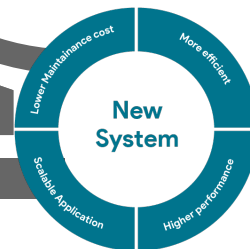


Legacy IT Environment



Modern Application Platform

## 微服務 Micro-service 應用程式開發與系統平台技術顧問諮詢服務建議方案

Service Proposal for Micro-service App Development & Infra Plan & Design Consulting Service

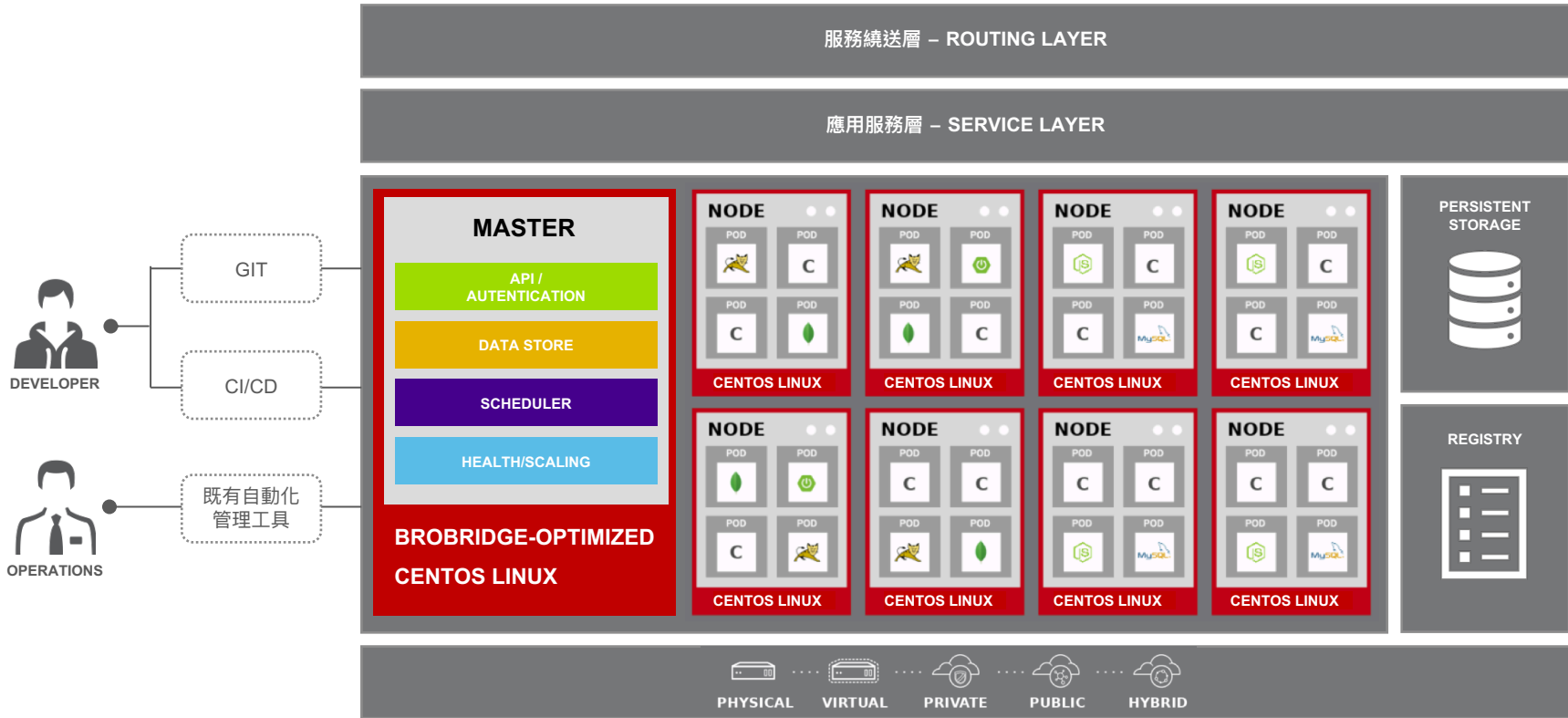
01

# 容器化管理平台建置規劃建議

Container Platform Plan & Design Proposal

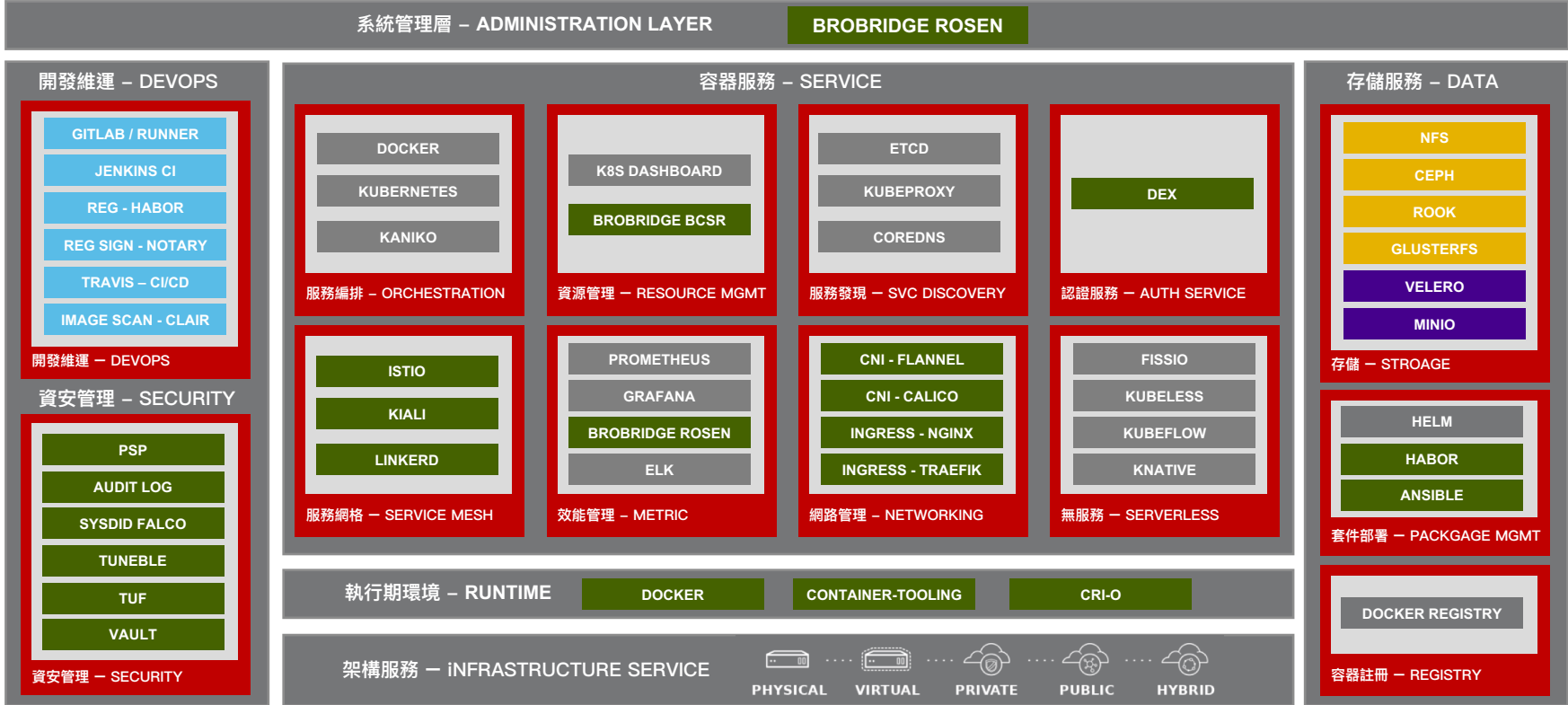
# KUBERNETES 基礎架構模型規劃

容器化管理平台叢集環境建置規劃-建置模型



# KUBERNETES 技術元件規劃

容器化管理平台叢集環境建置規劃-技術元件



- 基本安裝
- 儲存相關
- 備援相關
- CI/CD相關
- 資訊安全





**App Definition and Development**

Database: Veeva, KV, Confluent, SAP, Signio, Amazon, BROADCOM, etc.

Streaming & Messaging: cloudevents, NATS, etc.

Application Definition & Image Build: HELM, etc.

Continuous Integration & Delivery: argo, etc.

**Orchestration & Management**

Scheduling & Orchestration: kubernetes, etc.

Coordination & Service Discovery: Consul, etc.

Remote Procedure Call: gRPC, etc.

Service Proxy: envoy, etc.

API Gateway: Kong, etc.

Service Mesh: LINKERD, etc.

**Runtime**

Cloud Native Storage: minio, etc.

Container Runtime: cri-o, etc.

Cloud Native Network: CNI, etc.

**Provisioning**

Automation & Configuration: Ansible, etc.

Container Registry: Harbor, etc.

Security & Compliance: Falco, etc.

Key Management: Vault, etc.

**Cloud Native Landscape**

Cloud Native Computing Foundation

Redpoint Amplify

Special: D2 IQ, etc.

Kubernetes Certified Service Provider: IBM, etc.

**Platform**

Certified Kubernetes - Distribution: OpenShift, etc.

Certified Kubernetes - Hosted: AWS, etc.

Certified Kubernetes - Installer: Kubeadm, etc.

PaaS/Container Service: Flynn, etc.

**Kubernetes Training Partner**

Members: IBM, etc.

**Observability and Analysis**

Monitoring: Prometheus, etc.

Logging: fluentd, etc.

Tracing: Jaeger, etc.

Chaos Engineering: Chaos Mesh, etc.

**Serverless**

Cloud Native Landscape

See the serverless reduction depth of a CNCF.io

**Members**

Gridset logo

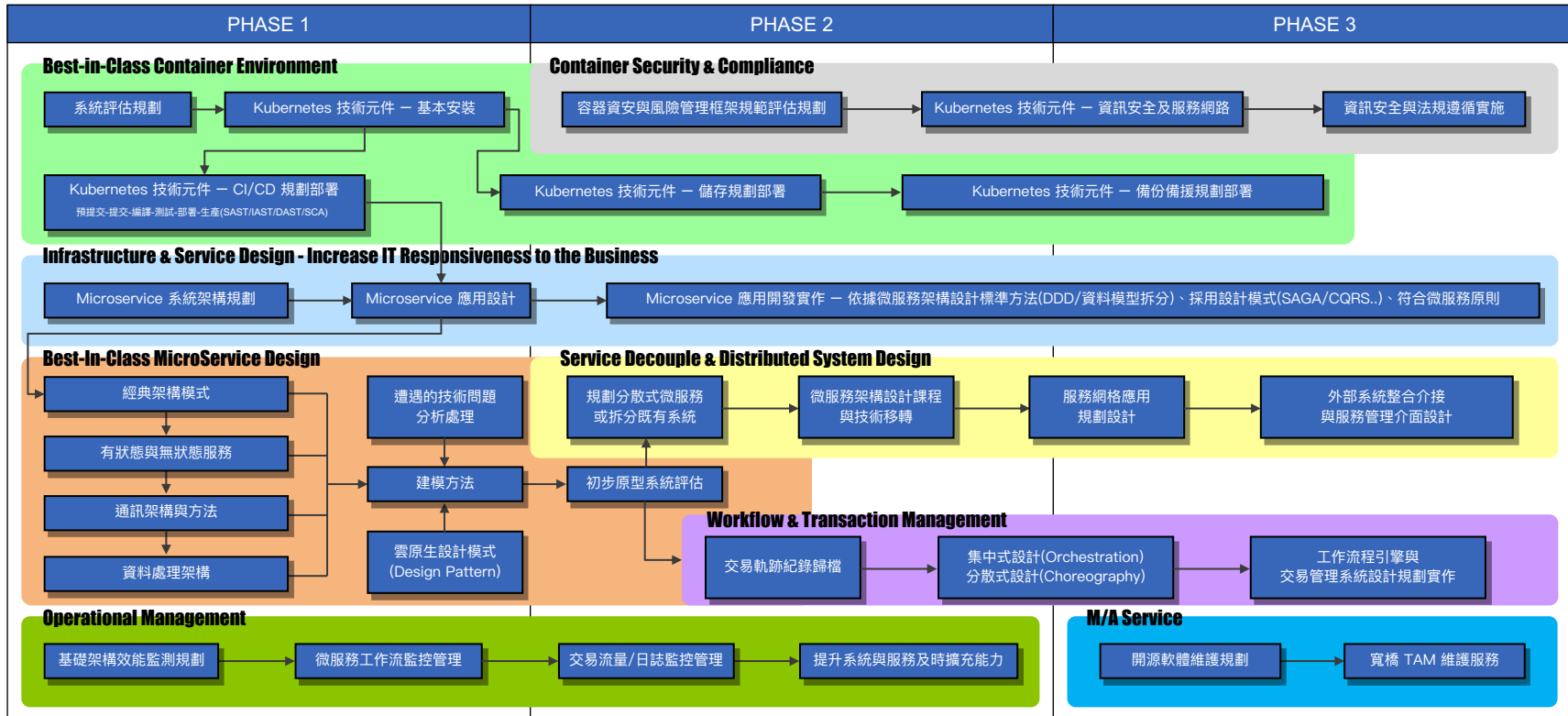
02

# 微服務導入顧問服務建議

MicroService Consulting Service

# 微服務導入路線圖

MICROSERVICE BEST-IN-CLASS ROADMAP



03

# Kubernetes 技術經理服務

Kubernetes Technical Account Manager Service

# KUBERNETES TAM 服務主要工作內容

BROBRIDGE KUBERNETES TAM SERVICE – MAJOR ACTIVITIES

## 技術諮詢服務

- 最佳實踐
- 新興技術評估與建議
- K8S 已知/潛在問題說明
- 資安弱點說明
- 協助評估 K8S 延伸功能

## 技術發展規劃

- K8S 技術元件發展 ROADMAP
- K8S 技術元件深度剖析
- 產品經理-技術研發SME 會議
- 團隊技術教育訓練(半年一次)

