

\$ W6 · 5GC NETWORK FUNCTIONS

/> SMF / UPF

一文一武：SMF 想規則、UPF 做執行 —— 從 PFCP、PDU Session 到 ULCL 與 Traffic Influence

/> 課程大綱

01

角色

SMF / UPF 的分工

02

隧道

N3 · GTP-U · TEID

03

PFCP

N4 : SMF 的遙控通道

04

規則

PDR / FAR / QER / URR

05

會話

PDU Session 與 QoS
Flow

06

建立流程

TS 23.502 逐步拆解

07

UPF 選擇

拓樸、BFS 與 DFS

08

ULCL

上行分流與邊緣

09

流量影響

Traffic Influence

10

GTP5G

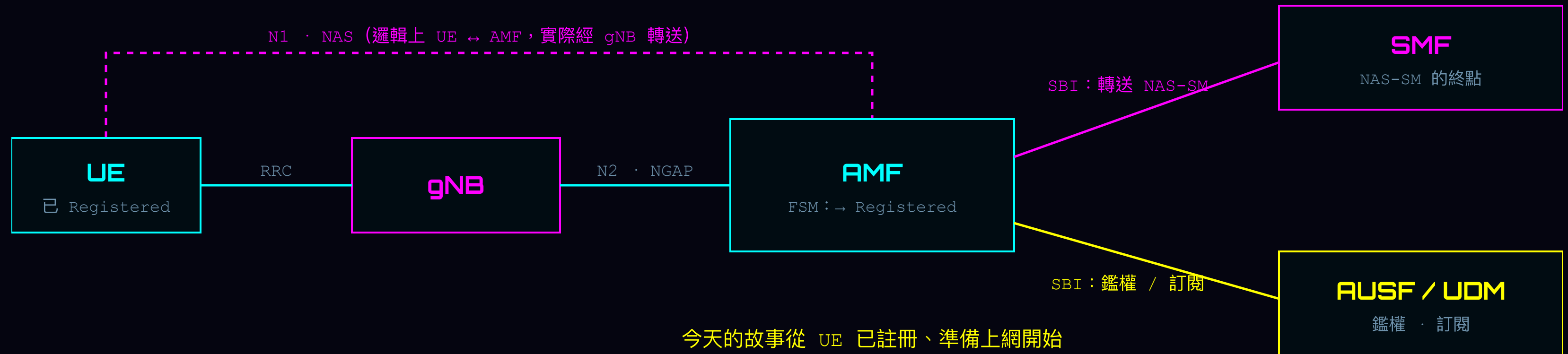
UPF 的資料面

\$ RECAP · W5-2

AMF 回顧

從 AMF 的視角，接到今天的 SMF 與 UPF

➤ 上次的重點：AMF



➤ AMF 只是代辦，真正的主角是？

上次看到的

AMF 視角

AMF 選 SMF、呼叫 CreateSMContext

SMF 回 N1N2MessageTransfer

AMF 轉給 gNB，Session 建好

今天要拆開的

SMF / UPF 視角

SMF 怎麼選 UPF、怎麼分 IP？

PFCP 裡到底下了什麼規則？

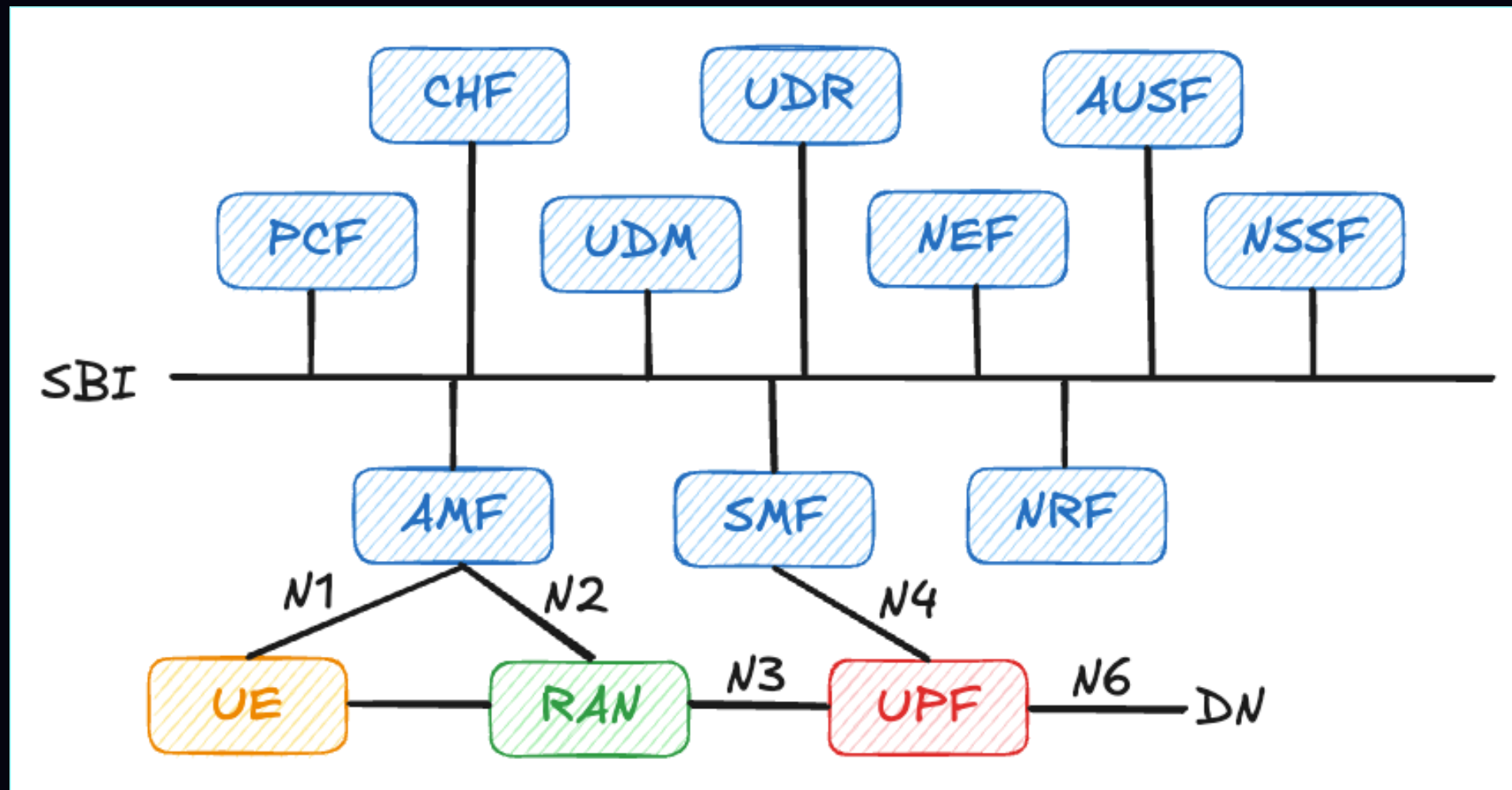
隧道兩端的 TEID 怎麼交換？

\$ PART 01

SMF 與 UPF

5GC 裡「一文一武」的搭檔

➤ SMF / UPF 在 5GC 的位置



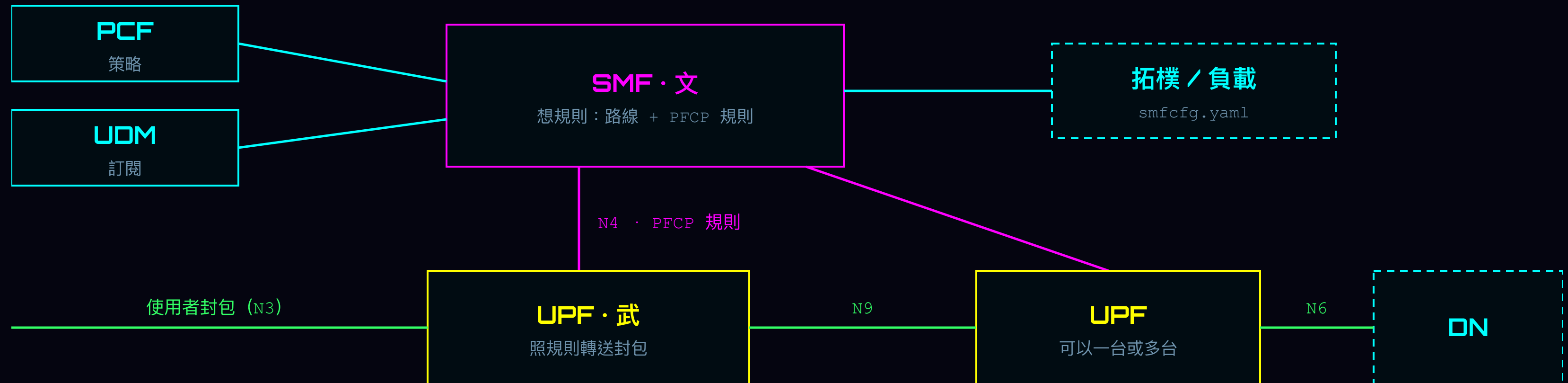
SMF 掛在 SBI 上

UPF 不在 SBI 上

兩者靠 N4 連接

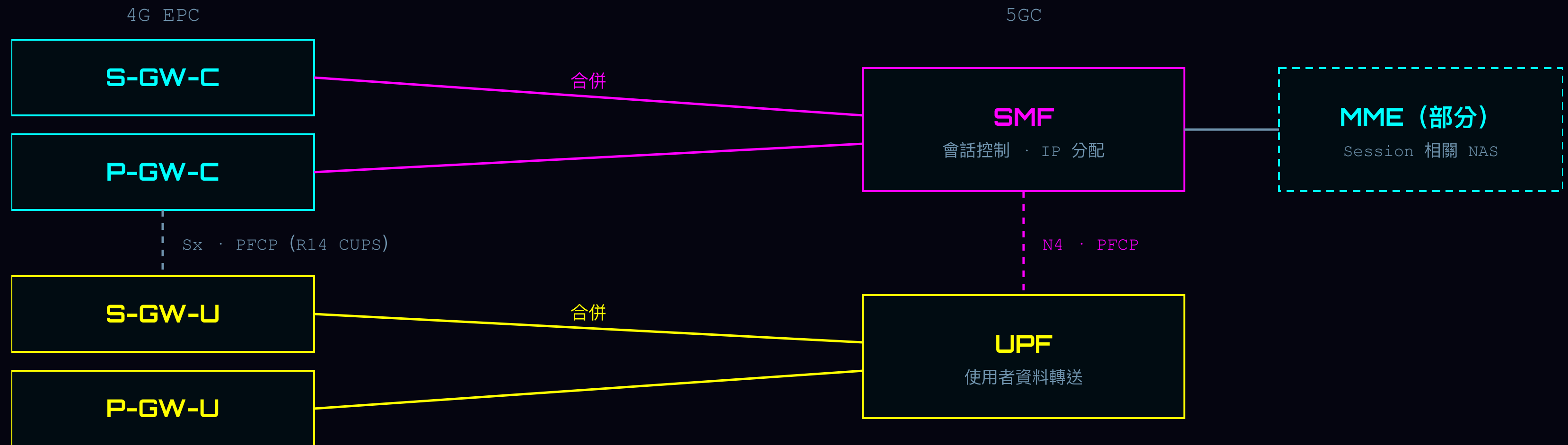
UPF 接 N3、N6

一文一武

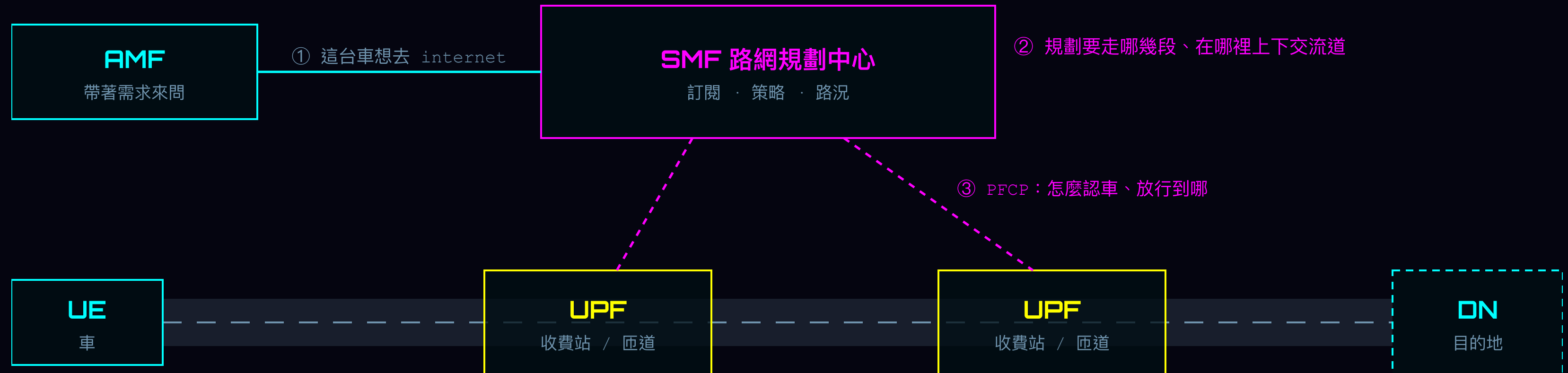


UE 註冊完成後，SMF 協同 PCF 等 NF 建立會話，再透過 N4 把規則下發到一個或多個 UPF。

➤ 從 4G 開道到 SMF / UPF



➤ SMF：高速公路的路網規劃中心



➤ SMF 主要功能 (1/2)

SESSION

PDU Session 管理

建立、修改、刪除 PDU Session

UE IP

IP 位址分配與管理

為 UE 分配 IP，並管理 IP 位址資源池

UPF CONTROL

UPF 選擇與控制

選擇適當的 UPF，透過 N4 控制封包轉送行為

ROUTING

流量路由控制

配置使用者面的流量路由規則

➤ SMF 主要功能 (2/2)

QOS

QoS 控制

實施 QoS Flow 與服務資料流 (SDF) 的映射

CHARGING

計費控制

向 CHF 提供計費相關資訊

POLICY

策略執行

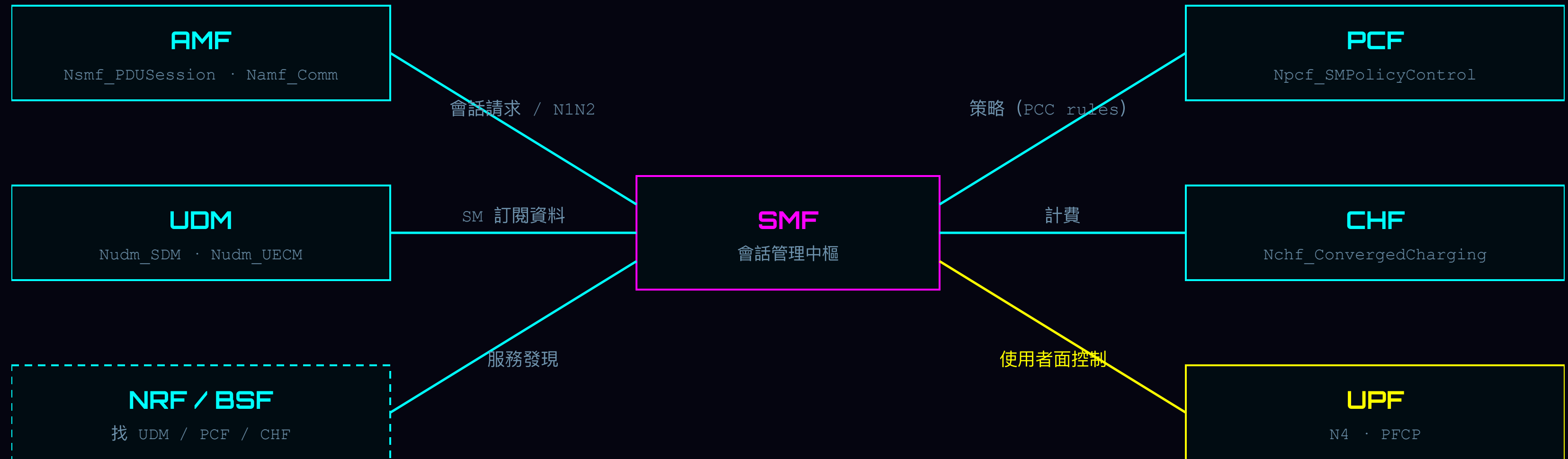
執行來自 PCF 的策略決定

ROAMING

漫遊

漫遊場景中 V-SMF 與 Home SMF 協作

➤ SMF 的協作夥伴



➤ UPF 主要功能 (1/2)

