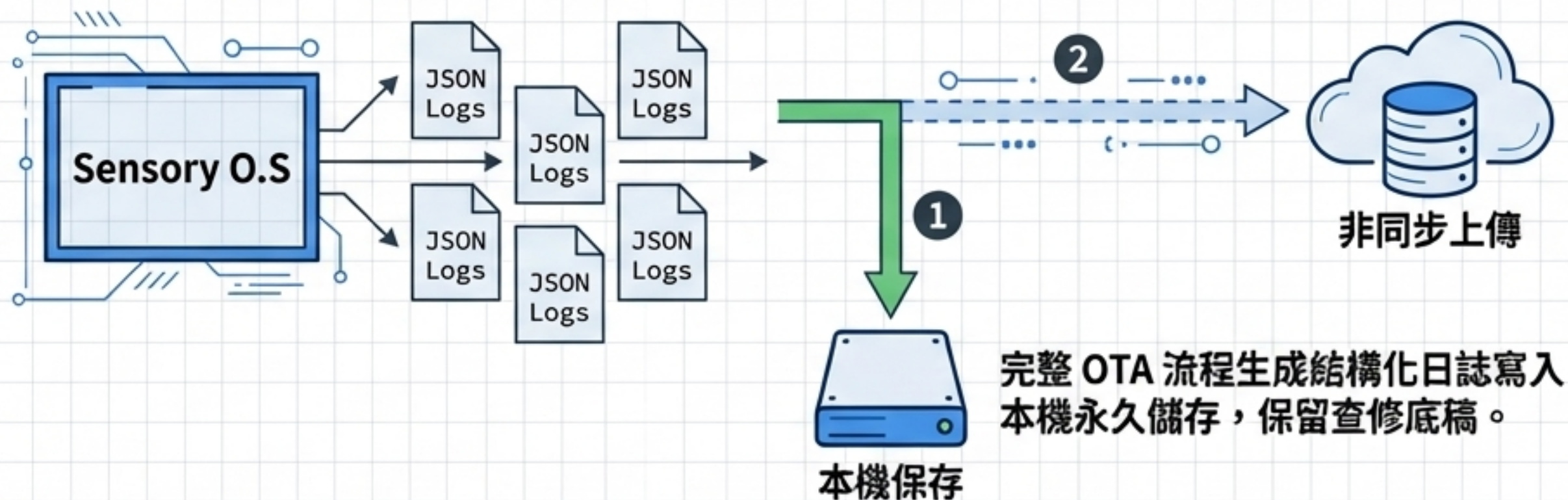
退出安全模式與可觀測性日誌管線



System Recovery (系統恢復)

解除硬體鎖定，退出 OTA 模式，全面恢復機器人對外服務接口與移動控制能力。

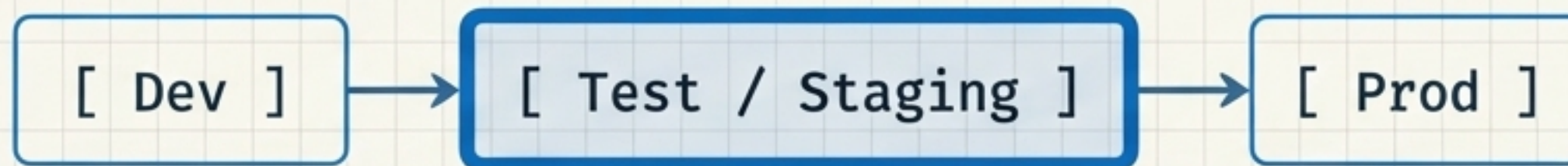
Log Pipeline (可觀測性 Observability)



網路若斷線，日誌將於網路恢復後自動補傳至雲端，確保 Troubleshooting 具備完整上下文，並即時回報最終更新進度。

完整 OTA 流程生成結構化日誌寫入本機永久儲存，保留查修底稿。

部署前品質確保與 極端情境模擬



必須在與產線硬體一致的測試環境
完整走完 OTA 流程

破壞性測試 (Chaos Testing)

☐ 斷電中斷測試

在 docker pull 或容器替換階段強制斷電，重啟後驗證 OTA Daemon 能否優雅恢復或執行回滾。

☐ 網路重連測試

下載期間強制斷網，驗證系統自動重連機制與斷點續傳能力。

☐ 自檢失敗模擬

刻意推送損壞的映像檔，驗證系統能否在時限內精準觸發本地標籤回滾機制。

開發行動藍圖：跨節點模組責任矩陣

開發團隊 (Team)	核心責任與開發目標 (Core Responsibilities)
雲端/後端團隊 (Cloud/Backend)	<ul style="list-style-type: none">- 實作 AWS ECR 權限配置與自動化映像簽署。- 開發 MQTT 任務派發微服務與 OTA 狀態監控儀表板。
Sensory O.S 團隊 (Edge System)	<ul style="list-style-type: none">- 開發 OTA Daemon 狀態機 (安全模式切換邏輯)。- 封裝 Docker API 執行腳本 (pull, tag, run) 與健康檢查探針。- 實作自動回滾腳本。
App 容器團隊 (Application)	<ul style="list-style-type: none">- 優化 Dockerfile，分離依賴層與應用層，極大化 Layer Caching 命中率。- 於應用程式內實作高可靠的 HTTP /health 自檢接口，精確反映內部健康度。