## 系統檢查與問題處理

- 系統健康檢查(半年一次)
- 技術問題分析
- 技術問題解決
- 技術問題跟進

## 產品或技術需求申請

- 提交產品功能 RFE
- 功能提交進度報告
- 維運與開發整合方案
- 客戶 ROUND TABLE 會議

## 客戶例行性活動

雙週例會

季度 TAM 執行報告

提供 TAM Days 資訊分享

滿意度調查 (半年一次)

# KUBERNETES TAM 服務內容說明

BROBRIDGE KUBERNETES TAM SERVICE WITH IMPLEMENTATIONS

| 基礎功能 | 技術元件內容                           |
|------|----------------------------------|
| 服務編排 | ■ DOCKER ■ K8S ■ KANICO          |
| 資源管理 | ■ K8S DASHBOARD ■ BROBRIDGE BCSR |
| 服務發現 | ■ ETCD ■ KUBEPROXY               |
| 認證服務 | ■ DEX                            |
| 服務網格 | ■ ISTIO *進階功能                    |
| 效能管理 | ■ PROMETHEUS ■ GRAFANA ■ ROSEN   |
| 網路管理 | ■ CALICO                         |
| 無服務  | *進階功能                            |
| 存儲服務 | ■ NFS ■ CEPH ■ VELERO ■ MINIO    |
| 套件管理 | ■ HELM ■ HAVOR ■ ANSIBLE         |
| 容器註冊 | ■ DOCKER REGISTRY                |
| 開發維運 | ■ GITLAB ■ NOTARY                |
| 資安管理 | ■ PSP ■ AUDIT LOG *進階功能          |

建議至少一年 K8S  
國際CNCF認證  
專業 TAM 服務

BROBRIDGE
DATASHEET  
TAM Service

## 寬橋 Kubernetes 技術經理服務

**客戶使用情境說明**

在下列情況下，您應該考慮技術經理服務

- 您的組織和工作人員將得到專家的幫助，學習 K8S 相關技術。
- 您希望透過充分利用 K8S 與 Linux 技術來簡化基礎架構運營。
- 您需要一個能夠理解您的需求並具體地分析、規劃、設計的單一聯繫點，找出更好地利用 K8S 技術的機會。
- 您需要協助開發策略，並在內部宣傳您的 K8S 和 Linux 投資的價值。
- 尋求經驗性的即時引導，使您能夠避免常見的陷阱和管理複雜性，並制定組織內部的標準規範。

**K8S 技術客戶經理服務工作標準說明**

| 工作內容          | 描述  | Q-TAM | M-TAM |
|---------------|---|-------|-------|
| 服務時間          | 12 個月 (每週一至週五上班時間、例假日除外)                      | ✓     | ✓     |
| 每月定期檢查        | 1部 Master + 3 部 Worker K8S 主機系統健康狀況檢查         | ✓     | ✓     |
| 遠程技術支援        | 電話 / eMail / 問題處理支援服務                         | ✓     | ✓     |
| 產品問題服務請求及進度追蹤 | 跟蹤用戶需求和故障問題，提交結果報告                            | ✓     | ✓     |
| 技術評估協助        | 1、指定 K8S 產品相關的評估與研析服務<br>2、知識傳遞和最佳實踐分享        | 4次    | 4次    |
| 軟體授權報告        | 用戶軟體授權的使用數量報告                                 | 4次    | 4次    |
| 最新修補程式發佈報告    | 針對 K8S 軟體產品的修補程式和設置                           | 4次    | 4次    |
| 多供應商合作        | 透過客戶既有的協作支援協議，TAM 將協助收斂問題的範圍，並將問題導向完整且有效的解決方案 | ✓     | ✓     |
| 到場服務          | 將與客戶的技術團隊安排到場時間，進而了解客戶環境和業務需求的變化。             | 4次    | 12次   |
| 月度工作報告        | 每月提交彙總一個月的工作內容和出現的問題故障報告                      | x     | ✓     |
| 季度工作報告        | 每季提交彙總一個季度的工作內容和出現的問題故障報告                     | ✓     | x     |
| 問題匯總報告        | 每月提交彙總一個月的工作內容和出現的問題故障報告                      | ✓     | ✓     |



04

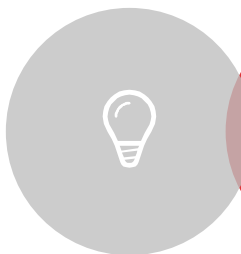
# ROSEN 系統管理平台功能介紹

THE INTRODUCTION OF ROSEN PLATFORM

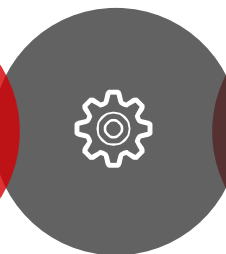
# 為何選擇 ROSEN ?

容器暨混合雲系統管理平台

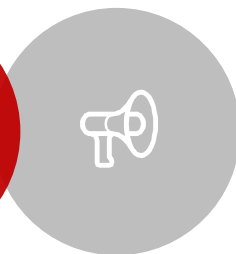
簡單界面更容易維運管理多座 KUBERNETES



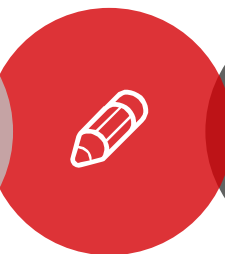
即時監控應用程式、叢集健康狀態機制



即時問題追蹤機制狀態



透過全局動態系統拓樸圖示



權限管理功能符合企業使用情境



ROSEN 容器暨混合雲系統管理平台提供企業/機關的 IT 資訊技術單位或是應用開發部門以最靈活、全局的服務視角進行全方位管理平台，不管是在公有雲、私有雲以及實體主機均可以主動式管理，有效確保 SLA，資源的利用率，並可整合持續整合、持續部署(CI/CD)以及組態合規。



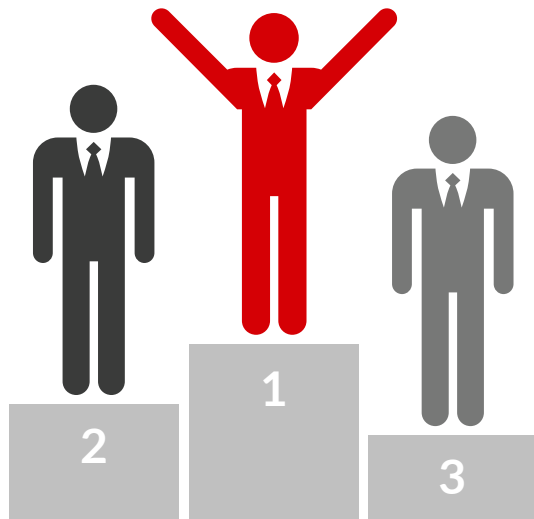


# ROSEN 容器暨混合雲系統管理平台

推出獨步全球的「全局拓撲視圖技術」

ROSEN 微服務管理的解決方案，擁有多叢集容器（Multiple Cluster）的平台架構優勢，在未來管理巨量微服務的需求上，能更大幅簡化於管理跨多個容器叢集、應用程式、服務及鏡像倉儲；且能協助企業中多雲的環境在使用與管理上更充分運用 Kubernetes。

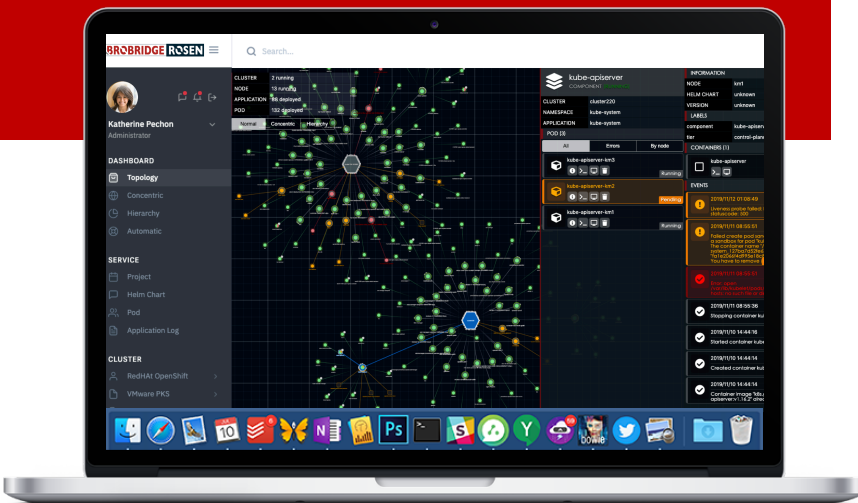
只需透過 ROSEN 即可大幅縮短 Kubernetes 叢集的繁瑣部署與維運管理作業過程，同時可經由應用佈署管理而降低運營成本、更具靈活效益，讓企業能在數位科技的微服務時代裡，快速上手管理工具平台。



# ROSEN 產品功能

HYBRID CLOUD MANAGEMENT SYSTEM

## 容器暨混合雲系統管理平台 BROBRIDGE ROSEN



- 全局動態系統拓撲視圖
- 降低同時維護管理多座 Kubernetes 叢集複雜度
- 透過Web管理操作介面
- 叢集健康狀態監控機制
- 應用程式健康狀態顯示
- 問題即時追蹤機制
- 降低維護管理的難度
- 多專案管控機制、多租戶、多使用者，更能整合各種帳戶認證機制
- 可執行權限設定與角色層級的權限控管
- 權限管理功能使其更符合企業法規遵循與稽核管理使用情境

容器平台與微服務應用架構已遍及台灣各大型企業 IT 資訊技術單位或相關軟體開發部門。  
ROSEN 以全局的服務視角進行容器平台全方位管理，不管是在公有雲、私有雲以及實體主機均可主動管理，有效確保 SLA，資源的利用率，並可介接持續整合、持續部署(CI/CD)以及法規稽核與組態合規。