FORWARDING

封包路由與轉送

根據 SMF 的控制規則轉送使用者資料封包

DPI

封包檢測

深度封包檢測，支援服務資料流的識別

QOS

QoS 處理

流量整型、流量管制與封包標記

USAGE REPORT

使用量報告

向 SMF 回報用量，支援計費與策略

UPF 是唯一處理使用者資料流量的核心網路元件，相當於 4G 的 S-GW-U + P-GW-U。

➤ UPF 主要功能 (2/2)

LOCAL BREAKOUT

上行流量分流

本地分流，把特定流量就近處理

BUFFERING

下行封包緩衝

UE 閒置時暫存下行封包，等 Paging 完成

MULTI-PATH

多路徑

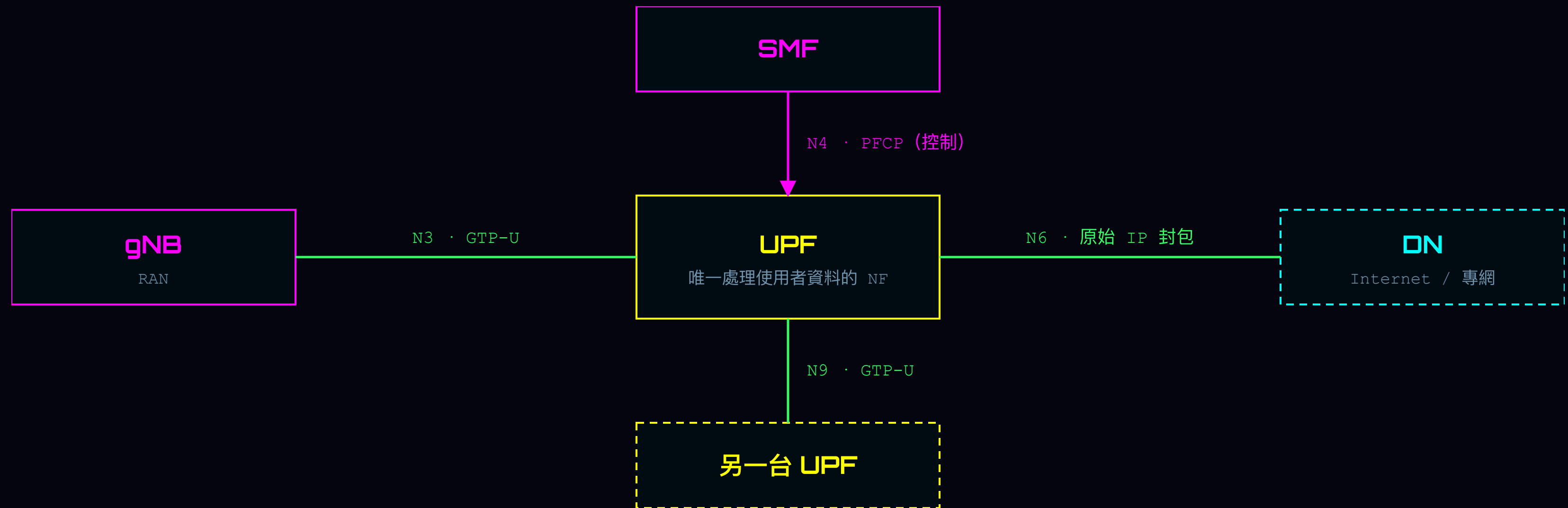
支援 PDU Session 的多路徑傳輸 (如 MA PDU)

EDGE

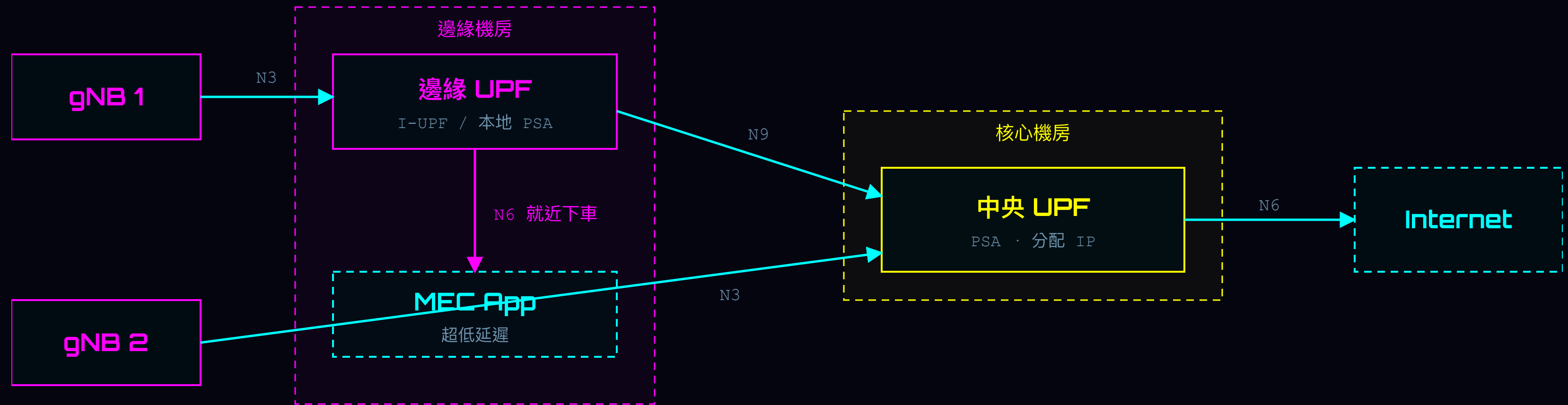
邊緣運算

可部署在網路邊緣，降低延遲

➤ UPF 的四個介面



/> UPF 的分散式部署



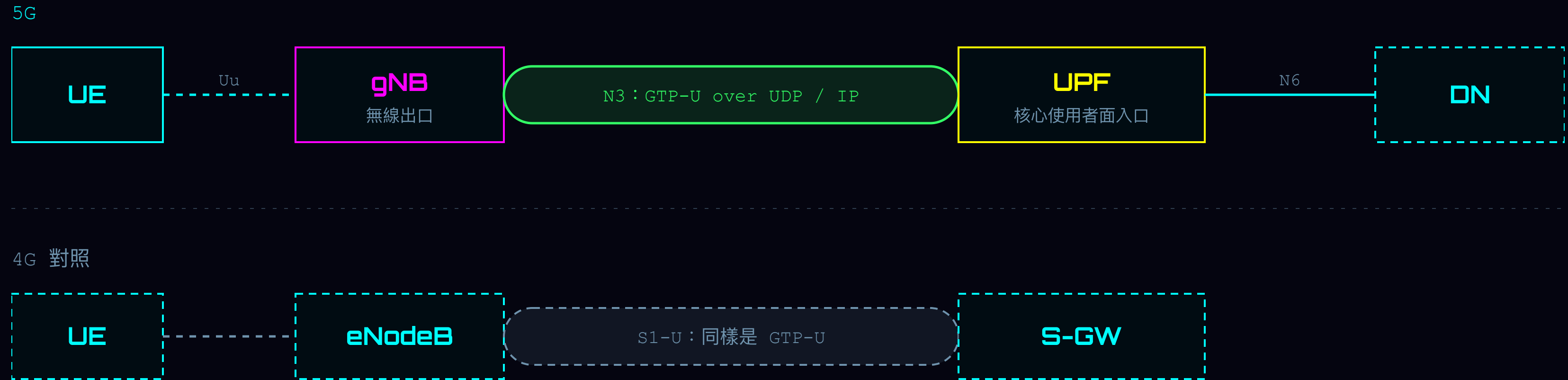
SMF 集中、UPF 分散：CUPS 帶來的彈性，也是 ULCL 與 Traffic Influence 的基礎。

\$ PART 02

N3 與 GTP-U

使用者資料怎麼從 gNB 走進核心網

➤ N3：RAN 與 UPF 之間的使用者面



N3 是邏輯介面名稱；可以把它理解成「5G 版的 S1-U」，對端從 S-GW 換成 UPF。

➤ GTP-U：在 IP 外面再包一層

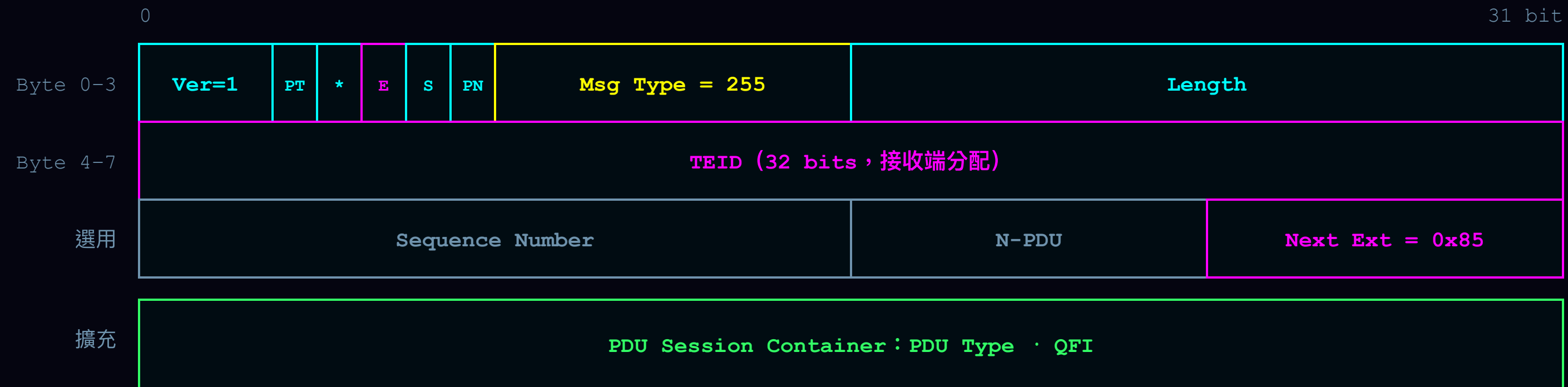


外層 IP：gNB ↔ UPF 的傳輸網位址

TEID：標識「屬於哪一條隧道 / 哪一條 PDU Session」

內層：UE 自己的 IP 封包，完全不被改動

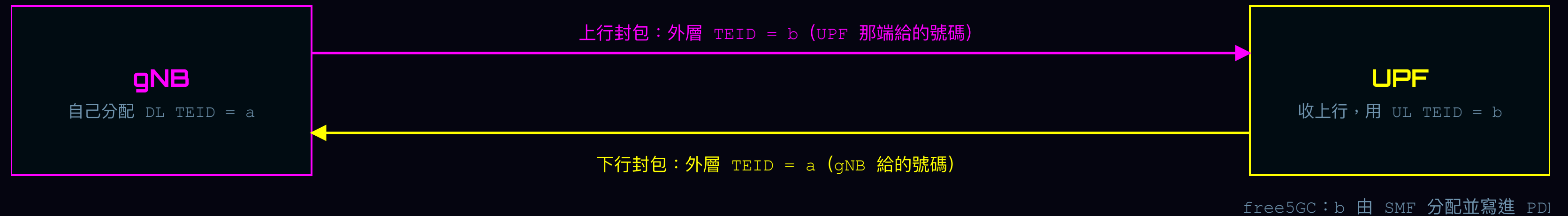
➤ GTP-U HEADER 欄位



Msg Type 255 = G-PDU (使用者資料)；其他如 Echo、End Marker

E = 1 時帶擴充標頭；5G 用 PDU Session Container 攜帶 QFI

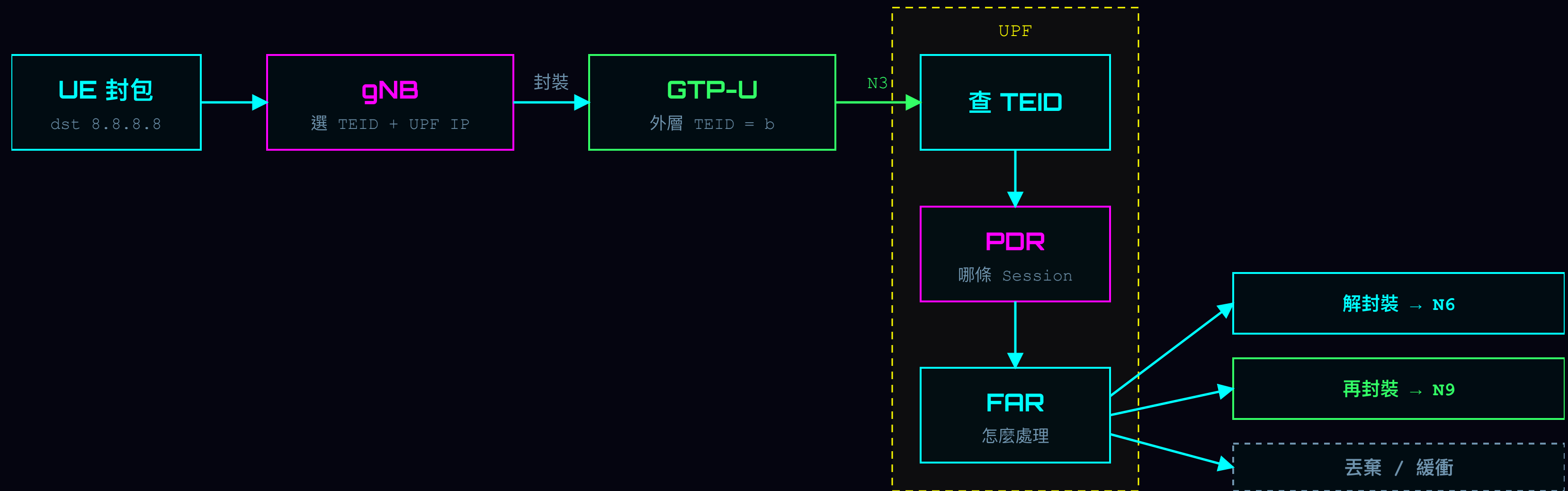
➤ TEID：誰收封包，誰給號碼



原則：誰收封包，誰給號碼

所謂 GTP-U 隧道，其實就是兩端事先協定好的一組標頭資訊：IP + TEID。

➤ GNB 與 UPF 如何處理 GTP-U



➤ 除錯：用 WIRESHARK 看三種協定

FILTER	PORT	看什麼
ngap	SCTP 38412	ResourceSetupRequest / Response 裡的 TEID
pfcf	UDP 8805	SMF 下了哪些 PDR / FAR / QER
gtp	UDP 2152	外層 TEID、QFI、內層 UE 封包