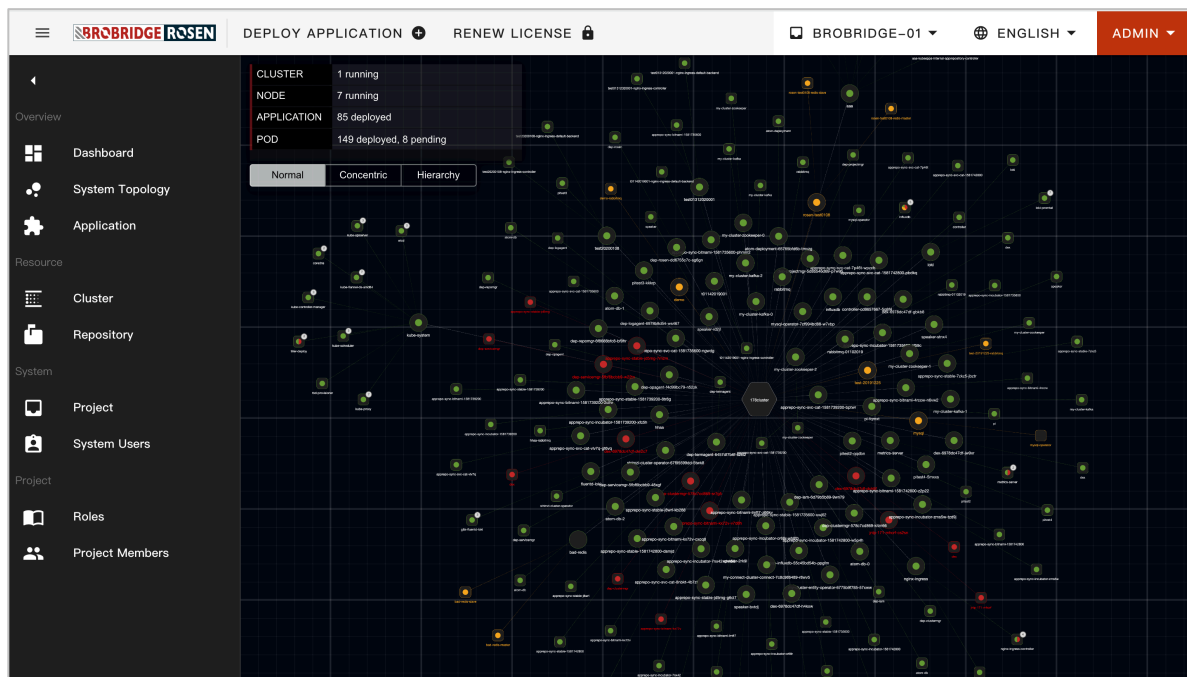


# 全局動態系統拓撲視圖

TOPOLOGY VIEW OF WHOLE MICROSERVICE SYSTEMS

顯示整合跨多叢集的及時系統拓撲資訊，並能即時瀏覽每個節點(NODE)、應用程式和服務元件的所有狀態。

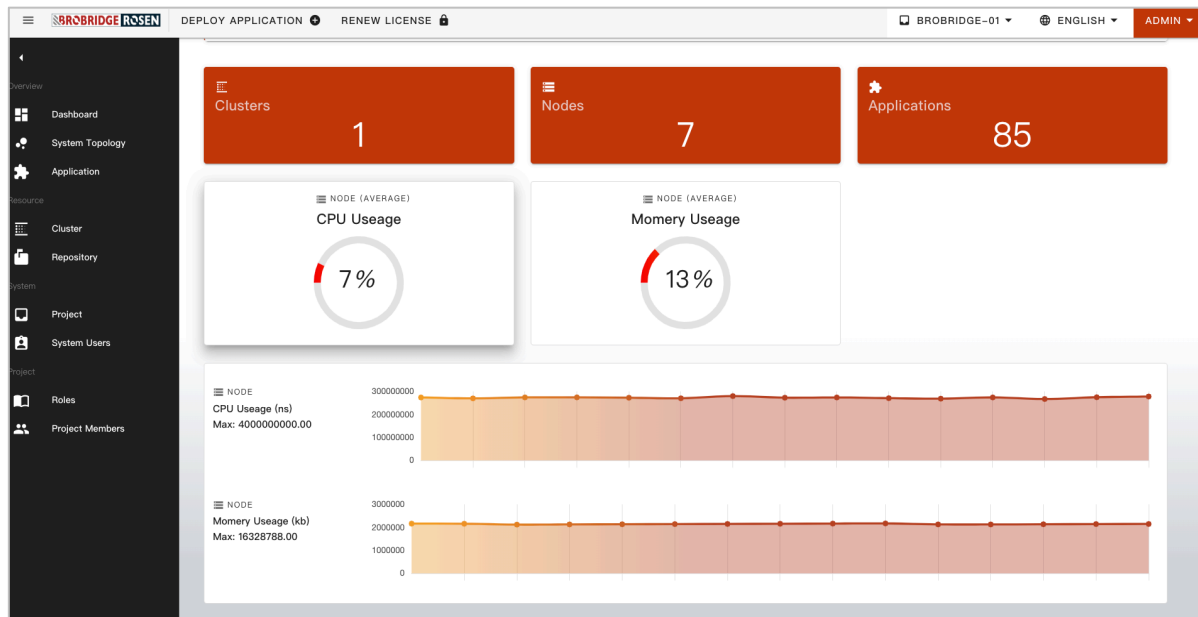
在視圖上查看容器平台與微服務系統之間的狀態，快速定位與追蹤有問題的元件，確保叢集已跟上最新安全性與維護更新狀態。



# 圖形化容器管理平台

GRAPHIC INTERFACE FOR CONTAINER PLATFORM MANAGEMENT

- ▶ 提供了跨多叢集的及時系統拓撲資訊，且ROSEN平台可將Helm或相關管理設定檔的工具匯集，讓各個應用服務中的各種元件只需要透過URL簡單操作，便能瀏覽所有節點與應用程式和服務元件的狀態。
- ▶ 在平台上透過圖示監控集中管理機制，讓維運人員與管理者透過圖形化管理介面顯示節省時間在自行管理Kubernetes繁瑣的步驟。



# 監控應用程式健康狀態與問題追蹤

APPLICATION MONITORING / HEALTH STATUS AND ISSUE TRACKING

在應用程式介面可隨時監控所有叢集節點之間的健康狀態，並且提供適當監控系統機制，即時確認事件發生狀態，更能迅速解決問題事件。

減少人力的耗用，還降低營運成本，並且能事先偵測到系統元件故障狀況並及時修復，避免系統服務中斷或管理者資料損失。

The screenshot displays a web interface for monitoring applications. At the top, there is a navigation bar with the logo 'BROBRIDGE ROSEN', buttons for 'DEPLOY APPLICATION', 'RENEW LICENSE', and a dropdown for 'BROBRIDGE-01'. There are also options for 'ENGLISH' and 'ADMIN'. The main content area is titled 'Applications 85' and includes a search bar. Below this is a table with columns for Name, Namespace, Status, Cluster Name, Helm Chart, Pods, and Update Time. One application is listed with a 'SUCCESS' status. Below the table, there is a detailed view for a specific pod, showing its name, namespace, cluster, and status. At the bottom, there are options to view 'CONTAINERS', 'LOG', and 'TERMINAL' for the selected pod.

| Name                                  | Namespace | Status    | Cluster Name | Helm Chart | Pods | Update Time         |
|---------------------------------------|-----------|-----------|--------------|------------|------|---------------------|
| apprepo-sync-svc-cat-1581746400-vcfwr | aaa       | ● SUCCESS | 178cluster   | unknown    |      | 02/15/2020, 2:00 PM |

**POD 1**  
apprepo-sync-svc-cat-1581746400-vcfwr CONTAINERS

Cluster: 178cluster    Namespace: aaa    Helm Chart: unknown    Helm Chart Version: unknown  
Status: ● Succeeded    Creation Time: 02/15/2020, 2:00 PM

**CONTAINER 1**  
sync LOG TERMINAL



# 多專案管控與使用者管理功能

MULTIPLE PROJECT SUPPORT AND USER'S MANAGEMENT

在平台上能支援多租戶完全隔離各租戶的專案範圍、將各專案加入多個使用者，完全實現軟硬體共用最大化作用，更是提供租戶安全性服務。

整合 Active Directory 之帳戶認證機制，可透過安全管理控制方法用來限制使用者訪問系統和資源，將權限分配給不同的角色。

The screenshot displays the 'Projects' management page in the Brobridge Rosen system. The page includes a navigation bar with 'BROBRIDGE ROSEN', 'DEPLOY APPLICATION', and 'RENEW LICENSE'. The main content area shows a table of projects with the following data:

| Project Name   | Description            | Activate                            | More | Updated Time         |
|--|------------------------|-------------------------------------|------|----------------------|
| <input type="checkbox"/> Brobridge-02                | Private control        | <input type="checkbox"/>            | ...  | 02/18/2020, 10:40 AM |
| <input type="checkbox"/> Digitalocean-k8s-management |                        | <input type="checkbox"/>            | ...  | 02/18/2020, 10:39 AM |
| <input type="checkbox"/> AWS-k8s-manager             |                        | <input type="checkbox"/>            | ...  | 02/18/2020, 10:39 AM |
| <input type="checkbox"/> Project-201224              | My first project       | <input checked="" type="checkbox"/> | ...  | 02/18/2020, 10:38 AM |
| <input checked="" type="checkbox"/> BB-191212        | Brobridge test manager | <input type="checkbox"/>            | ...  | 02/18/2020, 10:37 AM |
| <input type="checkbox"/> Project-200101              |                        | <input type="checkbox"/>            | ...  | 02/18/2020, 10:37 AM |
| <input type="checkbox"/> BB-02                       | For test members       | <input checked="" type="checkbox"/> | ...  | 02/18/2020, 10:35 AM |
| <input type="checkbox"/> BROBRIDGE-01                |                        | <input type="checkbox"/>            | ...  | 12/24/2019, 3:21 PM  |



# 四種常見應用情境與挑戰

4 USE CASES

## 快速定位故障



- 依賴原始的 kubectl 命令和相關工具。
- 每個管理終端需安裝設定叢集工具環境。
- 維運人員需要自行處理原始的故障訊息。
- 依賴繁瑣命令操作叢集來找出問題。
- 不易從日誌系統查找服務關聯性。

業務挑戰#1

## 部署狀態驗證



- 依賴原始的 kubectl 命令和相關工具。
- 每個管理終端需安裝設定叢集工具環境。
- 不易從有巨量服務的叢集中尋找特定目標元件。
- 需切換不同工具來確認部署的成果和狀態。

業務挑戰#2

## 混合雲多叢集管理



- 實現統一的管理機制相當困難。
- 缺少跨雲多叢集的管理工具。
- 多叢集維運操作不易。
- 無法同時管理並監管混合雲架構的系統元件狀態。

業務挑戰#3

## 微服務全局拓樸監控



- 巨量服務難以時刻關注所有狀態。
- 錯綜複雜的微服務問題難以追蹤。

業務挑戰#4

# 情境比較一：快速定位並排除故障

THE COMPARISON BETWEEN WITH / WITHOUT ROSEN SOLUTION - FAST TO LOCATE THE PROBLEM AND SOLVE

## 無 ROSEN 平台時的困難

- 依賴原始的 kubectl 命令和相關工具
- 每個管理終端需安裝設定叢集工具環境
- 維運人員需要自行處理原始的故障訊息
- 依賴繁瑣命令操作叢集來找出問題
- 不易從日誌系統查找服務關聯性
- 倚賴原始的操作命令才能進入容器 Console
- 不易從有巨量服務的叢集中尋找特定目標物件
- 除錯過程需切換不同工具，處理流程上較不方便

## 有 ROSEN 平台時的好處

- 操作流程全部有圖形化介面
- 免安裝、免設定，透過瀏覽器隨開即用
- 從圖型管理介面上快速查看錯誤原因
- Prometheus 的錯誤訊息可直接關聯管理介面
- 可從系統拓樸快速定位故障的 Pod
- 可直接開啟並查看容器之即時 Log
- 過程中無需下指令或切換不同工具

麻煩費時

導入前

導入後

簡單快速





## 情境比較二：部署狀態驗證

THE COMPARISON BETWEEN WITH / WITHOUT ROSEN SOLUTION - DEPLOYMENT VERIFICATION

### 無 ROSEN 平台時的困難

- 依賴原始的 kubectl 命令和相關工具
- 每個管理終端需安裝設定叢集工具
- 不易從有巨量服務的叢集中尋找特定目標元件
- 需切換不同工具來確認部署的成果和狀態

麻煩費工

導入前

### 有 ROSEN 平台時的好處

- 免安裝、免設定，透過瀏覽器隨開即用
- 圖型管理介面上快速查看並調整
- 操作簡單，無需學習多種工具
- 減少部署完成後的驗證時間

導入後

簡單快速



# 情境比較三：混合雲多叢集管理

因應不同業務、維運層面的分散、分治、分流等需求，必然引入跨雲、多叢集的解決方案。

THE COMPARISON BETWEEN WITH / WITHOUT ROSEN SOLUTION - HYBRID CLOUD AND MULTIPLE CLUSTER

## 無 ROSEN 平台時的困難

- 實現統一的管理機制相當困難
- 缺少跨雲多叢集的管理工具
- 多叢集維運操作不易
- 無法同時管理並監管混合雲架構的系統元件狀態

缺少方案

導入前

## 有 ROSEN 平台時的好處

- 支援跨雲管理
- 可同時進行多叢集、雲端與地端的資源管理
- 提供圖形化操作介面，方便維運操作
- 統一介面收集各個叢集資訊
- 視覺化呈現跨雲、跨叢集的即時全局狀態

導入後

全局管理



## 情境比較四：微服務全局拓樸監控

THE COMPARISON BETWEEN WITH / WITHOUT ROSEN SOLUTION - MONITORING AND MANAGEMENT WITH MICROSERVICE TOPOLOGY SYSTEM

### 無 ROSEN 平台時的困難

- 巨量服務難以時刻關注所有狀態
- 錯綜複雜的微服務問題難以追蹤

### 有 ROSEN 平台時的好處

- 全局微服務拓樸視圖，掌控系統全貌
- 降低巨量服務的狀態監控難度
- 所見即所得的管理和維運操作
- 能快速定位和追蹤微服務的問題
- 能快速微調和調配微服務部署任務

不易維運

導入前

導入後

掌握全局



05

# ROSEN 微服務管理模組

ROSEN MODULES FOR MICROSERVICE MANAGEMENT

# BROBRIDGE ROSEN 交易管理追蹤與微閘道管理模組

BROBRIDGE ROSEN EXTENTION FOR TRANSACTION TRACING & MICRO API GATEWAY MANAGEMENT

## ■ Software-Defined Gateway Manager

- 實現 Service Mesh 的連線控管機制和管理介面
- 能夠觸及 Rosen 所監控的叢集節點（不限單一叢集），進行 API Gateway 的部署和管理
- 基於標準 Istio Ingress 的 API Gateway 管理

## ■ Transaction Tracer

- 垂直整合 Infra 層到服務層的協定（例如：節點事件、Metrics、服務狀態、連線到交易連線狀態）
- 支援 OpenTracing 協定實現應用程式層的交易追蹤管理
- 提供交易追蹤監控介面
- 內建分散式交易解決方案（Twist）

- David #1 API Gateway + Service Mesh = Gateway Manager ◦
- 嘉哲 #1 提供對應的“application development framework”來進行AP使用上的落地 ◦ Transaction Tracer + Gravity



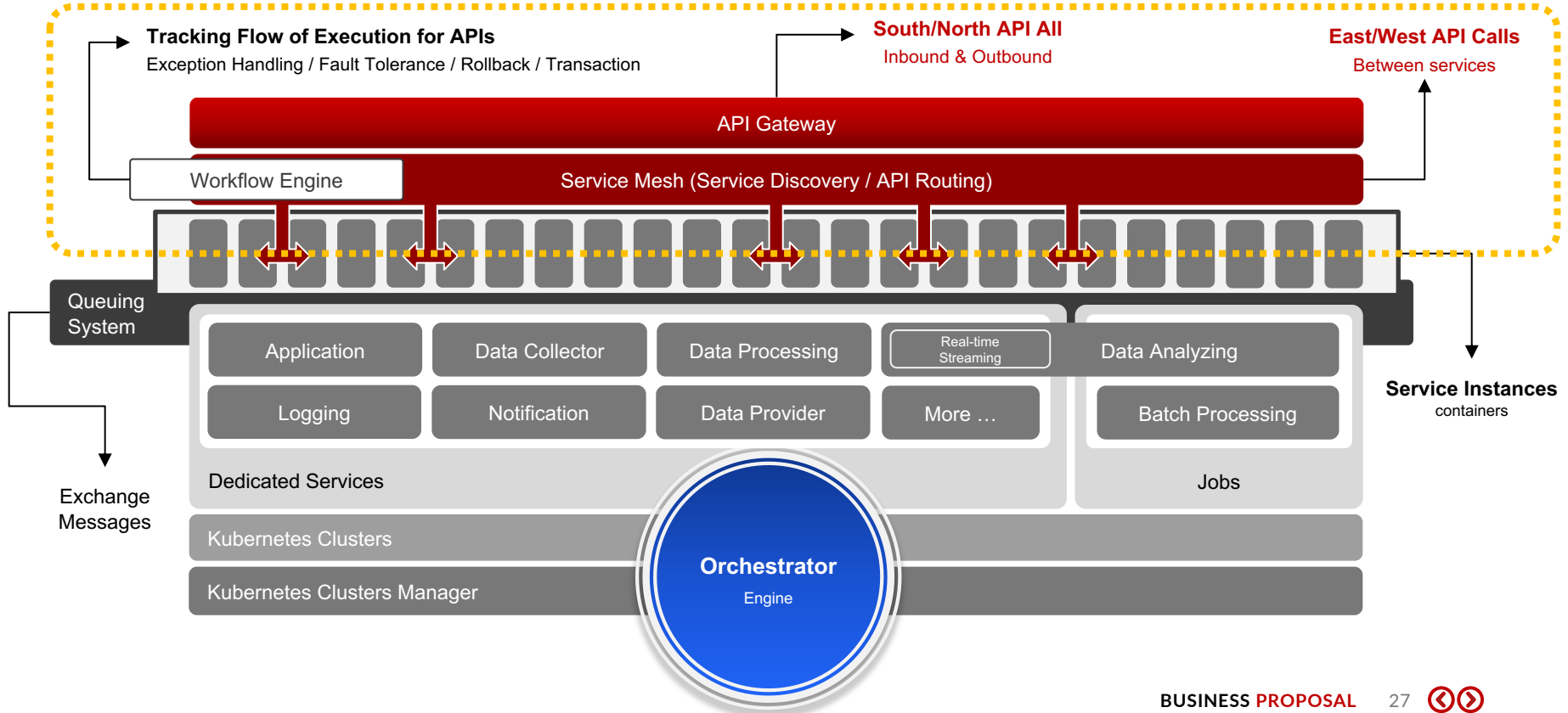


## **SOFTWARE DEFINED GATEWAY MANAGER**

A Service Mesh based gateway manager for the microservice running on Kubernetes

# API GATEWAY 在微服務架構中所扮演的角色

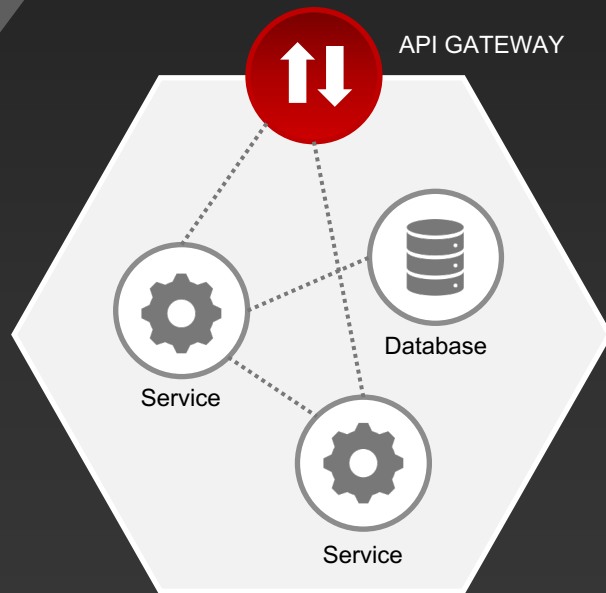
API GATEWAY ON MICROSERVICE ARCHITECTURE



# 為什麼需要 API GATEWAY ?

API 的對外管控窗口

- 整合內部微服務架構
- 提供統一對外的 API 存取介面
- 可整合授權驗證
- API 管理和控制
- API 存取路徑和方法控管
- 導流量機制
- 整合 Service Mesh 的管理機制

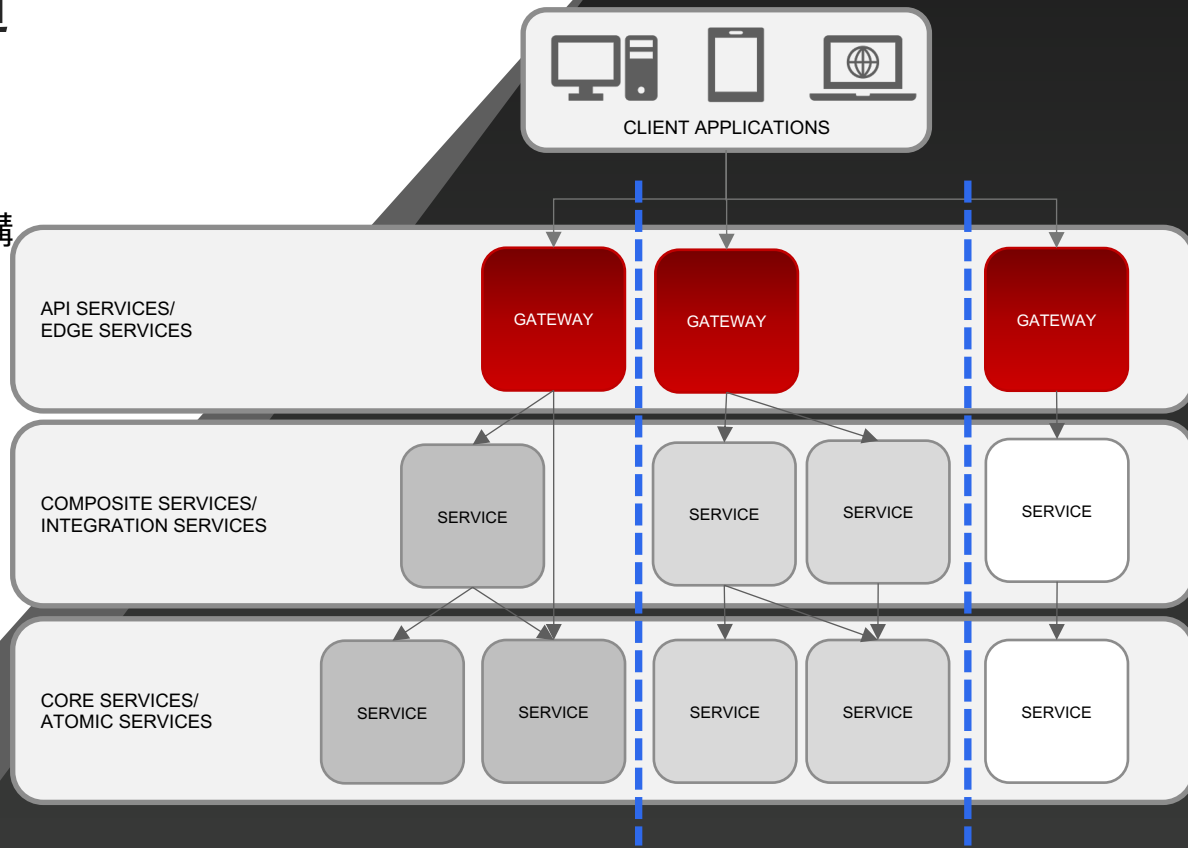




# 分段服務下的微閘道

GATEWAY FOR SEGMENTED MICROSERVICE

- 邊緣服務
- 實際上可遍佈於整個微服務架構
- 服務與其他元件間的溝通介面
- 可依照需求快速部署
- Service Mesh 的連線管控節點



# 管理和部署 API GATEWAY 的困難和需求

GATEWAY 的管理是一件麻煩且需要多元整合的工作

**Gateway**  
Management



散落在容器平台之中



缺少統一管理的介面和機制



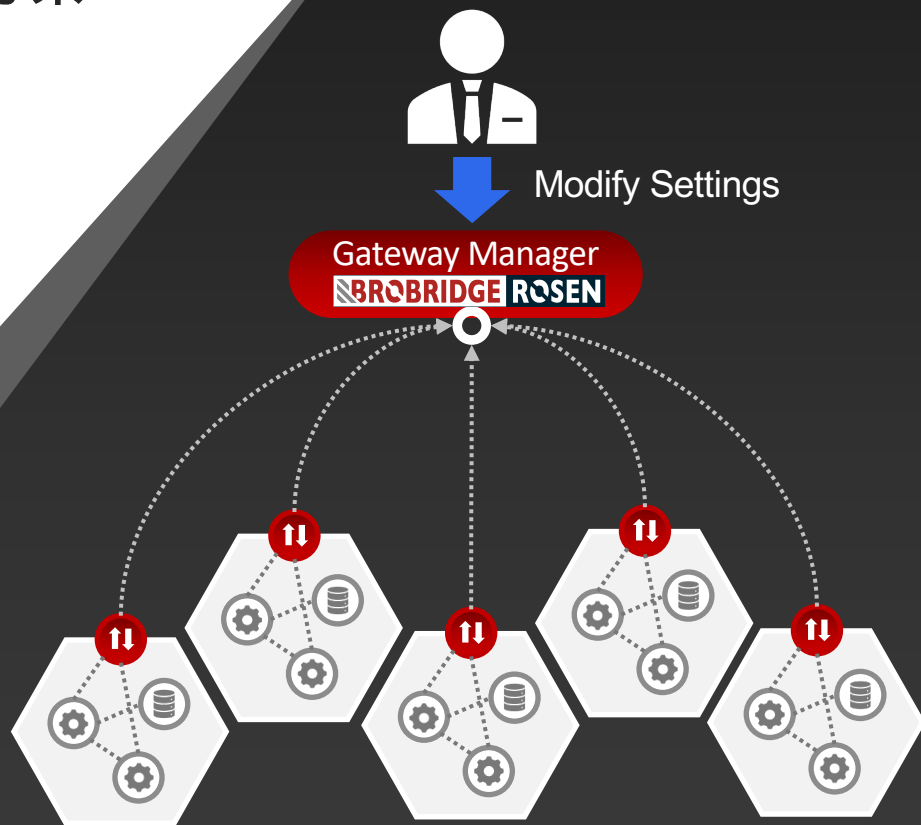
需要更多 Service Mesh 的管理整合



# GATEWAY MANAGER 解決方案

微閘道管理模組

- 軟體定義的 API Gateway
- 快速、彈性、容易部署
- 整合稽核管理，強化資訊安全
- 集中管理容器平台上所有的閘道元件
- 提供統一的管理和監測介面
- 具備監控事件可進行第二次應用開發
- 整合並支援 Service Mesh
  - 支援並相容 Istio 解決方案
  - 實現東西向的網路連線管控