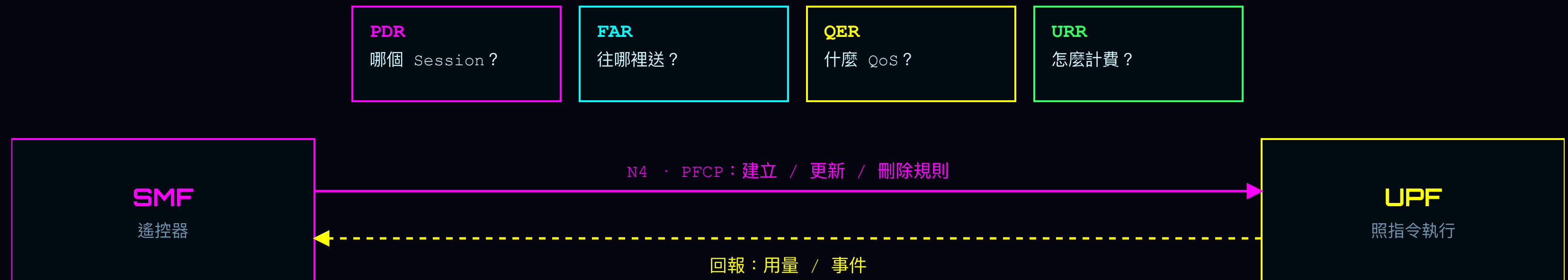
三者 TEID 對得起來，就能確認某條流量為什麼走到特定 UPF 介面。

\$ PART 03

N4 與 PFCP

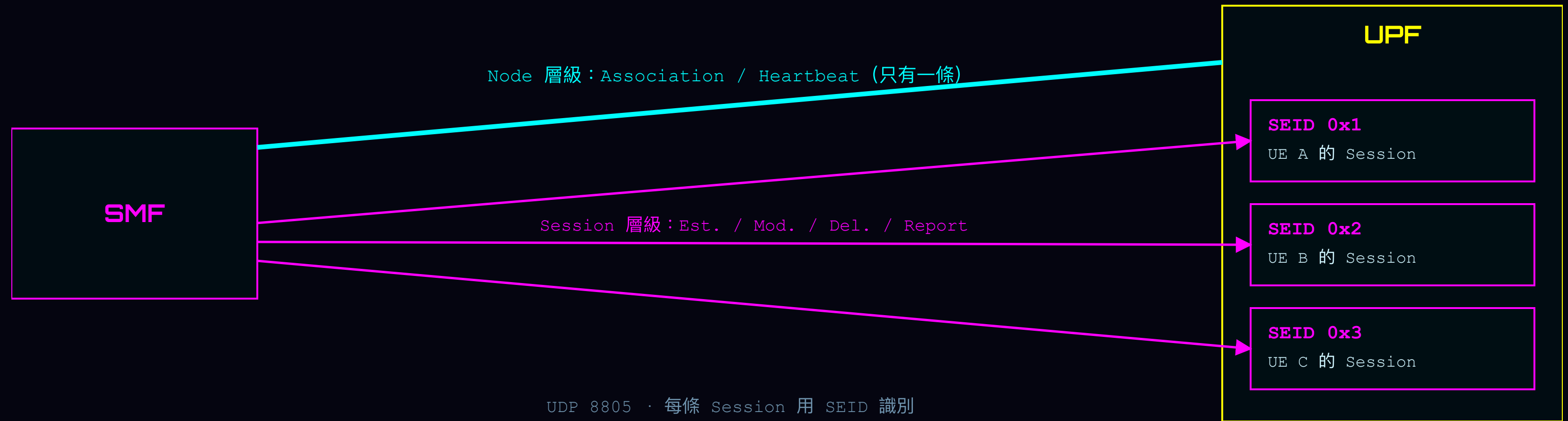
SMF 告訴 UPF 「怎麼處理封包」的語言

➤ N4 : SMF 的遙控通道



N4 上跑的協定是 PCF (Packet Forwarding Control Protocol)，定義在 TS 29.244。

/> PFCP 的兩種訊息

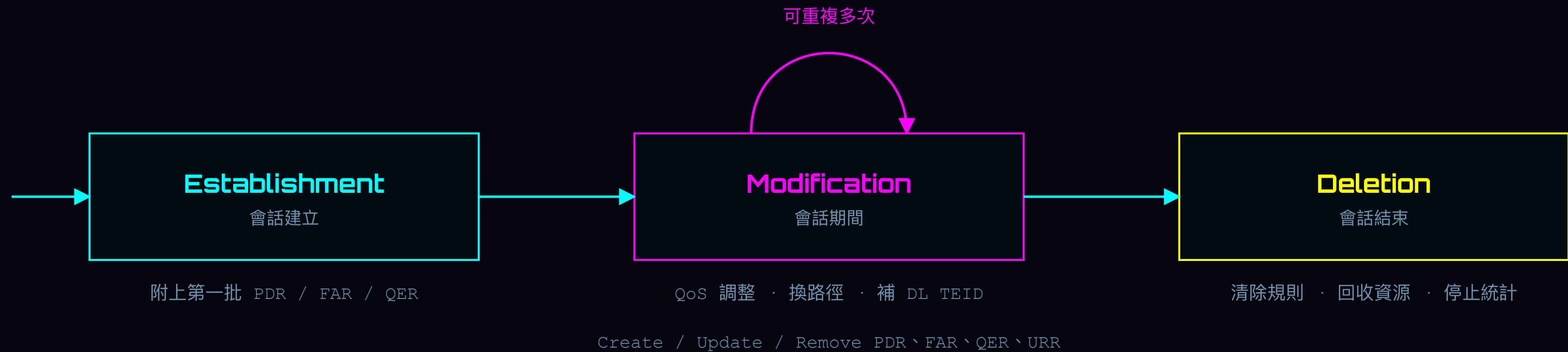


➤ ASSOCIATION：先建交，再辦事

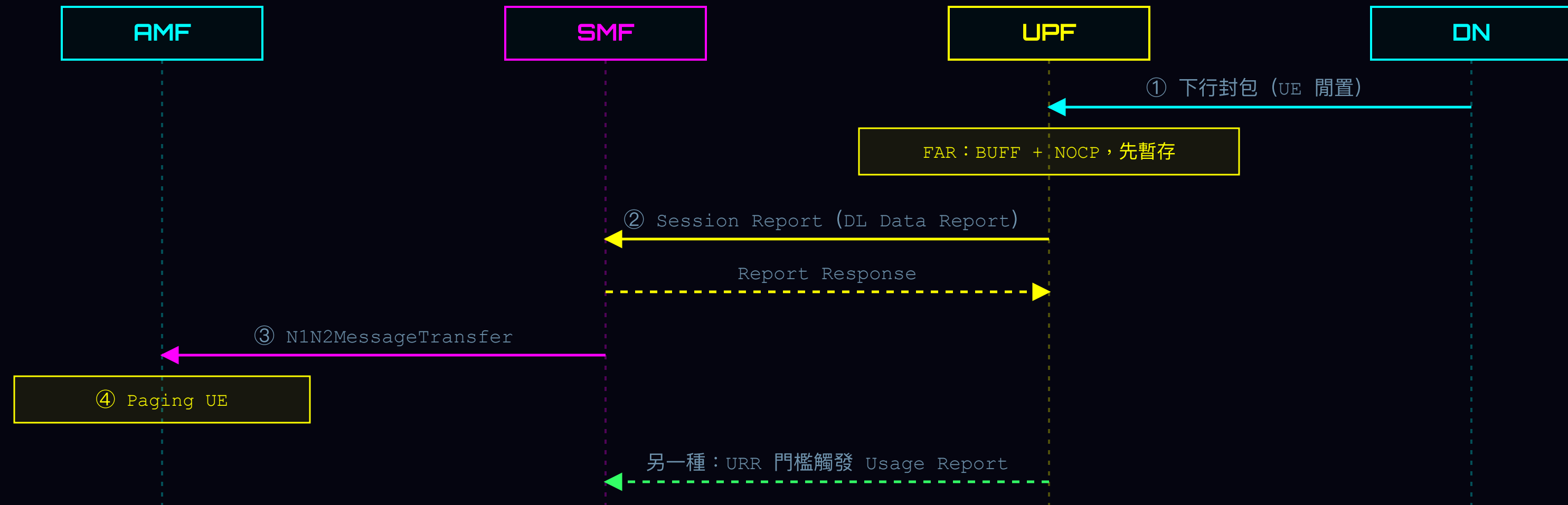


這就是 free5GC 要先啟動 UPF 的原因：SMF 一啟動就會主動發起 Association。

➤ 一條 SESSION 在 PCF 上的一生



➤ SESSION REPORT : UPF 主動回報



／> 常見 PFCP 訊息一覽

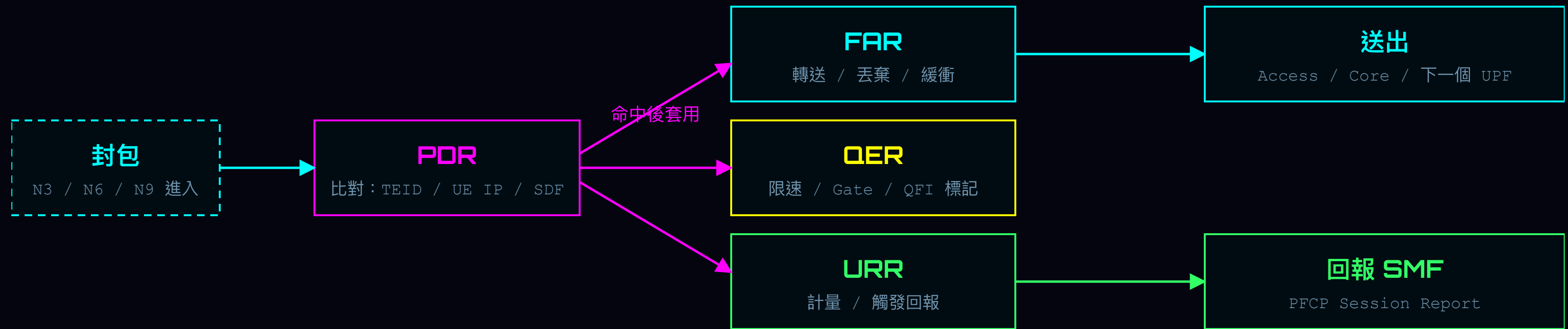
訊息	方向	用途
Association Setup	SMF → UPF	建立節點關係、交換能力
Heartbeat	雙向	偵測對端存活
Session Establishment	SMF → UPF	建 Session，Create PDR / FAR / QER / URR
Session Modification	SMF → UPF	Create / Update / Remove 規則
Session Deletion	SMF → UPF	刪除 Session，回傳最終用量
Session Report	UPF → SMF	用量回報、下行資料通知

\$ PART 04

／> 資料平面的規則

PDR · FAR · QER · URR : UPF 看得懂的指令

／> 規則鏈：從比對到動作

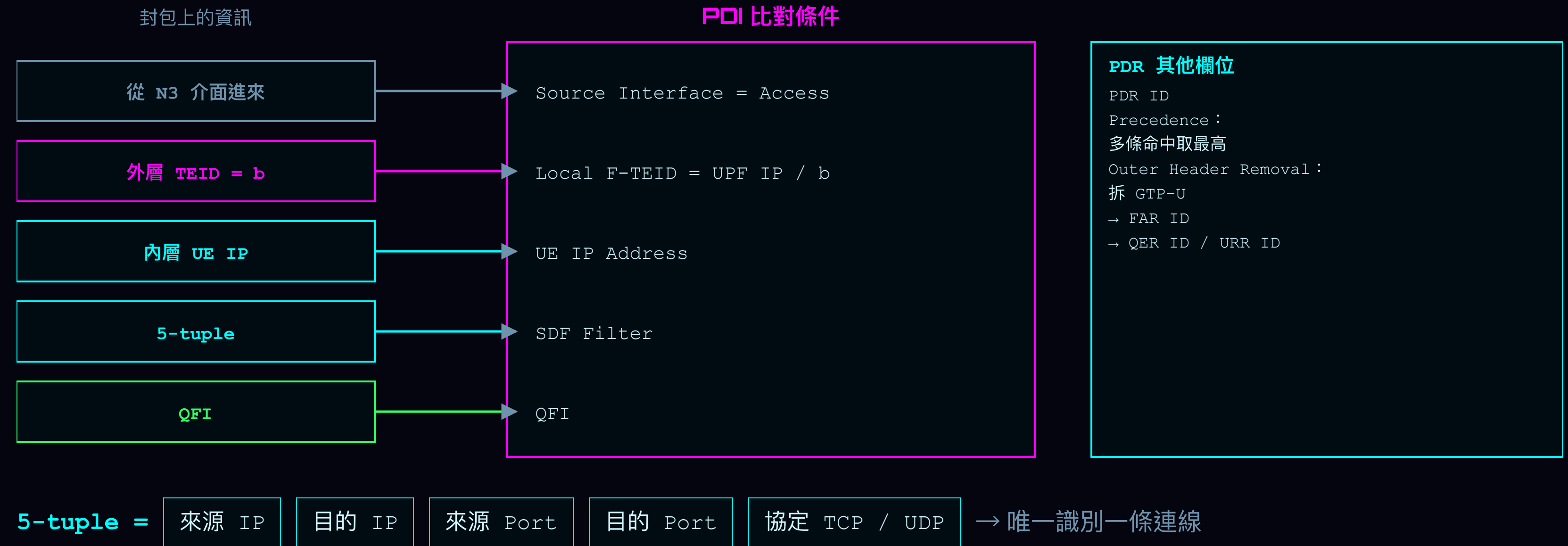


PDR 是入口：先決定「這是誰的封包」，再由它指到的 FAR / QER / URR 決定「怎麼處理」。

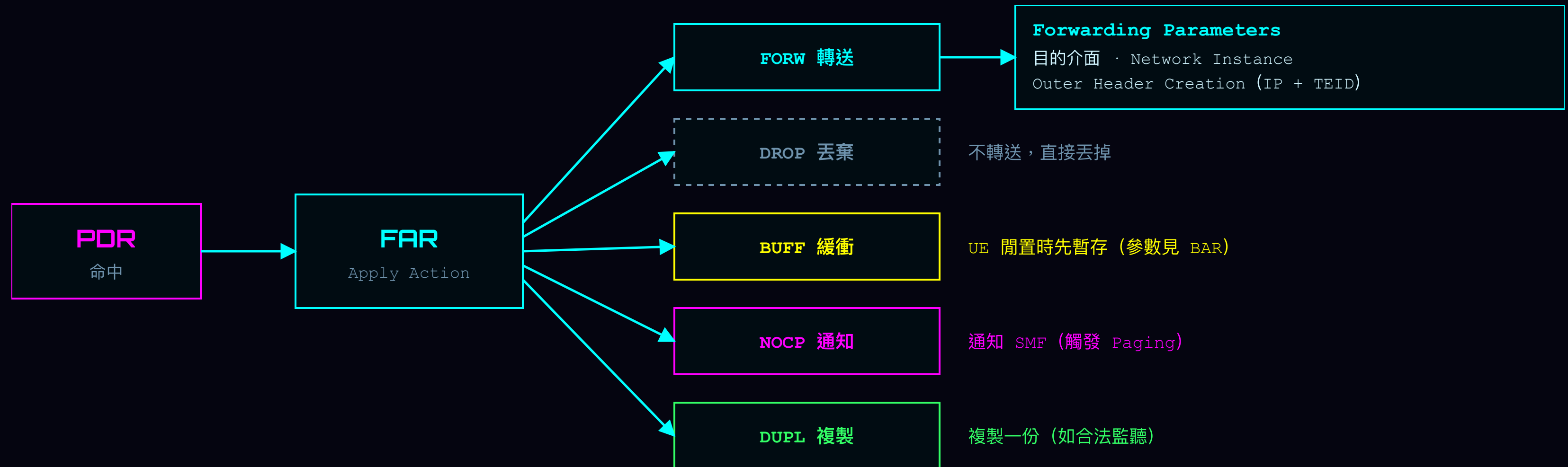
➤ 四種規則

規則	全名	負責
PDR	Packet Detection Rule	哪些封包屬於哪條 Session / 流量
FAR	Forwarding Action Rule	命中後怎麼處理：轉送、丟棄、緩衝
QER	QoS Enforcement Rule	頻寬、Gate、QFI 標記
URR	Usage Reporting Rule	統計流量與事件並回報
BAR	Buffering Action Rule	補充：下行緩衝的參數