# GATEWAY MANAGER 特色

THE BENEFITS OF GATEWAY MANAGER

## 軟體定義，快速部署

提供軟體定義的 API Gateway 元件，能依照需求，動態且快速的建立並部署。

## 基於 Service Mesh 的管理機制

相容於 Istio 解決方案，快速實現服務網格架構，監控內部服務間的流量和存取狀態，並實現加密連線等需求。



## 支援第三方授權認證機制

可整合第三方授權認證機制，實現 OAUTH 授權政策、API 連線限制等多層級的存取控管機制。

## 簡化 API 管理工作

能集中管理、監控以及設定 API 等相關行為，並且能記錄 API 的存取狀態。

- David #2 Message Queue 及 Message Pub/Sub 機制。 NATS + KAFKA
- David #3 Service Mesh。





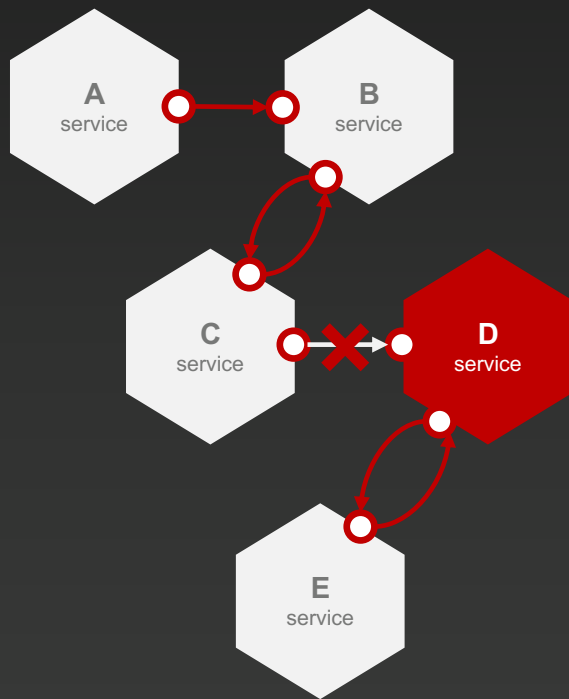
## **TRANSACTION TRACER**

A Business Transaction Tracer for the microservice running on Kubernetes

# 微服務下的交易追蹤需求

跨服務的交易追蹤

- 跨服務間的任務處理流程
- 每個 API 呼叫的狀態和日誌追蹤
- 交易事件處理機制
- 整合應用層的交易機制實現
  - 提交 (Commit)
  - 回滾 (Rollback)



# 微服務交易追蹤的困難

分散式架構下的訊息追蹤

**Transaction  
Tracing**



跨服務監控並關聯每一筆交易狀態



需支援不同的訊息交換和 API 呼叫

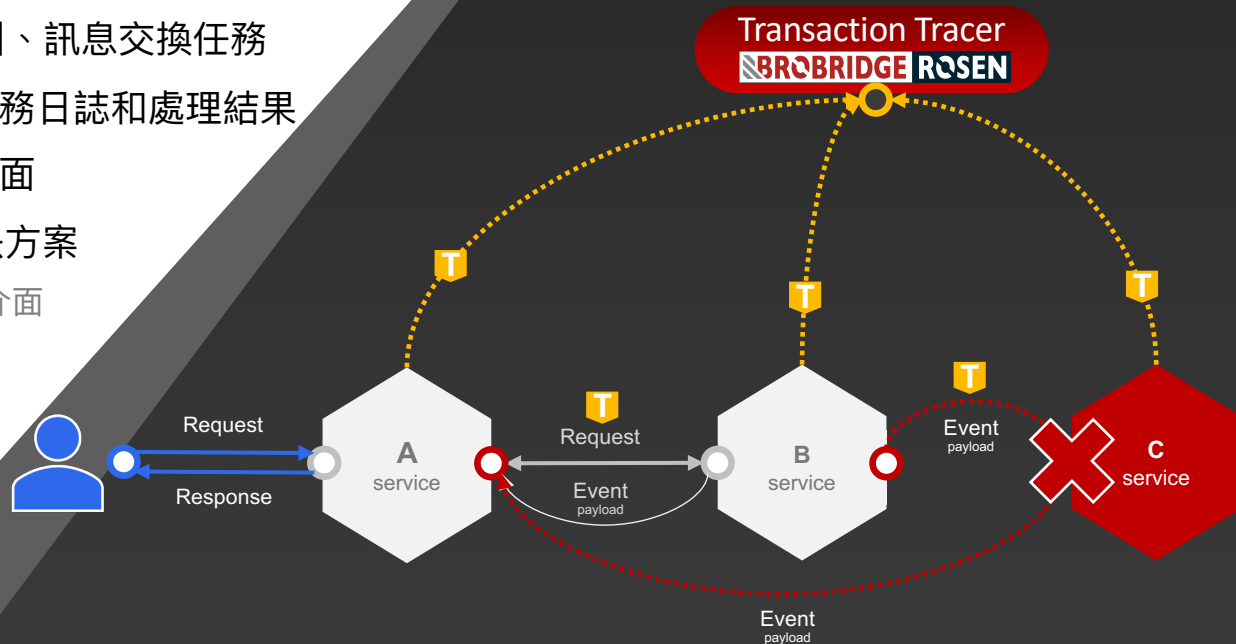


需支援不同語言、框架所開發的服務

# TRANSACTION TRACER 解決方案

交易追蹤模組

- 支援標準 OpenTracing 通訊協定
- 支援多種程式語言
- 訊息貼標以關聯 API 呼叫、訊息交換任務
- 可調閱交易流程中的各任務日誌和處理結果
- 提供統一的管理和觀察介面
- 內建 Twist 交易機制解決方案
  - 提供交易機制標準開發介面





# TRANSACTION TRACER 特色

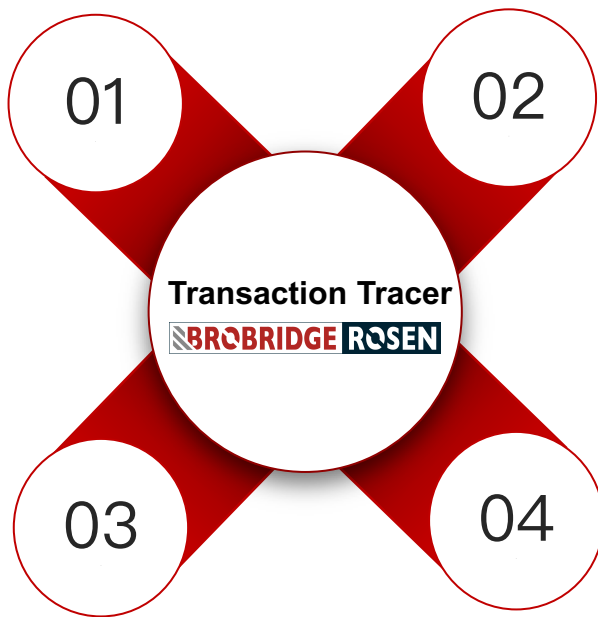
THE BENEFITS OF TRANSACTION TRACER

## 交易流程監控與日誌收集

提供統一介面，收集所有交易流程的狀態和當下日誌，亦可將同一筆交易內之任務資訊關聯起來。

## 相容 Service Mesh 機制

相容於 Istio 解決方案，能和服務網格的網路監控混搭使用，取得更完整的系統資訊。



## 支援多種程式語言和通訊方法

支援標準 OpenTracing 協定，能在多種程式語言、框架中進行任務貼標，並可以在不同的通訊方法中實現。

## 提供事件輸出，方便客製化功能

提供交易行為的事件輸出，以便整合其他系統或客製化更多功能。



06

# 微服務開發解決方案

BROBRIDGE MICROSERVICE ARCHITECTURE SOLUTION

# 微服務架構的兩大痛點

PAIN POINTS OF MICROSERVICE IMPLEMENTATION



# BROBRIDGE ROSEN 分散式交易和 CQRS 開發解決方案

BROBRIDGE ROSEN EXTENTION FOR DISTRIBUTED TRANSACTION & CQRS

## Twist

### Distributed Transaction Solution

- 部署即用
- 實現 TCC (Try-Confirm-Cancel) 模式
- 跨語言或應用程式框架
- 支援交易紀錄、追蹤
- 支援 Coordinator 橫向擴展
- 支援容錯和交易狀態復原
- 支援交易逾時機制

## GRAVITY

### CQRS Solution

- 部署即用，免開發
- 可擴充支援多種事件來源和通訊協定
- 可擴充支援多種資料庫系統
- 支援事件遺失容錯和資料還原
- 可客製化事件與資料過濾規則
- 可快速橫向擴展資料庫



**Twist**

A Transaction Solution for the microservice running on Kubernetes

# 為何採用 TWIST ?

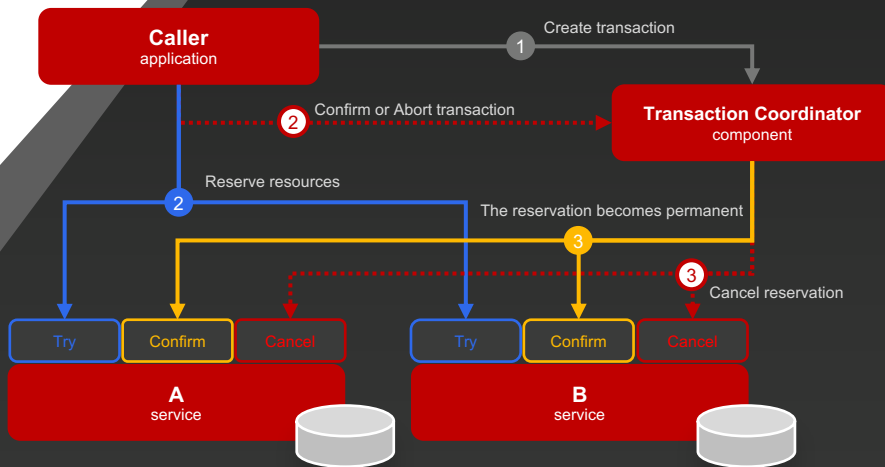
開箱即用的微服務交易機制解決方案



# TCC 交易模式

應用層的分散式交易機制實現

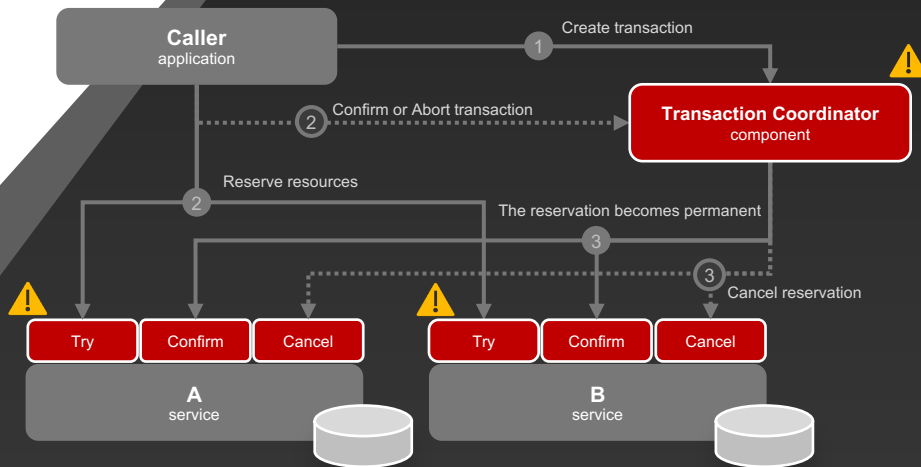
- Try-Confirm-Cancel (TCC)
- 應用層的 Two-Phase-Commit (2PC)
- 以 Coordinator 來協助並管控交易進行
- 主要執行流程：
  - Try: 預留交易資源
  - Confirm: 確定交易執行
  - Cancel: 取消交易



# 實現 TCC 時的考量

實務上的各種需求和挑戰

- 交易紀錄追蹤和收集
- Coordinator 的高可用、擴展機制
- 管理交易狀態和生命週期
  - 實現 Stateful 的 Runner 機制
  - 意外發生後的交易狀態回復機制
- 可支援多語言或應用開發框架
- 規範化交易機制界面
- 簡化開發的困難

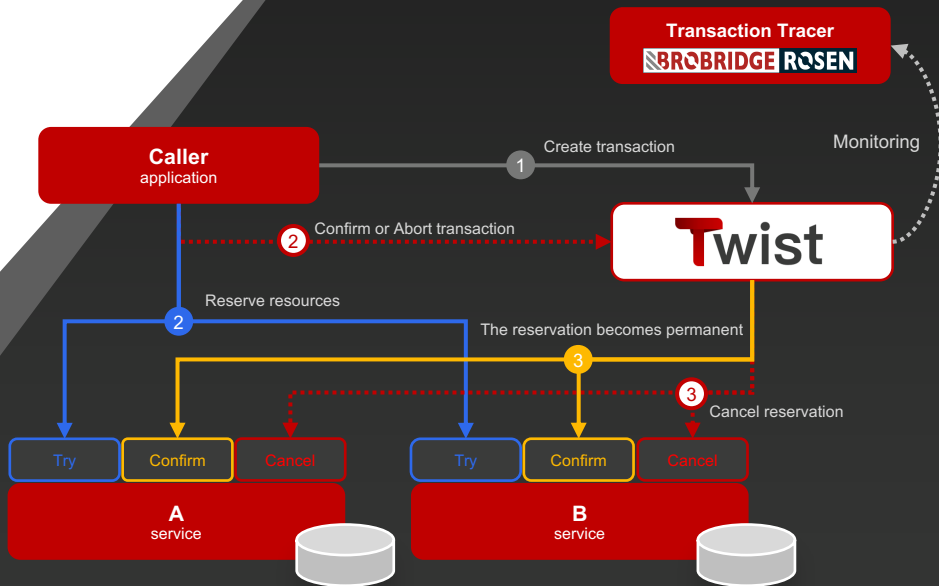




# TWIST SOLUTION

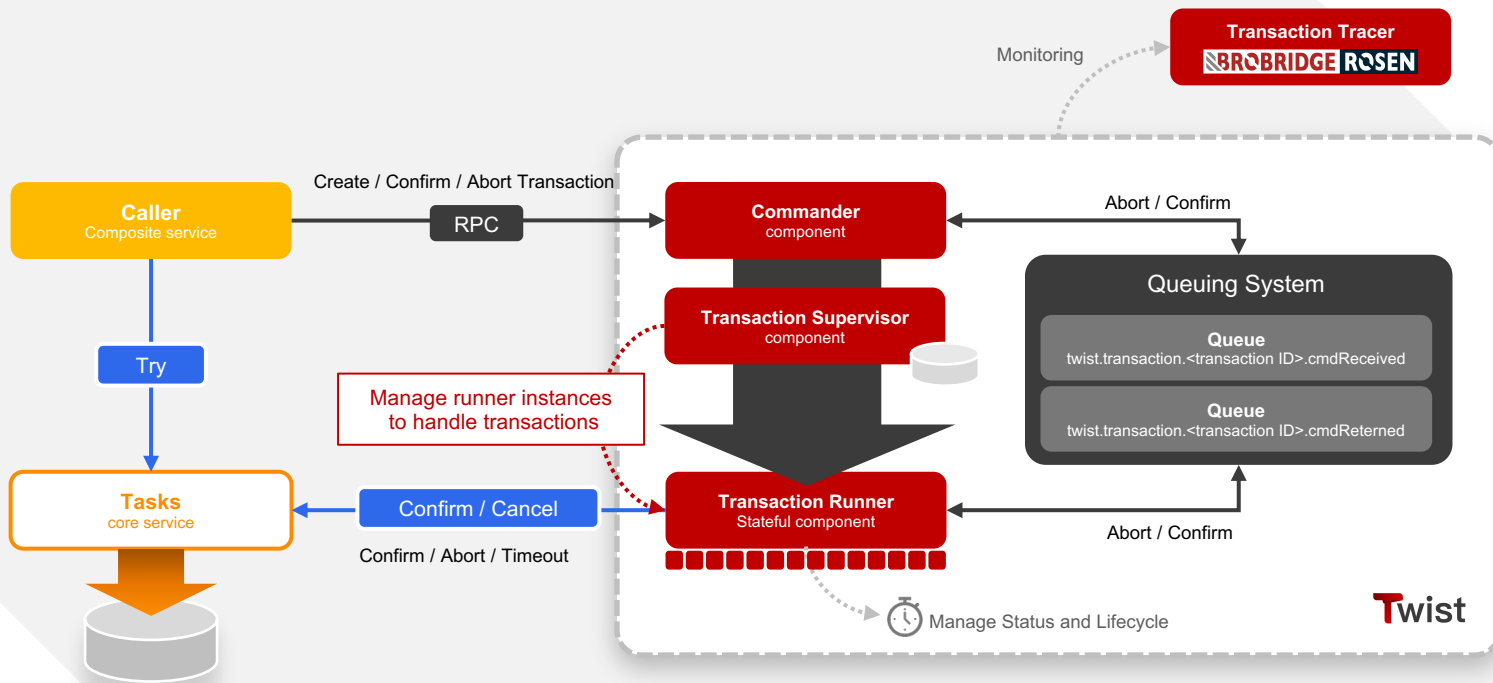
開箱即用的分散式交易機制解決方案

- 部署即用
- 實現 TCC (Try-Confirm-Cancel) 模式
- 跨語言或應用程式框架
  - 以標準 Restful API、gRPC 溝通
- 支援交易紀錄、追蹤
- 支援橫向擴展
- 支援容錯和交易狀態復原
- 支援交易逾時機制