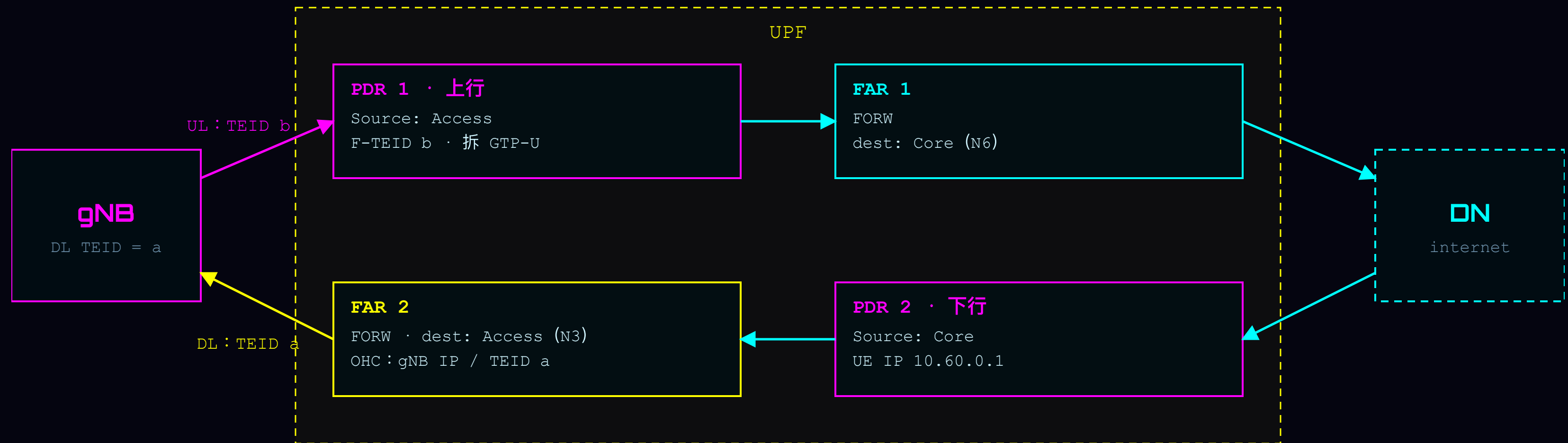
/> PDR：封包偵測規則



/> FAR：轉送動作規則



／> 範例：一條 SESSION 的上下行規則



OHC = Outer Header Creation：下行要「重新包上」指向 gNB 的 GTP-U 標頭。

/> QER 與 URR

QER : QoS Enforcement

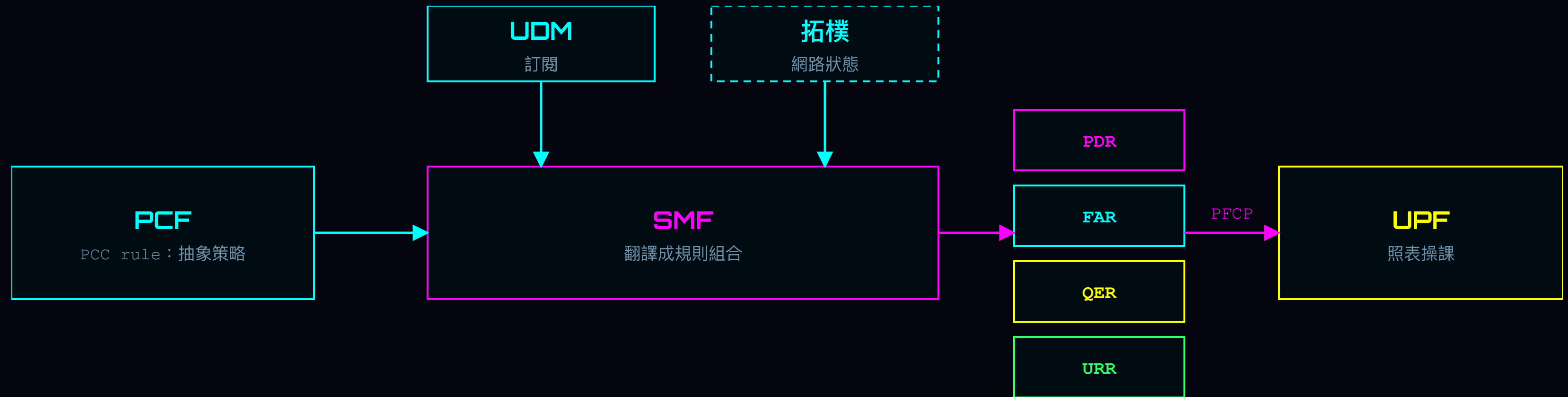


URR : Usage Reporting



例：額度用完 → Gate 關閉；用量達門檻 → 回報 SMF

／> 從抽象策略到具體規則



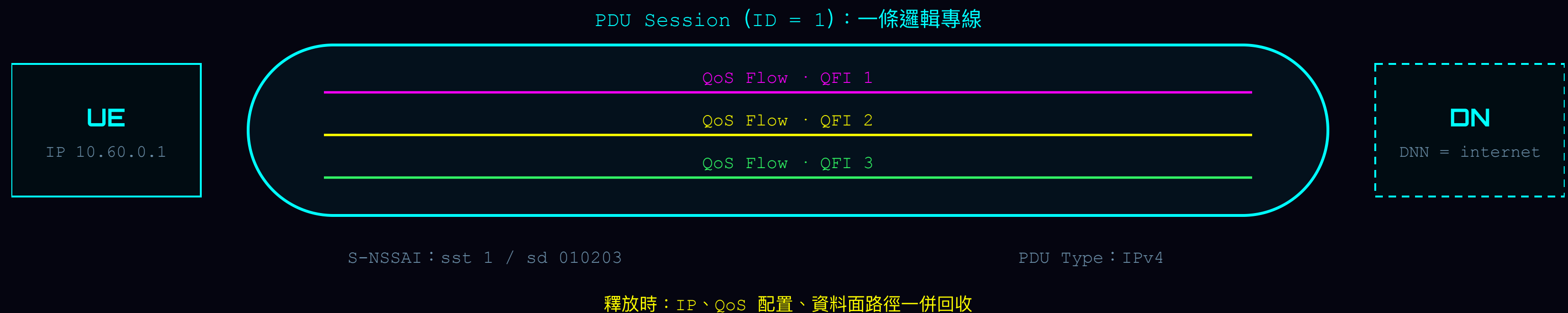
一條「上網連線」可能拆成多組規則：一般數據、VoNR 語音、企業專網各有自己的 PDR / QER。

\$ PART 05

PDU SESSION

UE 與資料網路之間的一條邏輯專線

➤ PDU SESSION 是什麼？

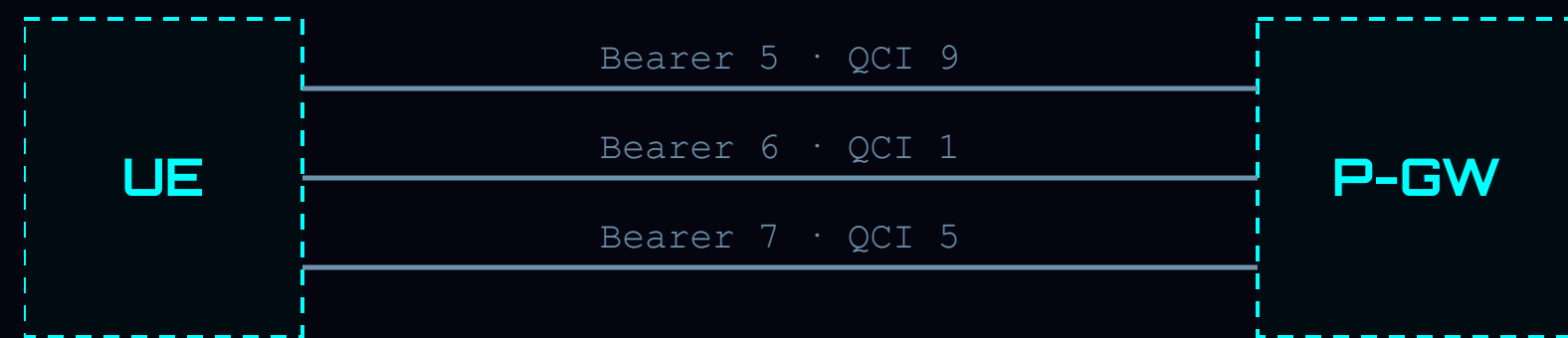


PDU SESSION TYPE

TYPE	內容	FREE5GC
IPv4	最常見的上網會話	✓ 官方支援
IPv6 / IPv4v6	IPv6 前綴由 Router Advertisement 提供	—
Ethernet	承載乙太網路訊框，工業 / TSN 場景	—
Unstructured	不解析內容，例如 IoT 專用協定	—

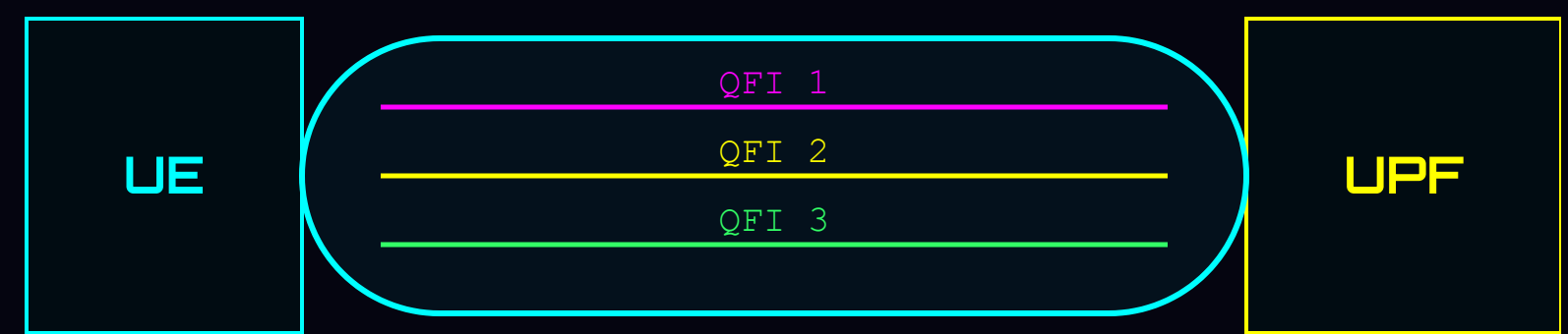
➤ PDU SESSION VS EPS BEARER

4G : EPS Bearer



每個 Bearer 一條 GTP 隧道

5G : PDU Session



典型：一條 N3 隧道 + 多個 QoS Flow

每條會話綁定一個 S-NSSAI

5G 新增 QoS 只要在同一條會話加 QoS Flow，不必另建 Dedicated Bearer。

／> 一條會話，多個 QOS FLOW



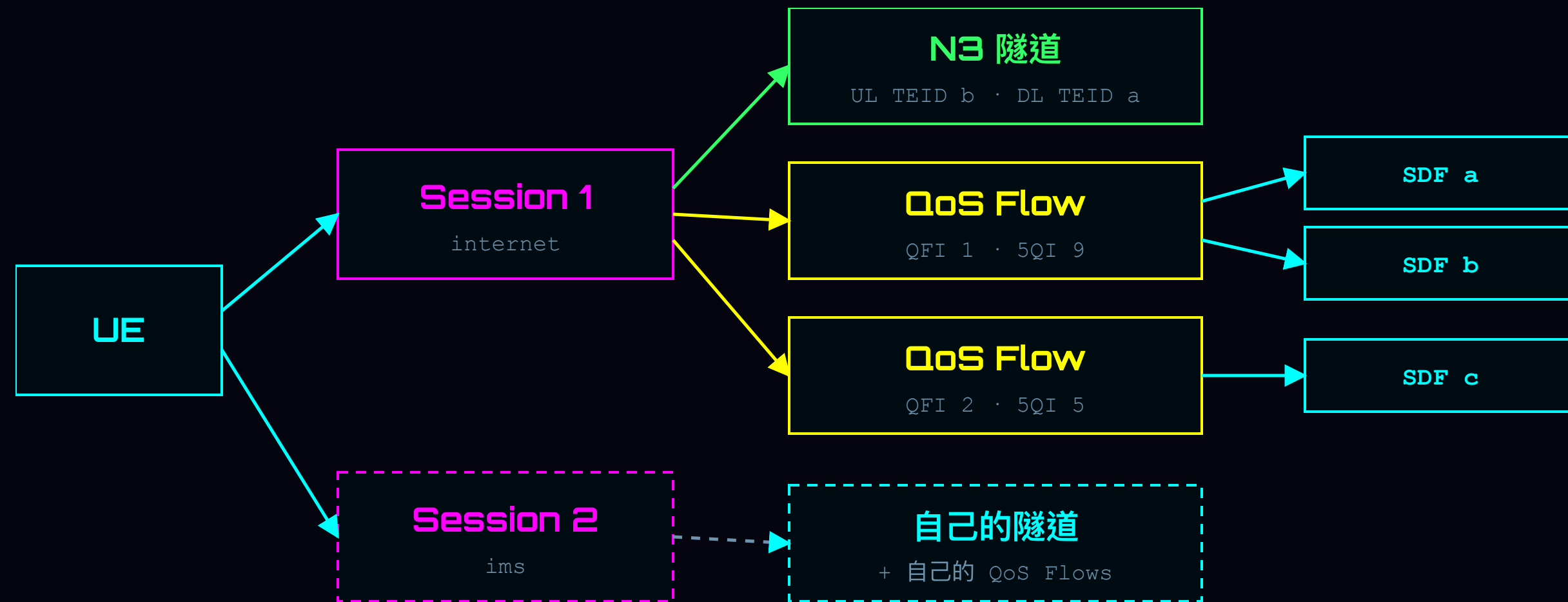
UE 用 QoS Rule、UPF 用 PDR 把 SDF 對應到 QFI；gNB 再把 QoS Flow 對應到無線承載（DRB）。

／> SESSION、隧道、QOS FLOW 的關係



Session 是邏輯連線；隧道是它在 N3 / N9 上的載體；QoS Flow 是 Session 內的品質分類，在隧道裡用 QFI 區分。

➤ 從屬與數量關係



數量關係

UE : Session = 1 : N

Session : 隧道 = 1 : N

(典型：每段 1 條)