# TWIST ARCHITECTURE

High Available Transaction Coordinator Implementation



# 實現和維運 TCC 的痛點

The Drawback of TCC: implementation and system maintenance

**TCC**  
Pattern



Coordinator 的任務管理機制設計不易



難以保持高穩定性、彈性與擴展性



TCC 叢集系統失效後的狀態復原機制

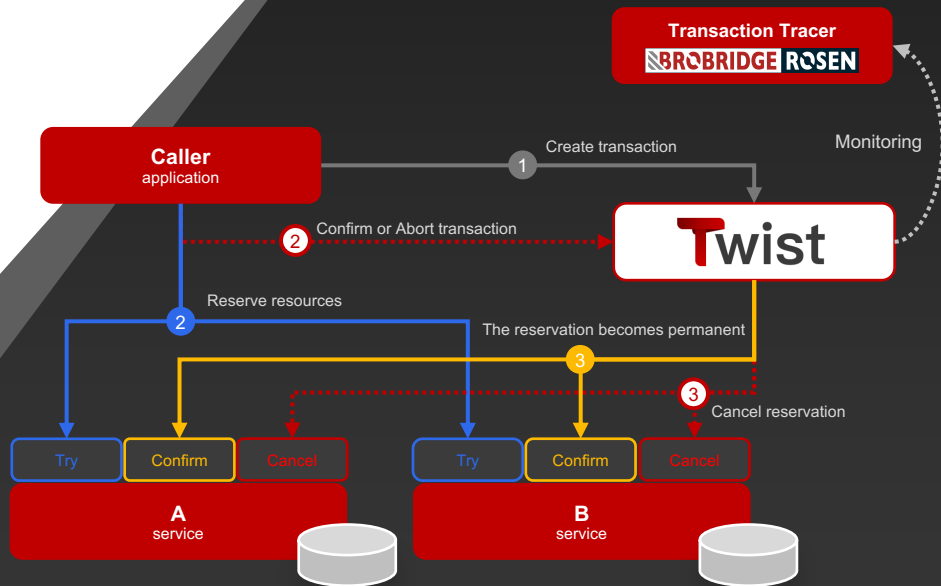


Coordinator 過於集中化，不易分散部署和維運



# 交易 API 介面規範

遵循規範實現交易機制



# TWIST 解決方案優點

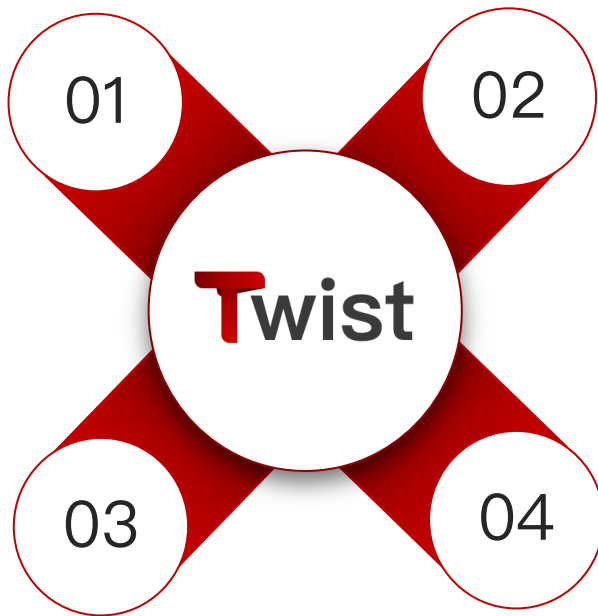
THE BENEFITS OF TWIST

## 開箱即用，無需開發

簡化 TCC 導入工作，無需自己實現系統架構、Coordinator 叢集、狀態復原、溝通介面設計等問題，部署即可使用。

## 不綁定程式語言和開發框架

以通用的通訊方法（如：Restful API、gRPC）實現交易機制，只要遵循API 呼叫的設計規範，無論哪一種程式語言或是開發框架，皆可直接使用。



## 實現應用層分散式交易機制

實現跨服務情境下的交易機制，能滿足微服務架構和分散式系統下的高併發交易需求，及資料一致性的目標。

## 可紀錄追蹤每筆交易

搭配 OpenTracing 協定，並整合 Rosen 管理工具，能調閱每筆交易紀錄，其中包括時間、狀態以及執行日誌資訊等細節。





**GRAVITY**

A CQRS Framework Solution for the microservice running on Kubernetes

# 為何使用 CQRS ?

CQRS 的優點



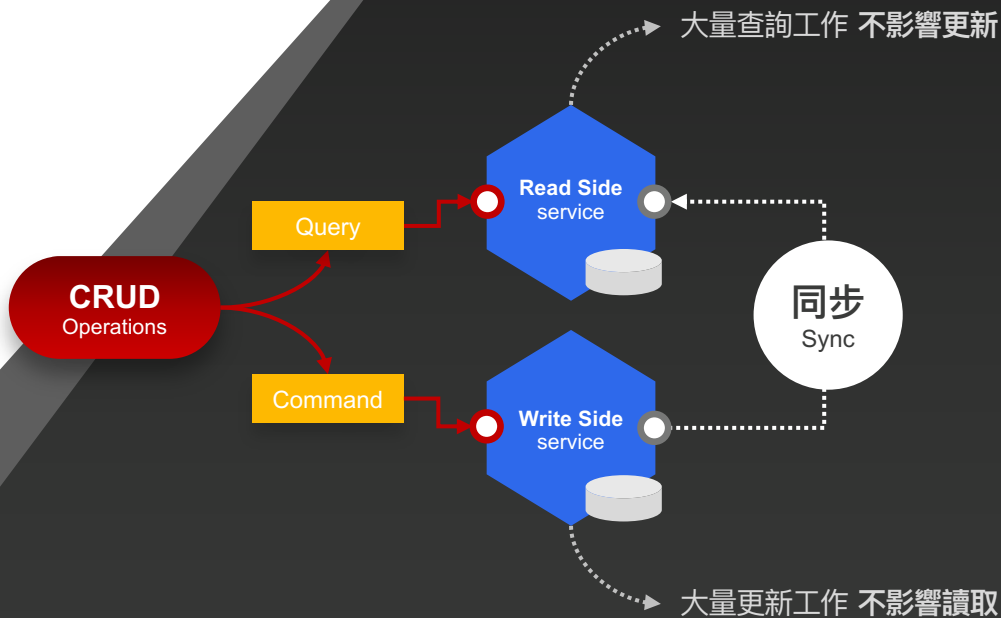
# GRAVITY



# CQRS 命令和查詢權責分離 (讀寫分離)

CQRS, COMMAND AND QUERY RESPONSIBILITY SEGREGATION

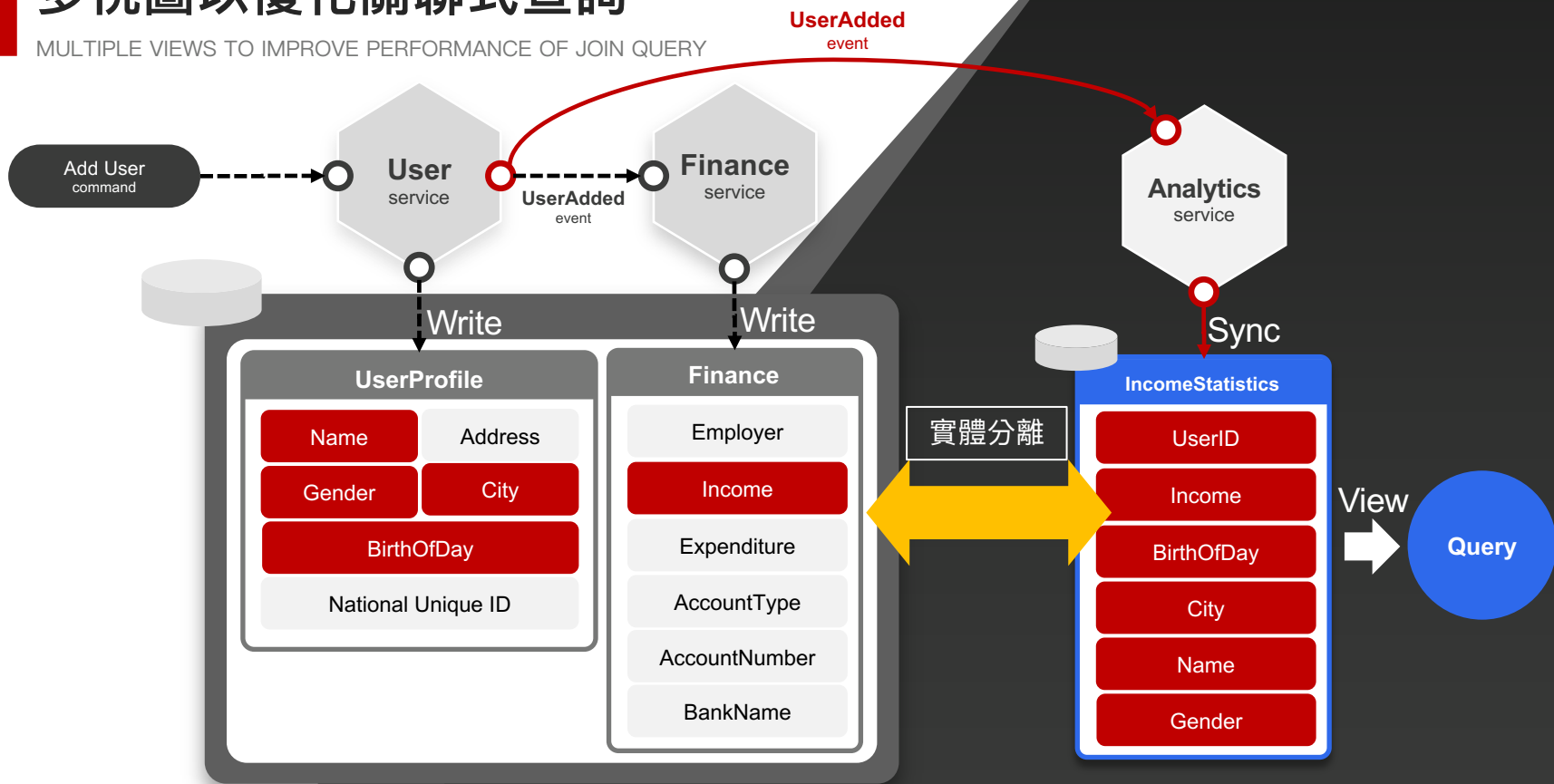
- 寫入變更和讀取動作分開
- 查詢和寫入工作互不影響
- 透過事件去觸發資料庫同步





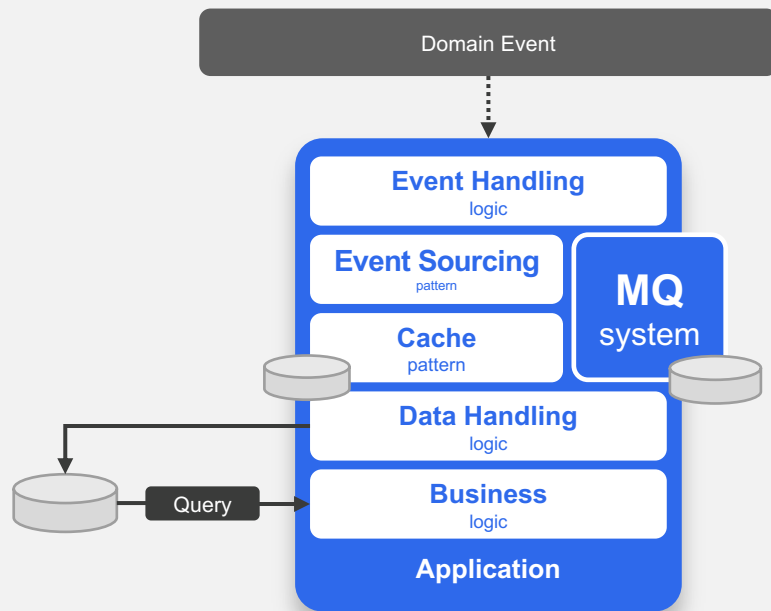
# 多視圖以優化關聯式查詢

MULTIPLE VIEWS TO IMPROVE PERFORMANCE OF JOIN QUERY

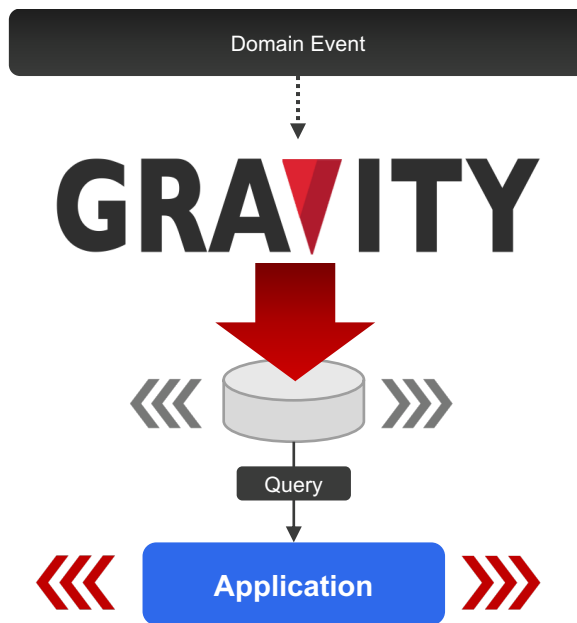


# 自己搭建實作和採用 GRAVITY 的比較

THE COMPARISON BETWEEN HOME MADE AND GRAVITY FRAMEWORK



⚠️ 架構複雜且不易自行實作、維護  
若設計不當，甚至難以橫向擴展應用程式



將資料層於應用層分開，容易維護和擴展  
應用程式專心業務邏輯開發



# 傳統 CQRS 的實現細節

WORKING ITEMS FOR CQRS IMPLEMENTATION

- 搭建並維運訊息佇列系統 (Message Queuing System)
- 考慮資料一致性的解決方法
- 實現事件監聽和處理
- 意外處理事件遺失、容錯、資料回復
  - 實現事件來源 (Event Sourcing)
- 可快速重置、服務擴展，資料回復時避免衝擊其他系統效能
  - 實現資料快照 (Data Snapshot)



# 實現和維運 CQRS 的痛點

The Drawback of CQRS: implementation and system maintenance

## CQRS Pattern



自行實作不易且結構複雜



難以保持高穩定性、彈性與擴展性



長期維護困難



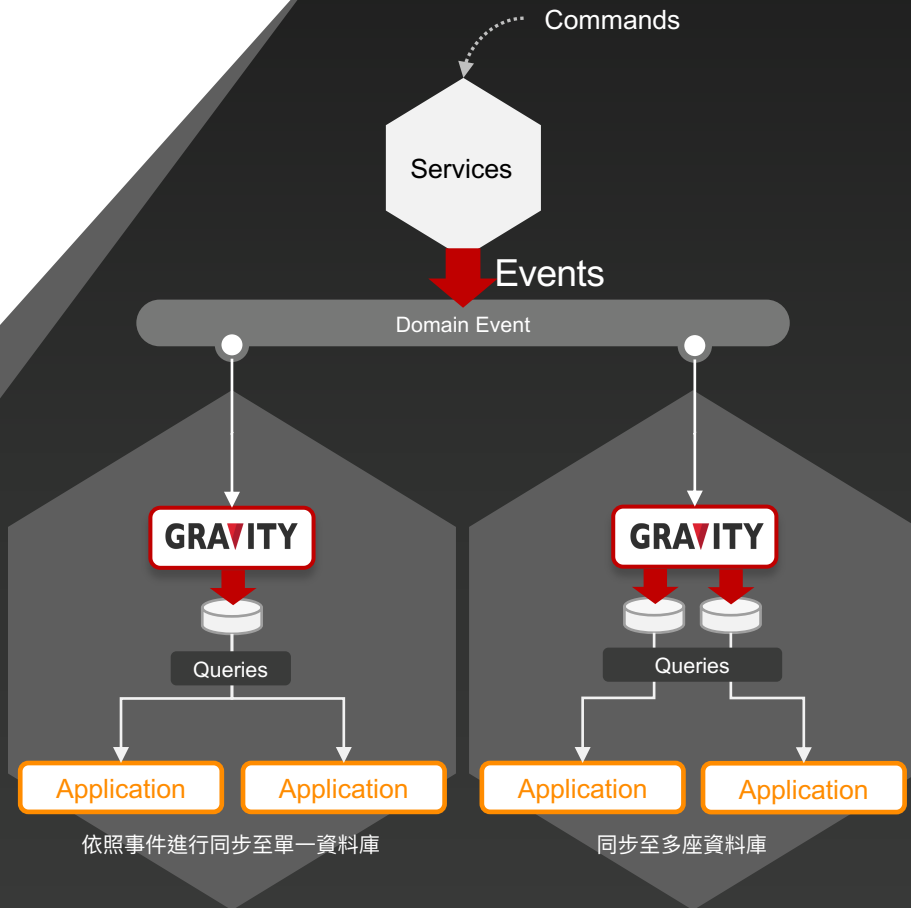
抵觸企業組織權責架構 應用層實現難以被DBA管控



# GRAVITY SOLUTION

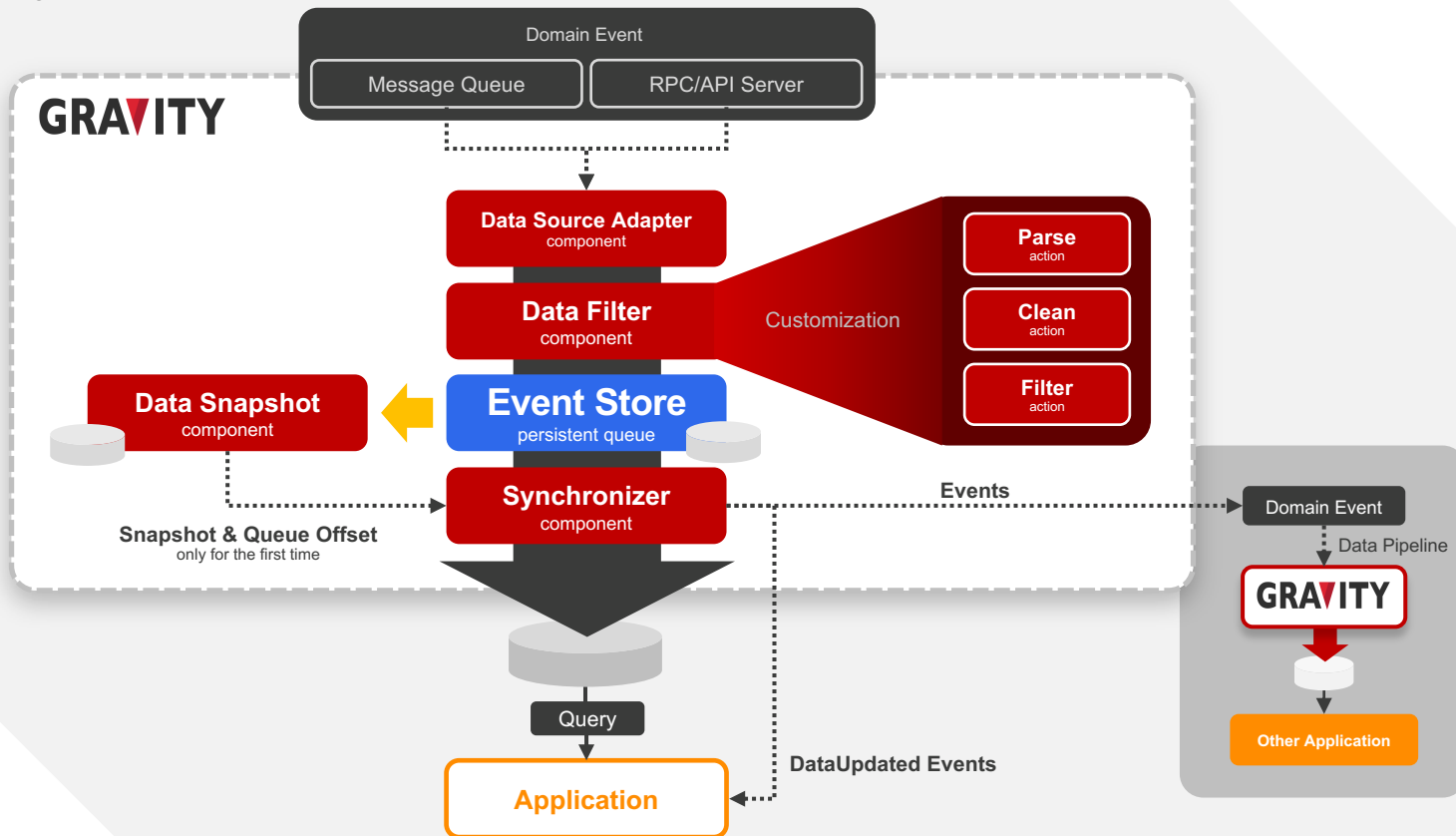
開箱即用的 CQRS 解決方案

- 部署即用，免開發
- 可擴充支援多種事件來源和通訊協定
  - Kafka, NATS, RabbitMQ
- 可擴充支援多種資料庫系統
  - SQL, NoSQL
- 支援事件遺失容錯和資料還原
  - 內建事件來源 (Event Sourcing)
  - 內建資料快照 (Data Snapshot)
- 可客製化事件與資料過濾規則
- 可快速橫向擴展資料庫