Session : QoS Flow = 1 : N

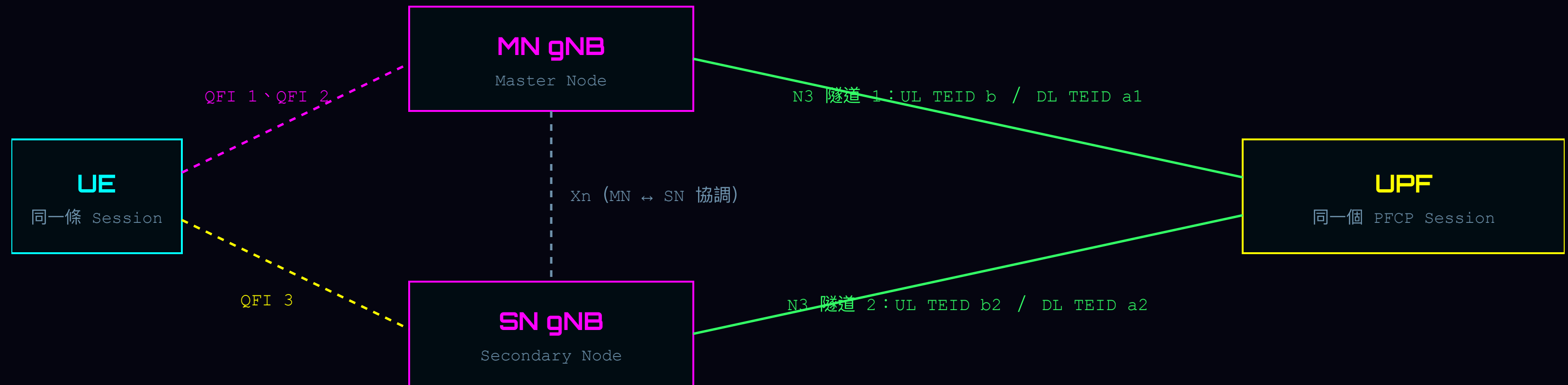
(QFI 6 bits: 0-63)

QoS Flow : SDF = 1 : N

QoS Flow : DRB = N : 1
(gNB 的 SDAP 決定)

例如 NR-DC 時
會有兩條 N3 隧道

➤ 一條 SESSION 多條隧道：NR-DC



建立時以 Additional UL / DL NG-U TNL Information 交換第二組 TEID

free5GC SMF: Tunnel + DCTunnel, 兩條隧道各有自己的 PDR / FAR

雙連結時，同一條 Session 的 QoS Flow 分給兩個 gNB，各自有一條 N3 隧道。

／> 一個上行封包上的三種識別碼



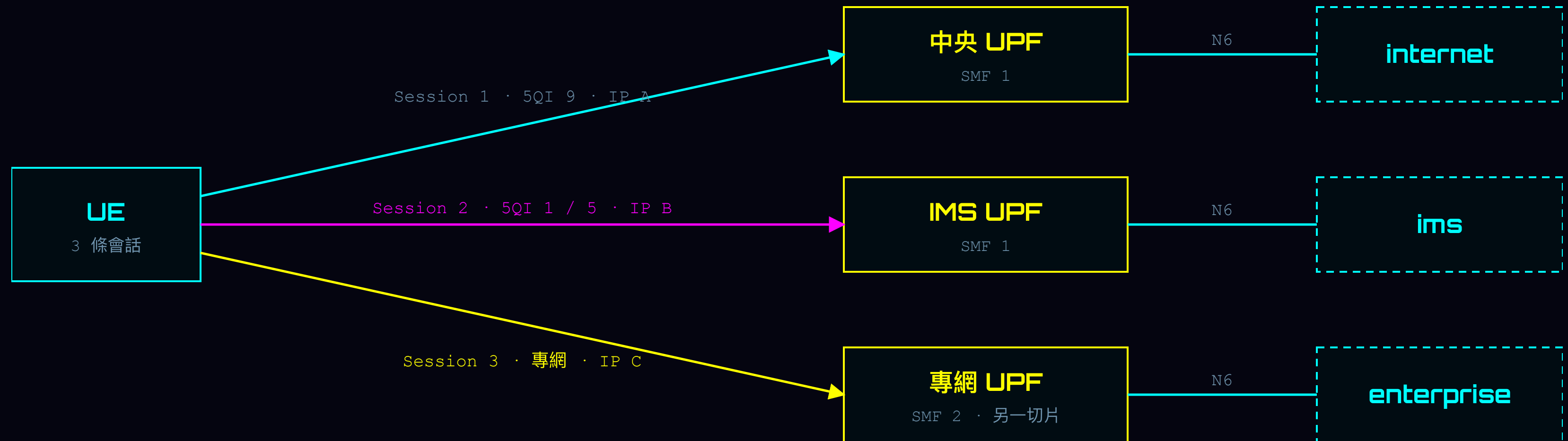
下行反過來：UPF 用 PDR 比對 SDF 決定 QFI，外層填 gNB 的 DL TEID

常見 5QI

5QI	類型	延遲預算	典型服務
1	GBR	100 ms	語音（VoNR）
5	Non-GBR	100 ms	IMS 信令
9	Non-GBR	300 ms	一般上網（預設）
82	Delay-critical GBR	10 ms	離散自動化

完整對照表在 TS 23.501 Table 5.7.4-1。

➤ 一支 UE，多條 PDU SESSION



\$ PART 06

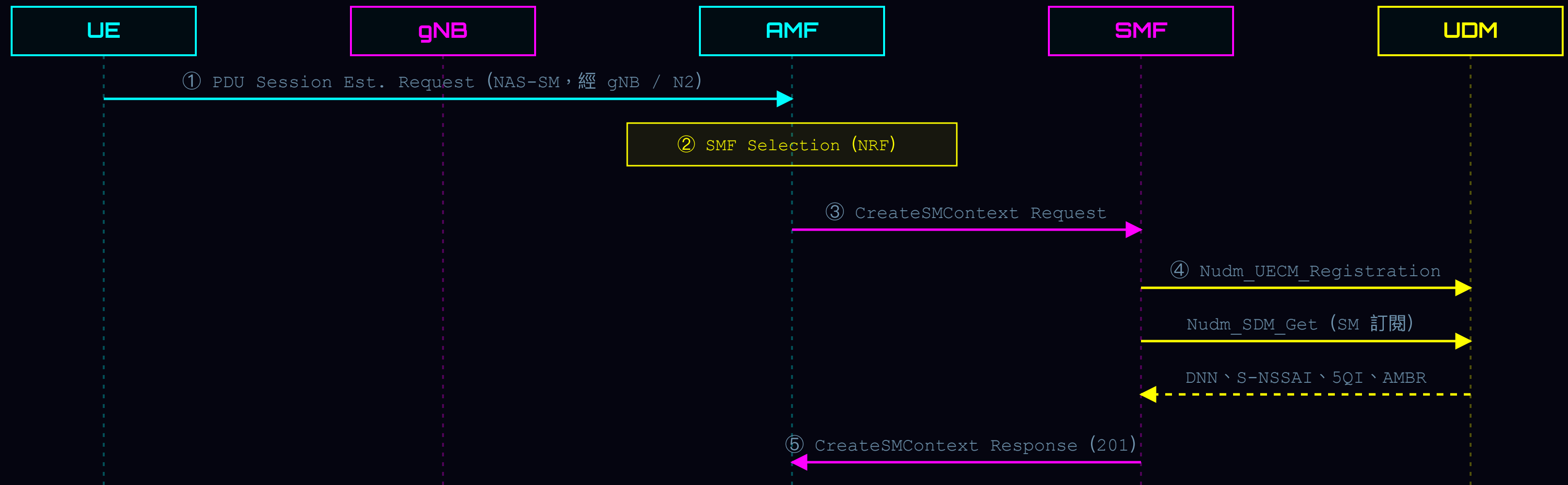
➤ PDU SESSION 建立流程

對照 TS 23.502 §4.3.2.2.1，逐步拆解每一個訊息

／> 四個階段



➤ STEP 1-5：請求與訂閱



➤ STEP 1-5 重點

- ①：Request 帶 PDU Session ID、DNN、S-NSSAI、PDU Type、SSC mode
- ②：AMF 依 DNN / S-NSSAI 向 NRF 找 SMF
- ③：AMF 把 NAS-SM 原封不動放進 CreateSMContext
- ④：SMF 向 UDM 登記自己，並取得 SM 訂閱資料
- ⑤：先回 201，告訴 AMF 「SM Context 已建立」

Step 6 是選用的二次認證（DN-AAA），一般實驗環境不會出現。

/> STEP 7-10：策略、選路與 N4

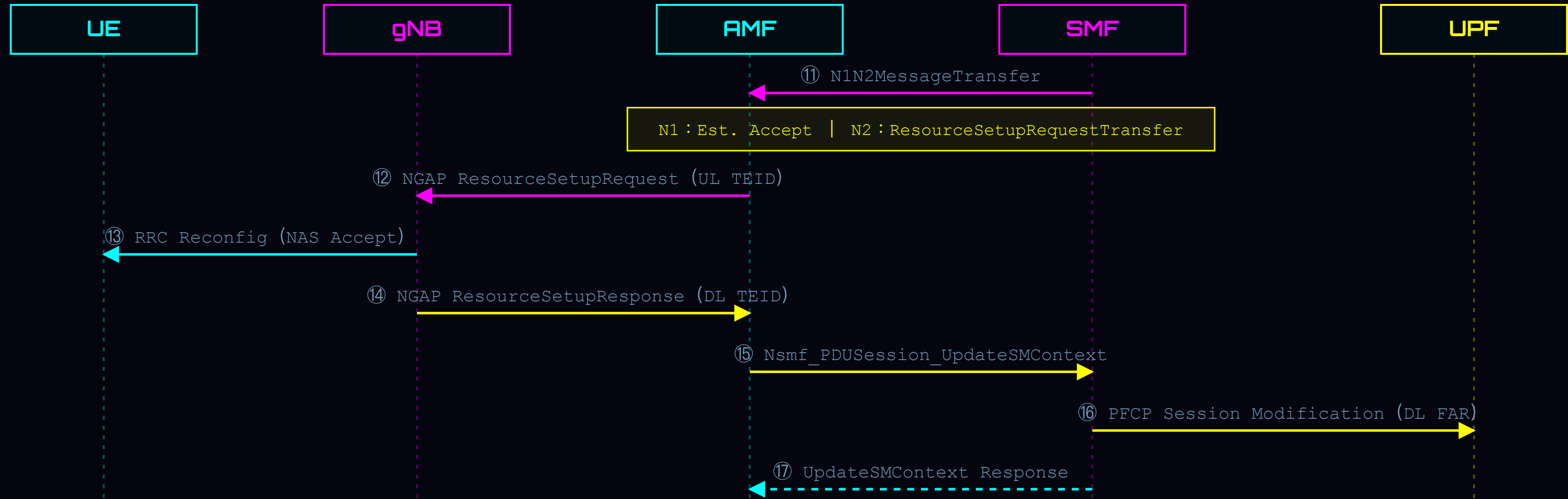


➤ STEP 7-10 重點

- ⑦：建立 SM Policy Association，拿到 PCC rules 與 Session rules
- ⑧：依 DNN、S-NSSAI、DNAI 選 UPF，並從該 UPF 的 pool 分配 UE IP
- ⑨：選用，SMF 主動通知 PCF 策略觸發條件
- ⑩：PFCP Session Establishment：上行規則就緒

此時 UPF 還不知道 gNB 的 DL TEID

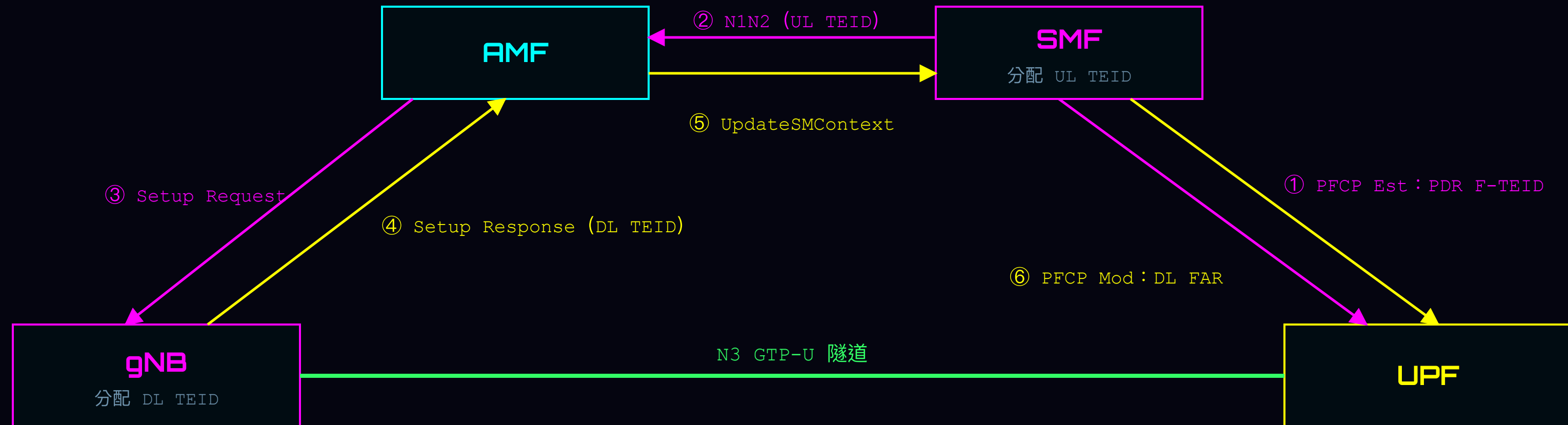
➤ STEP 11-17 : TEID 交換



➤ STEP 11-17 重點

- ⑪：SMF 一次交給 AMF 兩份東西：N1（給 UE）、N2（給 gNB）
- ⑫：AMF 把 N2 SM Info 包進 NGAP，並附上 NAS-PDU
- ⑬：gNB 配置無線資源，把 Accept 交給 UE
- ⑭-⑮：gNB 回報自己的 DL TEID，AMF 轉交 SMF
- ⑯：SMF 更新 DL FAR，下行路徑打通

➤ TEID 交換：兩個方向、兩條路



上行封包：外層 TEID = UPF 的 UL TEID

下行封包：外層 TEID = gNB 的 DL TEID

➤ SMF 「主動送 NGAP」 給 GNB ?



內容由 SMF 決定，傳遞由 AMF 負責：SMF 沒有、也不需要和 gNB 之間的連線。

RESOURCESETUPREQUESTTRANSFER

裡有什麼

IE	內容
UL NG-U UP TNL Information	UPF 的 N3 IP + UL TEID (上行要送去哪)
PDU Session Type	IPv4 / IPv6 / Ethernet ...
PDU Session AMBR	整條會話的上下行速率上限
QoS Flow Setup Request List	每個 QFI 的 5QI、ARP、GBR 參數
Additional UL NG-U ...	雙連結 (NR-DC) 時的第二條上行隧道

RESOURCESETUPRESPONSETRANSFER

裡有什麼

DL QoS Flow per TNL Info

最重要的欄位

gNB 的 N3 IP 位址

gNB 分配的 DL TEID

對應的 QoS Flow 清單

其他資訊

選用欄位

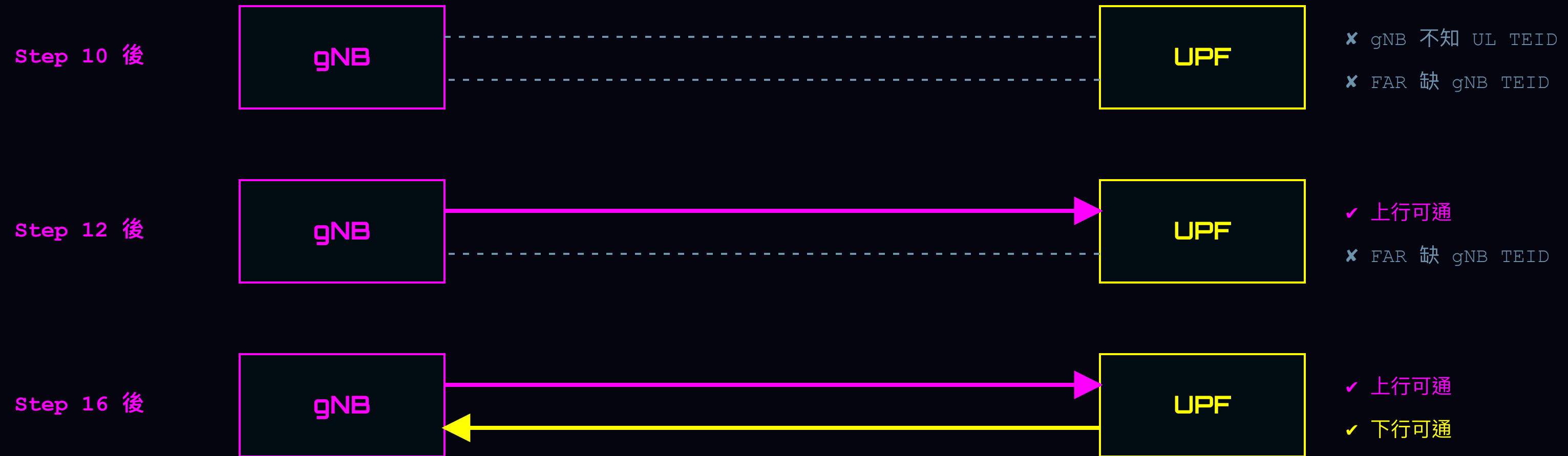
Additional DL TNL (NR-DC)

Failed QoS Flow List

Security Result

SMF 解碼這份 Transfer，拿到 IP + TEID 後寫進下行 FAR 的 Outer Header Creation。

／> 隧道狀態的變化



UE 收到 Accept 後可以先送上行；下行要等 Step 16 完成才會真正送到 UE。

FREE5GC : HANDLEPDUSESSIONSMCONTEXTCREATE

```
// internal/sbi/processor/pdu_session.go (節錄)
Consumer().SendNFDiscoveryUDM()
Consumer().GetSmData(...)           // ④ SM 訂閱
smContext.AllocUeIP()                // ⑧ 選 PSA + 分 IP
Consumer().BSFAwarePCFSelection()    // ⑦a
Consumer().SendSMPolicyAssociationCreate()
Consumer().CHFSelection()
smContext.ApplyPccRules(decision)
smContext.SelectDefaultDataPath()    // DFS 找路徑
smContext.SelectULCLDataPaths()
go ActivateUPFSession(...)           // ⑩ → ⑪
return 201 Created                    // ⑤
```

順序與規格一致

PFCP 在 goroutine

先回 201



FREE5GC :

SENDPDUSESSIONESTABLISHMENTACCEPT

```
smNasBuf, _ := BuildGSMPDUSessionEstablishmentAccept(smContext) // N1
n2Pdu, _ := BuildPDUSessionResourceSetupRequestTransfer(smContext) // N2
// └ ULNGUPTNLInformation: newGTP TunnelUPTransport(n3IP, LocalULTeid)
n1n2Request.N2InfoContainer.SmInfo.N2InfoContent.NgapIeType = PDU_RES_SETUP_REQ
Consumer().N1N2MessageTransfer(ctx, supi, n1n2Request, amfApiPrefix) // ⑪
```

UL TEID 是 SMF 自己分配的 LocalULTeid，也寫進 UPF 的 PDR F-TEID

NgapIeType 告訴 AMF：這份 N2 要放進 ResourceSetupRequest

➤ FREE5GC : 收到 DL TEID 之後

```
// HandlePDUSessionSMContextUpdate → ⑮
HandlePDUSessionResourceSetupResponseTransfer(n2SmInfo, smContext)
    GTPTunnel := DLQosFlowPerTNLInformation.UPTransportLayerInformation
    ctx.Tunnel.UpdateANInformation(GTPTunnel.IP, GTPTunnel.TEID)

// datapath.go : 產生下行 FAR
DLFAR.ForwardingParameters.OuterHeaderCreation.Teid =
    smContext.Tunnel.ANInformation.TEID
→ PFCP Session Modification Request                                // ⑯
```

/> 實作驗證：抓封包照順序看

1

ngap : InitialUEMessage / UplinkNASTransport

找到 PDU Session Establishment Request

2

pcfcp : Session Establishment

記下 Create PDR 裡的 F-TEID (UL TEID)

3

ngap : PDUSessionResourceSetupRequest / Response

UL TEID 是否一致？DL TEID 是多少？

4

pcfcp : Session Modification → gtp

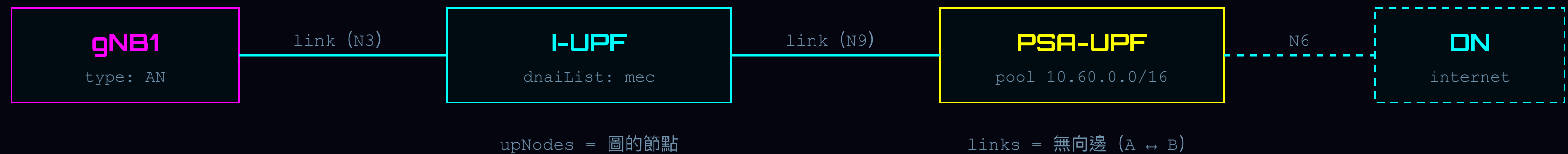
FAR 的 OHC TEID = 下行 GTP-U 的 TEID

\$ PART 07

➤ UPF 拓樸與選擇

free5GC SMF 如何在一張圖上選出錨點與路徑

➤ SMF 眼中的使用者面：一張圖



拓樸寫在 smfcfg.yaml 的 `userplaneInformation`：AN 與 UPF 是節點，links 是邊。

➤ SMFCFG.YAML : USERPLANEINFORMATION

```
userplaneInformation:
  upNodes:
    gNB1: { type: AN }
    I-UPF:
      type: UPF
      sNssaiUpfInfos:
        - sNssai: { sst: 1, sd: 010203 }
          dnnUpfInfoList:
            - dnn: internet
              dnaiList: [mec]
          interfaces: [N3, N9]
      PSA-UPF:
        type: UPF
        ... pools: [cidr: 10.60.0.0/16]
  links:
    - { A: gNB1, B: I-UPF }
    - { A: I-UPF, B: PSA-UPF }
```

切片與 DNN

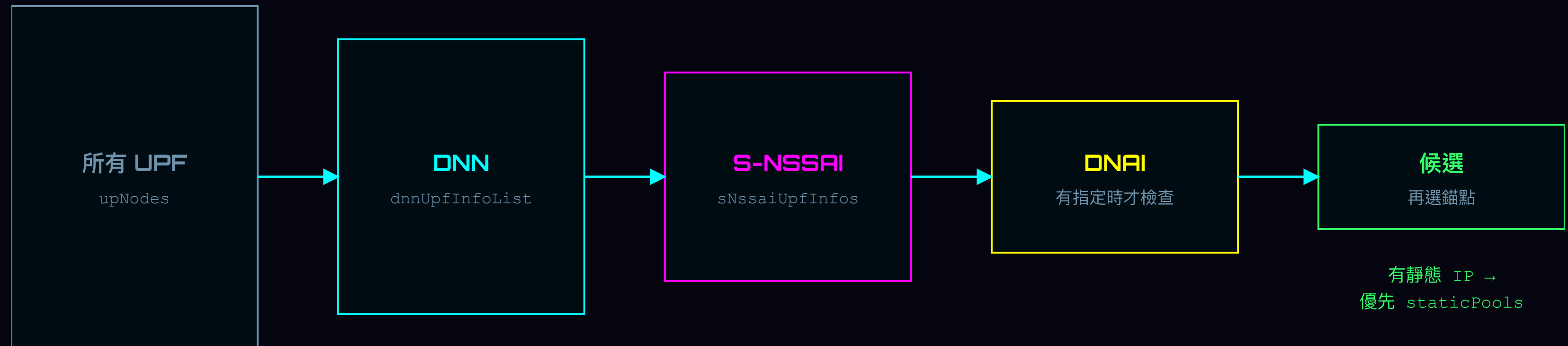
DNAI 位置

介面類型

IP pool

連線關係

／> 選擇條件：UPFSELECTIONPARAMS

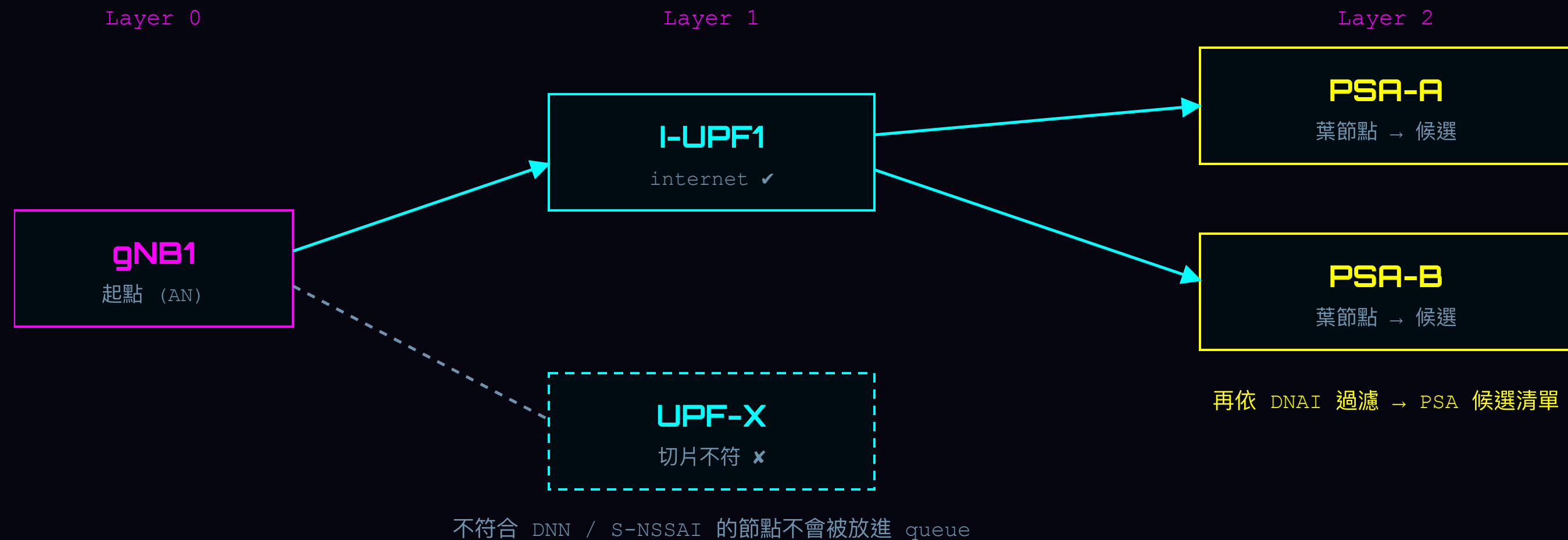


/> DIJKSTRA ? BFS ? DFS ? 先講結論

步驟	FREE5GC 實作	演算法
選錨點 PSA	selectAnchorUPF	BFS (queue)
找路徑	getPathBetween	DFS (遞迴)
ULCL 預設路徑	getAllPathByNodeName	DFS 列舉，取第一條

沒有 Dijkstra：連線沒有權重。另外常被混淆的 BSF 是 Binding Support Function，SMF 用它找 PCF，與選路無關。

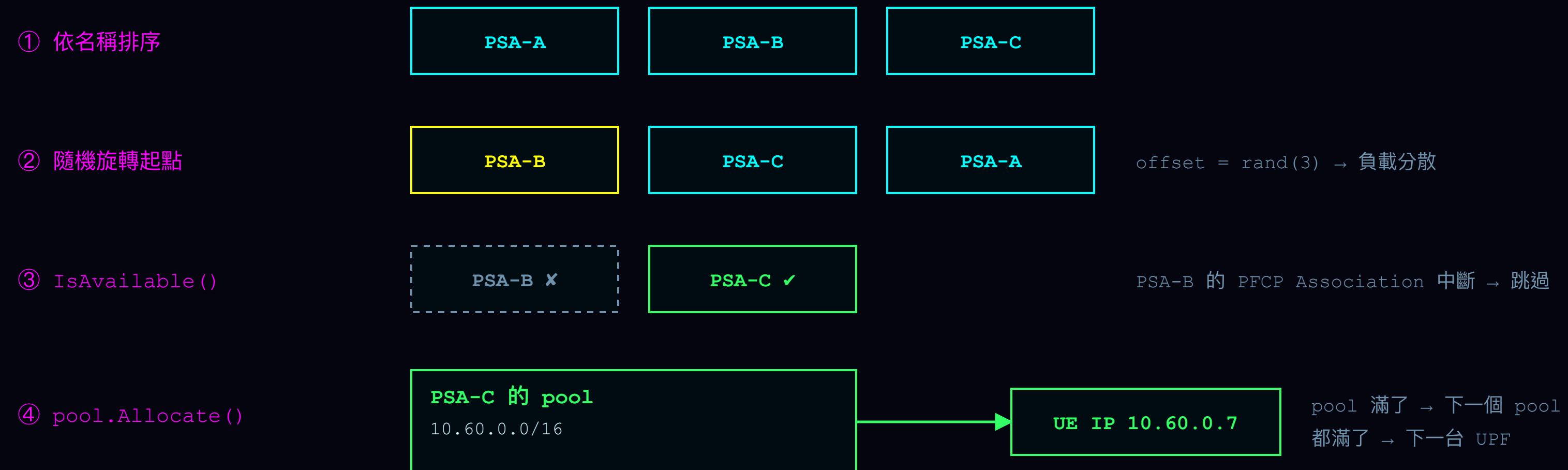
➤ STEP 1: BFS 找錨點候選



/> SELECTANCHORUPF (簡化)

```
queue := []*UPNode{source}           // 從 AN (gNB) 出發
for len(queue) > 0 {
    node := queue[0]; queue = queue[1:]
    visited[node] = true
    findNewNode := false
    for _, link := range node.Links {
        if !visited[link] && link.MatchedSelection(dnnAndSnssai) {
            queue = append(queue, link); findNewNode = true
        }
    }
    // 沒有更遠的可用鄰居 → 葉節點 → 錨點候選
    if !findNewNode && node.Type == UPF && node.MatchedSelection(selection) {
        upList = append(upList, node)
    }
}
```

/> STEP 2：挑一個 PSA 並分配 IP



➤ STEP 3 : DFS 找出到 PSA 的路徑

```
func getPathBetween(cur, dest, visited, sel) {  
    visited[cur] = true  
    if cur == dest { return [cur], true }  
    for _, node := range cur.Links {  
        if visited[node] { continue }  
        if !node.UPF.isSupportSnssai(sel.SNssai) {  
            visited[node] = true; continue // 剪枝  
        }  
        tail, ok := getPathBetween(node, dest, ...)  
        if ok { return [cur] + tail, true }  
    }  
    return nil, false  
}
```

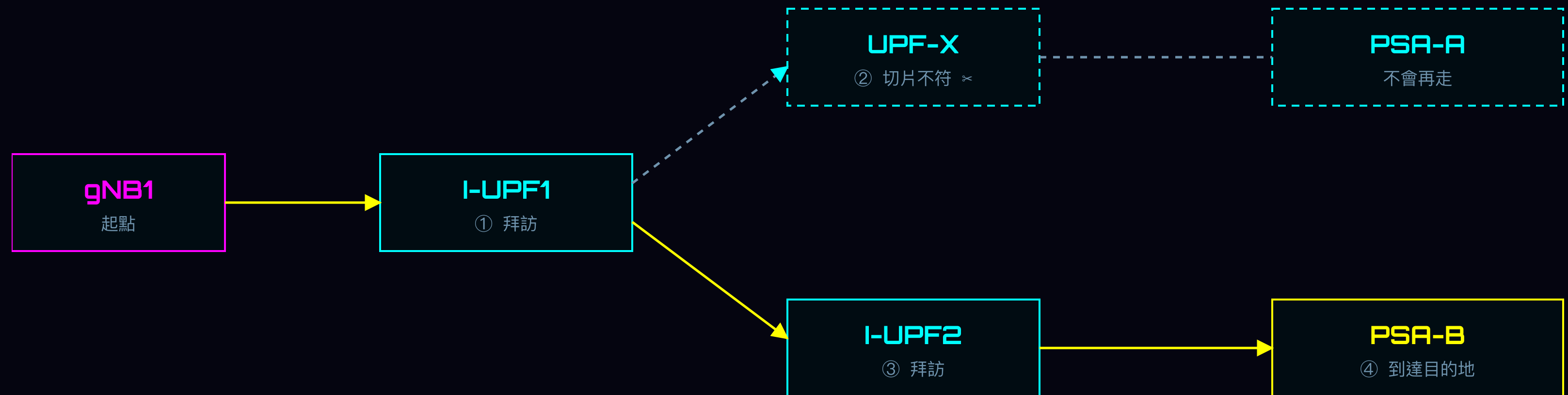
遞迴深搜

切片不符就剪枝

找到第一條就回傳

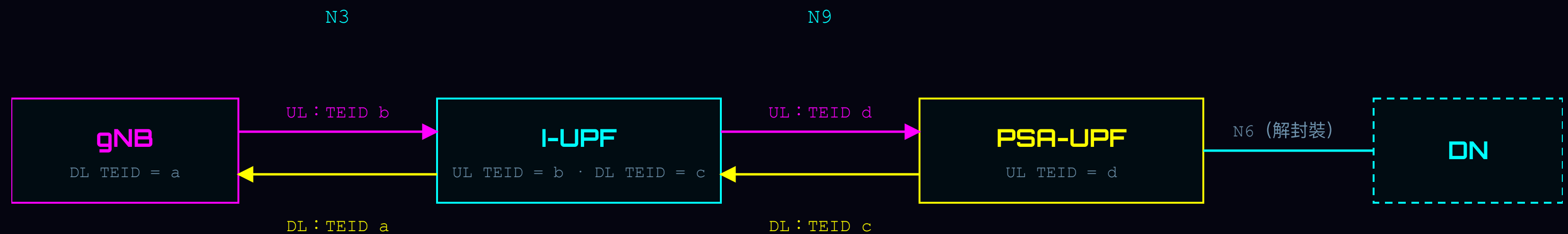
結果會被快取

➤ DFS 走訪示意



回傳路徑：gNB1 → I-UPF1 → I-UPF2 → PSA-B (第一條找到的，不保證最短)

➤ 路徑變成 DATAPATH：每一跳都要 TEID



每個 DataPathNode 都有 UpLinkTunnel / DownLinkTunnel，各自帶 PDR 與 TEID

／> 現行做法的特性與研究空間

特性

教學導向的簡單實作

連線無權重，不保證最短

取第一條找到的路徑

只取第一個 AN 當起點

可以研究的方向

課後專題靈感

加入延遲 / 負載權重

改用 Dijkstra 或 k-shortest

依 UE 位置選 AN 起點

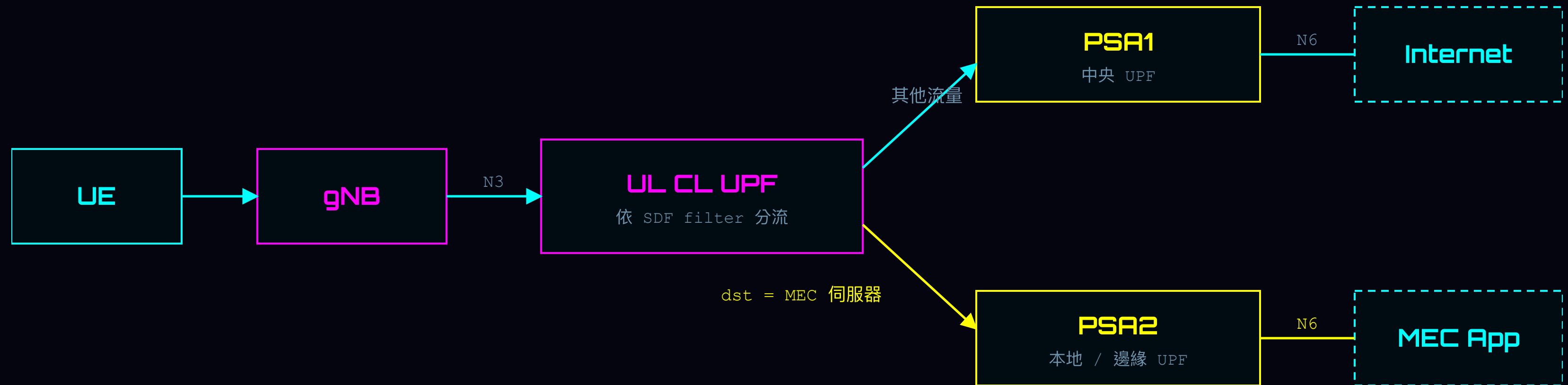
實驗拓樸通常只有 2~3 台 UPF，BFS + DFS 已足夠；規模變大才需要更好的選路策略。

\$ PART 08



Uplink Classifier：同一條會話，兩個出口

／> 為什麼需要 ULCL?



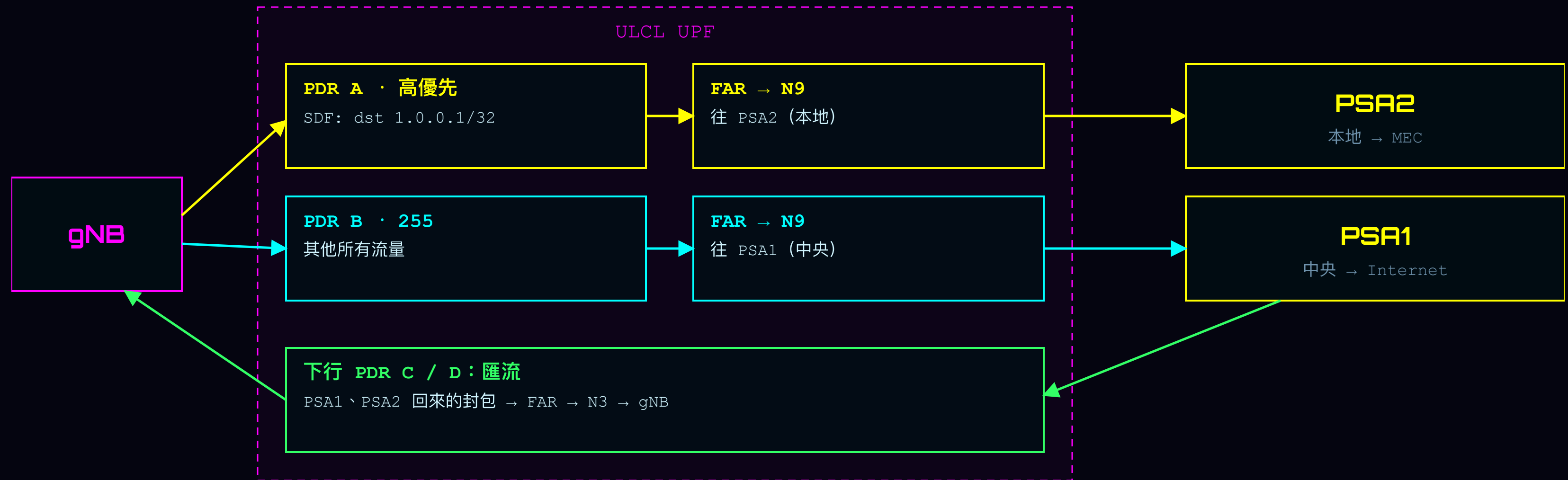
特定流量在邊緣就近下車 (Local Breakout)，其餘照常走中央 UPF；UE 完全無感。

➤ ULCL VS BRANCHING POINT



free5GC 以 ULCL 為主；原始碼中的 BPManager 統一管理「新增錨點並插入分流點」的過程。

/> ULCL UPF 上的規則



➤ FREE5GC 的 ULCL 設定

```
# smfcfg.yaml
configuration:
  ulcl: true
  userplaneInformation:
    upNodes: ...
    links: ...
```

```
# uerouting.yaml
ueRoutingInfo:
  UE1:
    members: [imsi-208930000000001]
    topology:
      - { A: gNB1, B: I-UPF }
      - { A: I-UPF, B: PSA-UPF }
    specificPath:
      - dest: 1.0.0.1/32
        path: [I-UPF]
  pfdDataForApp:
    - applicationId: app1 ...
```

依 SUPI 分組：組員走預設 topology，送往 specificPath.dest 的流量改走指定路徑。

FREE5GC : ADDPDUSESSIONANCHORANDULCL

1

SelectPSA2

從 ULCL 路徑池挑出尚未啟用的新錨點路徑

2

FindULCL

比對新舊路徑，找出分流點

3

EstablishPSA2

在 ULCL 之後的 UPF 建立 / 修改 PFCP Session

4

EstablishULCL

在分流點 UPF 加入上行分流規則

5

UpdatePSA2DownLink

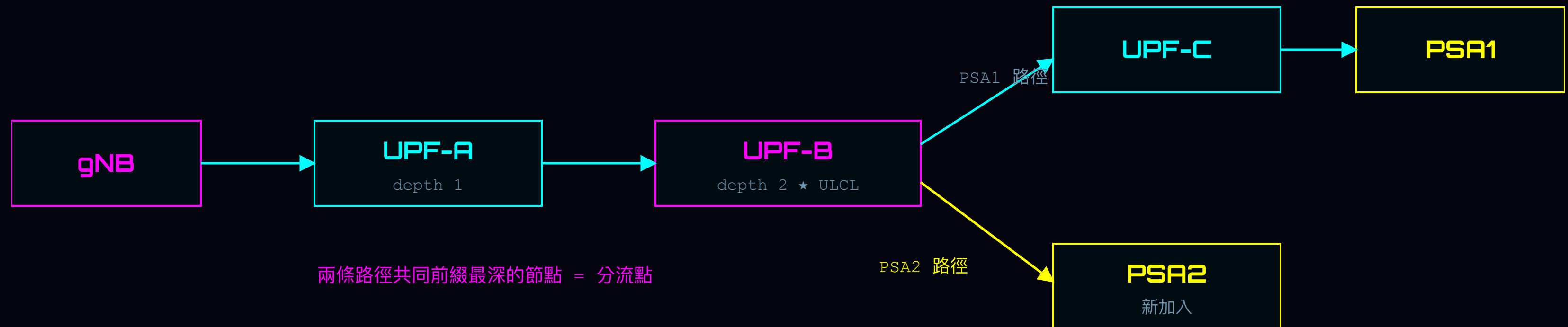
補上 PSA2 往回送的下行規則

6

UpdateRANAndIUPFUpLink

W6 · SMF更新RAN 與中間 UPF 的上行轉送

➤ FINDULCL：最深的共同節點



逐跳比對 PSA1 與 PSA2 路徑，記錄共同前綴深度；depth 最大的 UPF 就是 ULCL。

➤ 動手看：FREE5GC-COMPOSE 的 ULCL

1

config/ULCL/

I-UPF 與 PSA-UPF 兩台 UPF，smfcfg 開啟 ulcl: true

2

UE 建立 PDU Session

SMF 依 uerouting 建立預設與特定路徑

3

ping 1.0.0.1 / ping 8.8.8.8

在 I-UPF 與 PSA-UPF 抓封包比較

4

觀察 PFCP

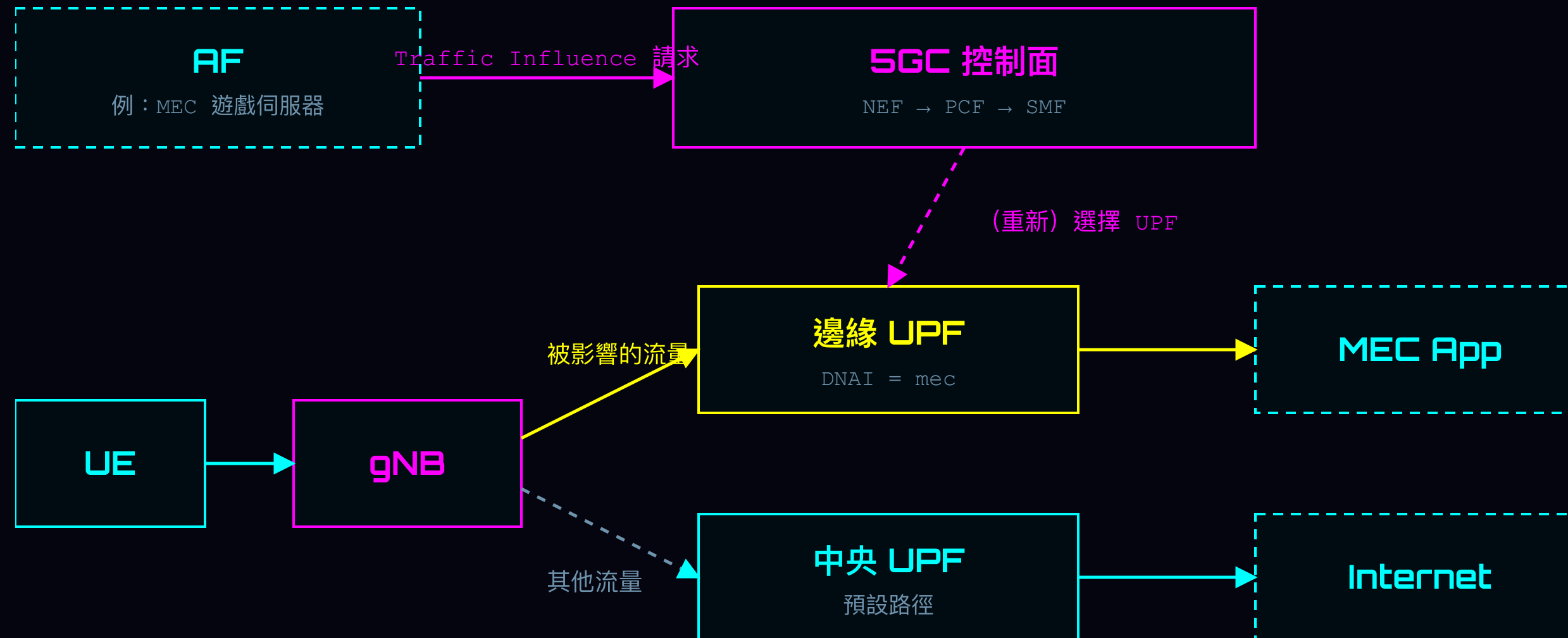
I-UPF 上有兩條帶不同 Precedence 的上行 PDR

\$ PART 09

TRAFFIC INFLUENCE

讓應用（AF）動態影響 SMF 的選路

➤ TRAFFIC INFLUENCE 是什麼？



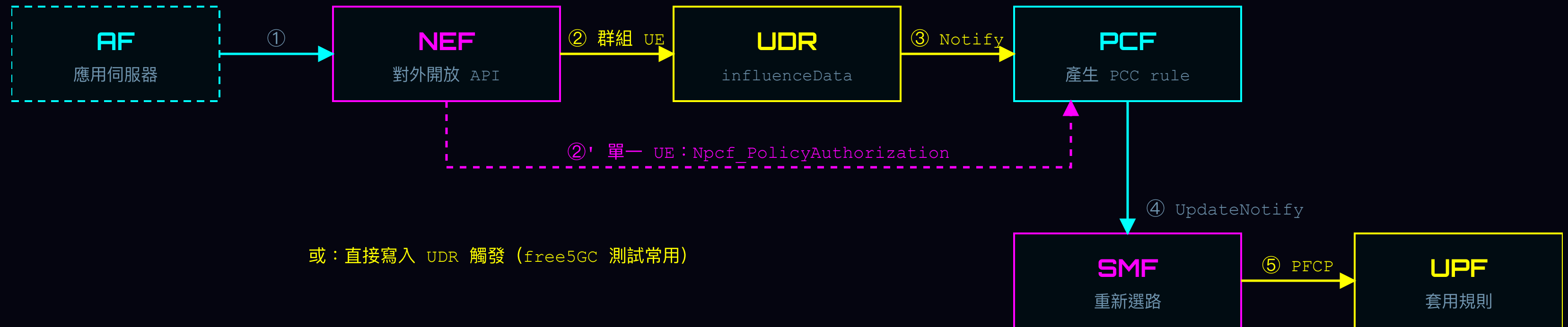
TS 23.501 §5.6.7

DNAI：抽象位置

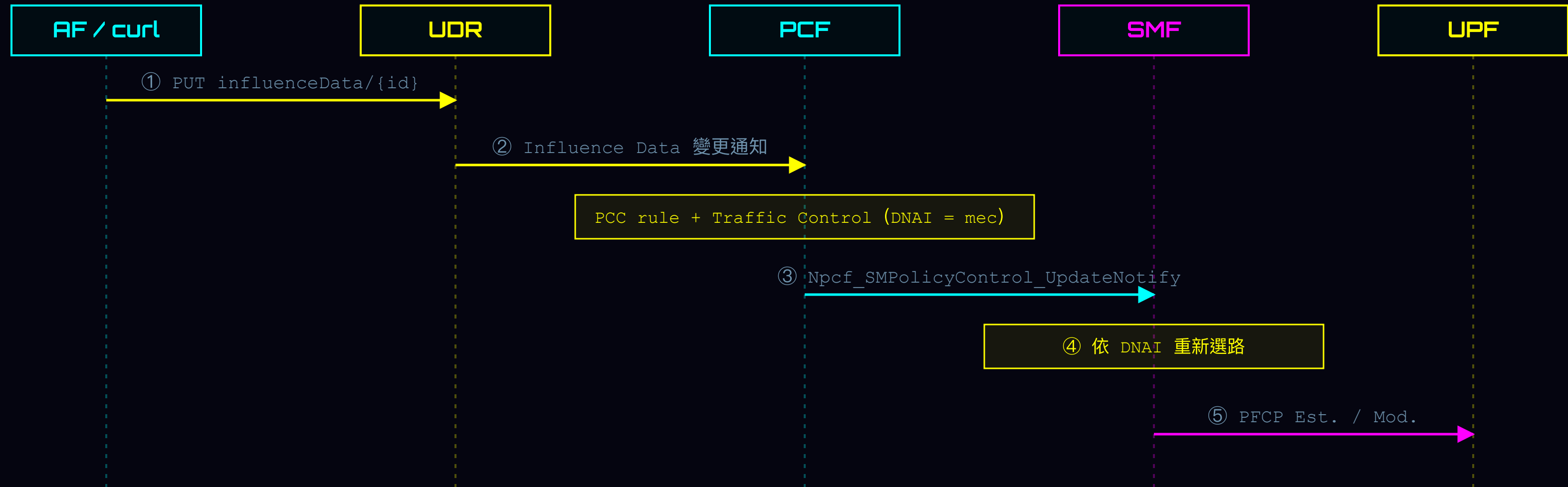
執行期動態下達

不必改設定檔

➤ FREE5GC 的 TRAFFIC INFLUENCE 架構



➤ 路徑一：寫入 UDR 觸發



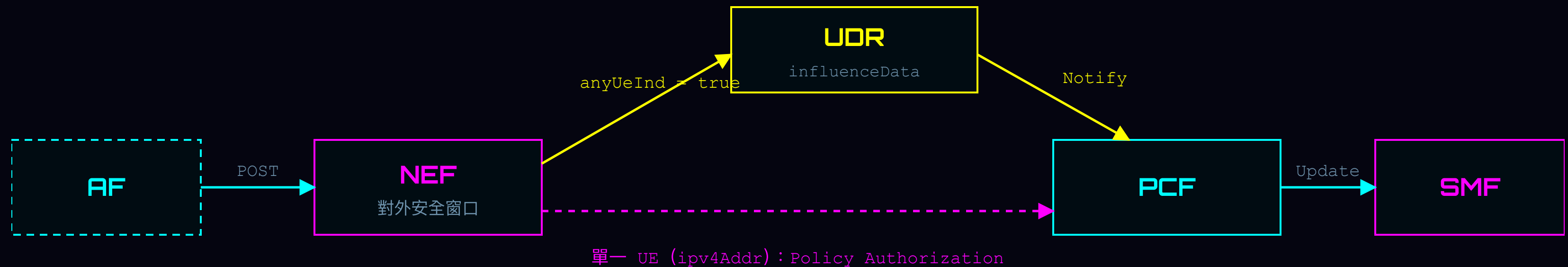
➤ UDR INFLUENCEDATA API

```
# 建立 / 更新
curl -X PUT --data @ti_data.json \
  http://<udr>:8000/nudr-dr/v1/application-data/influenceData/1
# 查詢
curl -X GET ".../application-data/influenceData?dnns=internet"
# 刪除：路由恢復預設
curl -X DELETE .../application-data/influenceData/1
```

UDR 只負責存資料並通知訂閱者（PCF）；UDR 的細節之後的章節再談。

➤ 路徑二：AF 透過 NEF

```
curl -X POST --data @af_ti_data.json \  
http://<nef>:8000/3gpp-traffic-influence/v1/af001/subscriptions
```



➤ TRAFFIC INFLUENCE 請求內容

```
{
  "afServiceId": "Service1",
  "dnn": "internet",
  "snssai": { "sst": 1, "sd": "010203" },
  "anyUeInd": true,
  "trafficFilters": [{
    "flowId": 1,
    "flowDescriptions": [
      "permit out ip from 1.0.0.1 to 10.60.0.0/16"
    ]
  }],
  "trafficRoutes": [{ "dnai": "mec" }]
}
```

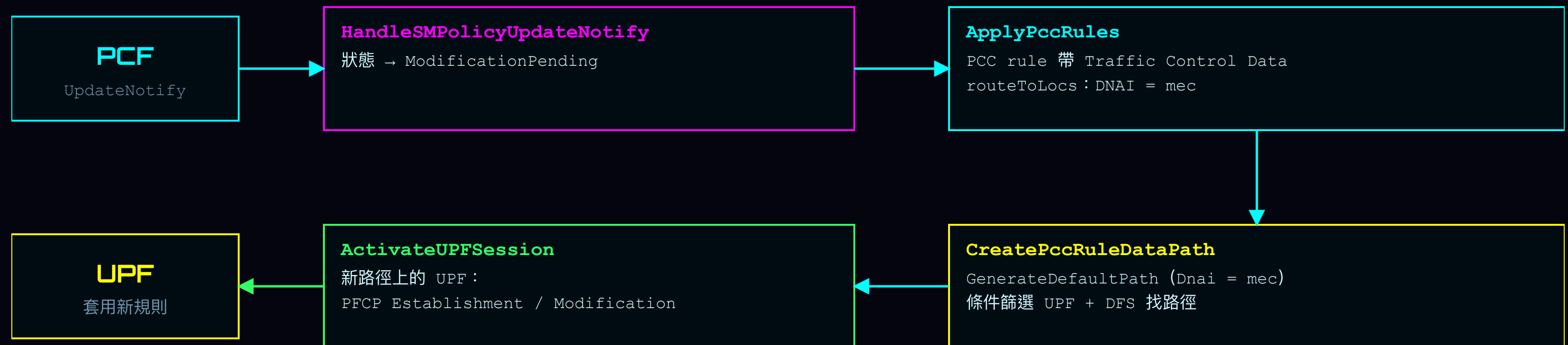
對哪些會話

哪些流量

導向哪個 DNAI

欄位為示意，實際格式請對照 free5GC 官網 Traffic Influence 指南的範例檔。

/> SMF 收到策略更新之後



Traffic Influence 最後還是回到同一套選路邏輯，只是多了 DNAI 條件

➤ ULCL 設定 VS TRAFFIC INFLUENCE

比較	USERROUTING 預先設定	TRAFFIC INFLUENCE
誰決定	維運人員寫設定檔	AF / 應用在執行期請求
生效時機	建立會話時	隨時：新增、修改、刪除
經過的 NF	只有 SMF	NEF → UDR → PCF → SMF
描述位置	specificPath (UPF 名稱)	DNAI (抽象位置)

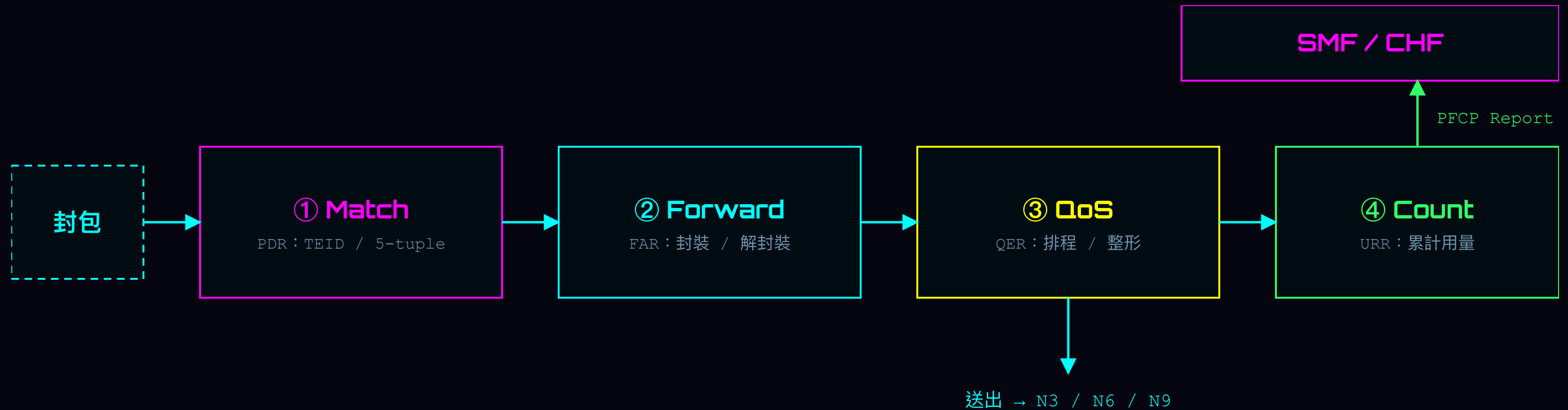
NEF、UDR 與 PCF 的完整功能，會在之後介紹其他 NF 的章節詳細說明。

\$ PART 10

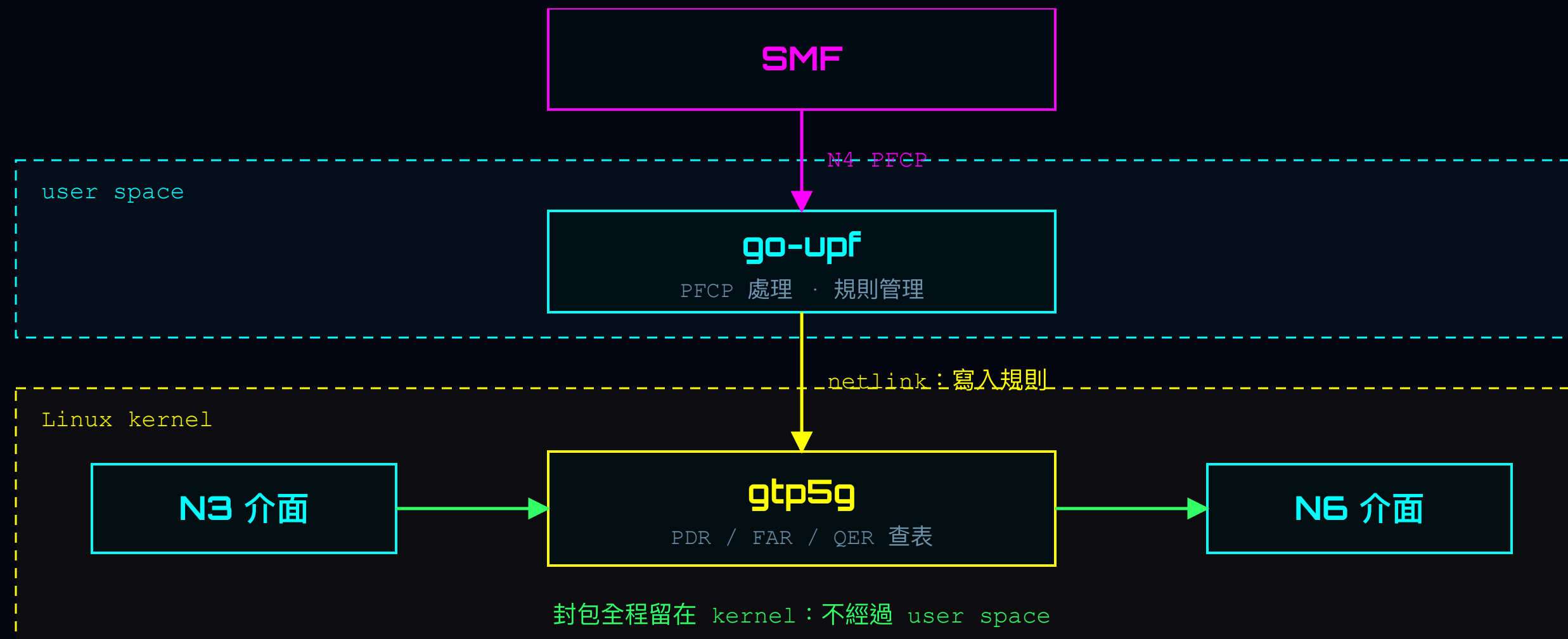
➤ UPF 與 GTP5G

free5GC 的資料面是怎麼跑起來的

➤ UPF 處理一個封包的四步



/> GO-UPF + GTP5G



➤ GTP5G 負責什麼

gtp5g (kernel)

5G 專用 GTP-U 引擎

管理隧道與 TEID / IP 映射

封裝與解封裝

配合 routing / qdisc / nftables

go-upf (user space)

PCFP 與規則管理

接收 SMF 的 PCFP 規則

經 netlink 寫進 gtp5g

收集用量、回報 SMF

好處：資料面直接用 Linux 核心的封包能力；UPF 程式專心處理 5GC 邏輯。

／> 其他資料面加速技術

DPDK

KERNEL BYPASS

使用者空間輪詢網卡，吞吐量極高

EBPF / XDP

可程式化 KERNEL

在網卡驅動層處理封包，不改核心

P4

可程式化交換器

把轉送邏輯下放到硬體管線

不論哪一種，上層都一樣：SMF 下 PFCP 規則，資料面引擎負責照規則執行。

／> 本章小結

SMF 想規則、UPF 做執行：N4 / PFCP 連接兩者

N3 / N9 用 GTP-U 隧道，TEID 由接收端分配

PDR 比對，FAR / QER / URR 決定轉送、QoS 與計費

建立流程：UL TEID 經 N1N2 → NGAP 給 gNB，DL TEID 反向回 SMF

free5GC 選路：BFS 找錨點、DFS 找路徑；ULCL 與 TI 實現就近分流

下一章：其他網路元件（PCF、UDM、UDR、NEF ...）