# GRAVITY SOLUTION

CQRS IMPLEMENTATION

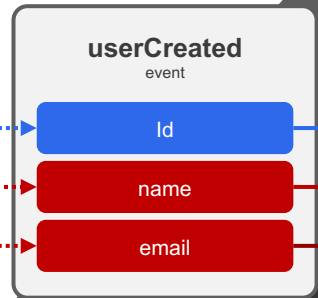


- 嘉哲 #1 提供對應的”application development framework”來進行AP使用上的落地。 Transaction Tracer + Gravity

# 設定事件處理規則以建立視圖

建立事件欄位與資料表欄位的投射關係

```
rules.json
{
  "rules": [
    {
      "event": "userCreated",
      "table": "users",
      "method": "update",
      "mapping": [
        {
          "source": "id",
          "target": "id",
          "primary": true
        },
        {
          "source": "name",
          "target": "col_name"
        },
        {
          "source": "email",
          "target": "col_email"
        }
      ]
    }
  ]
}
```



| id | col_name   | col_email          |
|----|------------|--------------------|
| 1  | Fred Chien | fred@brobridge.com |



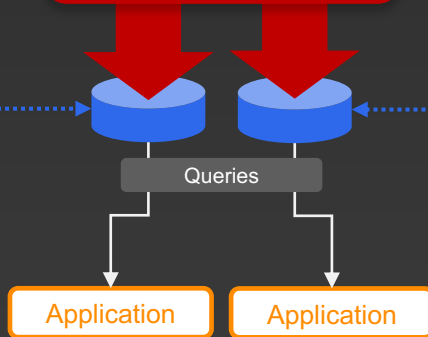
# 在多個資料庫上建立視圖

同一時間，一個事件同步到多個資料庫

```
databases.json
{
  "databases": {
    "db1": {
      "type": "postgres",
      "host": "192.168.1.1",
      "port": 32805,
      "secure": false,
      "username": "postgres",
      "password": "test",
      "dbName": "gravity"
    },
    "db2": {
      "type": "mysql",
      "host": "192.168.1.2",
      "port": 32805,
      "secure": false,
      "username": "mysql",
      "password": "test",
      "dbName": "gravity"
    }
  }
}
```

```
rules.json
{
  "rules": [
    {
      "event": "userCreated",
      "database": "db1"
    },
    {
      "event": "userCreated",
      "database": "db2"
    }
  ]
}
```

**GRAVITY**





# 同步完成觸發事件

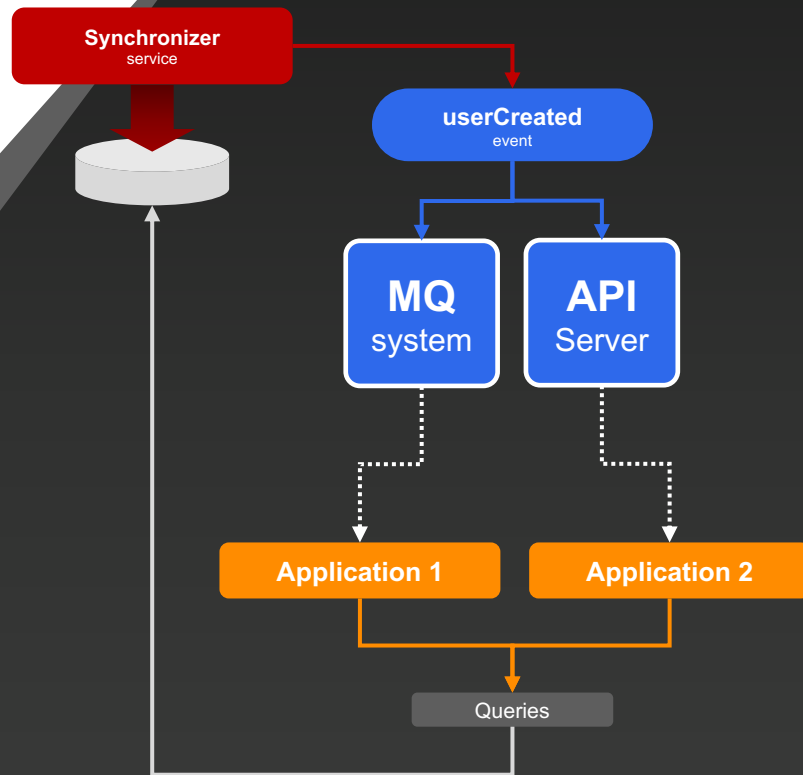
同步完成後拋出事件，供應用程式或其他系統使用

```
exporter.json
{
  "exporter": {
    "exporter1": {
      "type": "kafka",
      "host": "192.168.1.5",
      "port": 32806,
      "secure": false,
      "username": "kafka",
      "password": "test",
      "topic": "customized"
    },
    "exporter2": {
      "type": "restful_api",
      "url": "http://192.168.1.6/api"
    }
  }
}
```

GRAVITY

```
rules.json
{
  "rules": [
    {
      "event": "userCreated",
      "database": "db1",
      "exporter": [
        "exporter1",
        "exporter2"
      ]
    }
  ]
}
```

GRAVITY



# GRAVITY 解決方案優點

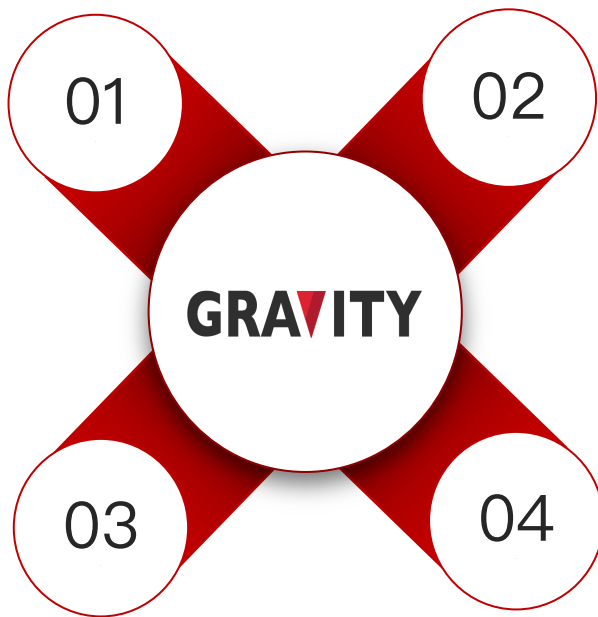
THE BENEFITS OF GRAVITY

## 開箱即用，無需開發

簡化 CQRS 導入工作，無需自己實現系統架構、處理事件、資料同步、復原等問題，部署即可使用。

## 彈性且穩定快速，機制完整

支援多種資料庫、資料源、通訊協定，並內建 Event Sourcing 和 Data Snapshot 兩種機制，保證事件資料不遺失，且能快速還原和複製。



## 支援二次開發

開發者可依照個別需求，撰寫元件以擴充事件、資料篩選機制。資料同步機制，亦有事件可供開發者監聽並進行業務邏輯開發使用。

## 改善資料庫存取效能

快速實現讀寫分離，能輕易選取事件資料和欄位，建立實體分離的視圖資料庫，避免關聯式查詢效能問題。



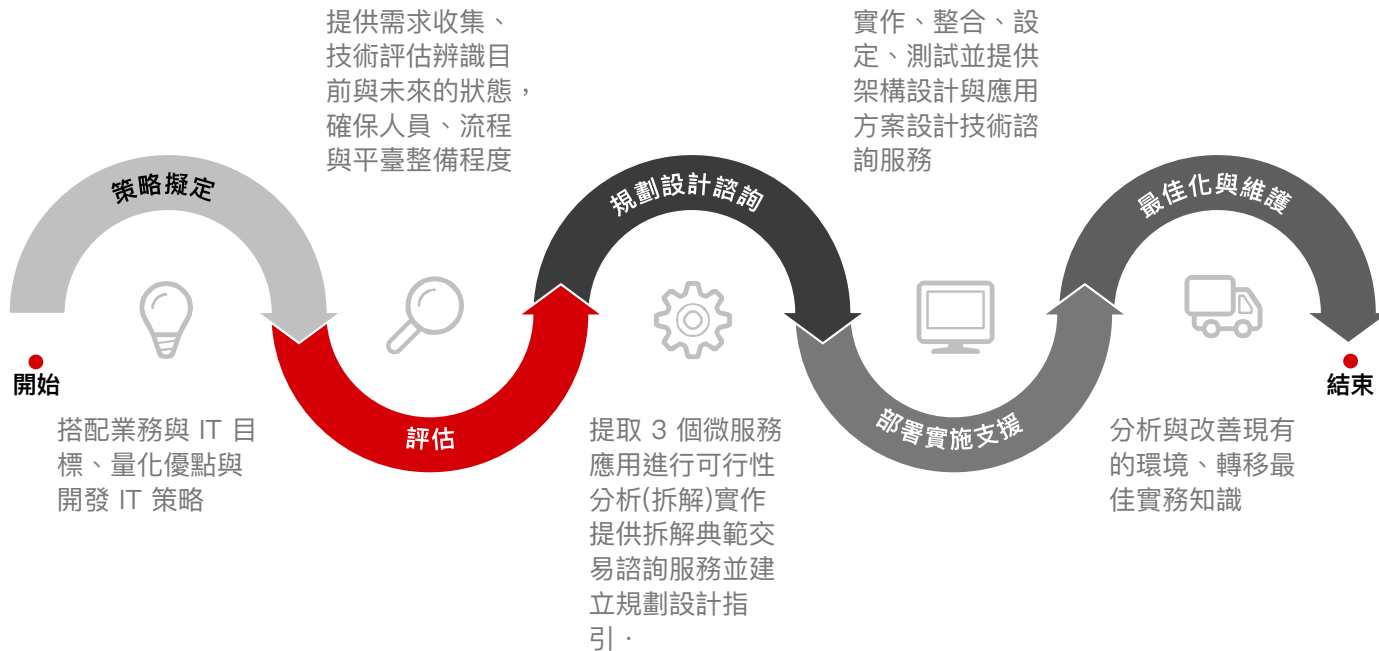
07

# 微服務導入服務說明

BROBRIDGE MICROSERVICE PROFESSIONAL SERVICE

# 微服務導入服務概述

BROBRIDGE MICROSERVICE PROFESSIONAL SERVICE



# 微服務導入服務 — 服務範圍

MICROSERVICE ASSESSMENT PROGRAM – CORE SCOPE

- Up to 9 worker nodes.  
(SIT/UAT/PROD x 3)
- Up to 3 instances' conduction of the Microservice Toolset
  - Gateway Manager
  - Transaction Tracer
  - Gravity – CQRS Manager
- Up to 3 information gathering workshops (容器化管理平台建置顧問諮詢服務)
- Up to 3 application/service decouple workshops (微服務導入顧問服務)
- Microservice Assessment Program stakeholder presentation
- Kubernetes TAM service as below.

| 工作內容          | 描述   | TAM  |
|---------------|--|------|
| 服務時間          | 12 個月 (每週一至週五上班時間、例假日除外)                           | ✓    |
| 每月定期檢查        | 1部 Master + 3 部 Worker K8S 主機系統健康狀況檢查              | ✓    |
| 遠程技術支援        | 電話 / eMail / 問題處理支援服務                              | ✓    |
| 產品問題服務請求及進度追蹤 | 跟蹤用戶需求和故障問題，提交結果報告                                 | ✓    |
| 技術評估協助        | 1、指定 K8S 產品相關的評估與研析服務<br>2、知識傳遞和最佳實踐分享             | 4次   |
| 最新修補程式發佈報告    | 針對 K8S 軟體產品的修補程式和設置                                | 4次   |
| 多供應商合作        | 通過客戶既有的協作支援協議，TAM 將協助收斂 OCP 問題的範圍，並將問題導向完整且有效的解決方案 | ✓    |
| 到場服務          | 將與客戶的技術團隊安排到場時間，進而了解客戶環境和業務需求的變化。                  | 每週一天 |
| 月度工作報告        | 每月提交匯總一個月的工作內容和出現的問題故障報告                           | ✓    |
| 季度工作報告        | 每季提交匯總一個季度的工作內容和出現的問題故障報告                          | ✓    |
| 問題匯總報告        | 每月提交匯總一個月的工作內容和出現的問題故障報告                           | ✓    |

# BROBRIDGE 專業技術服務

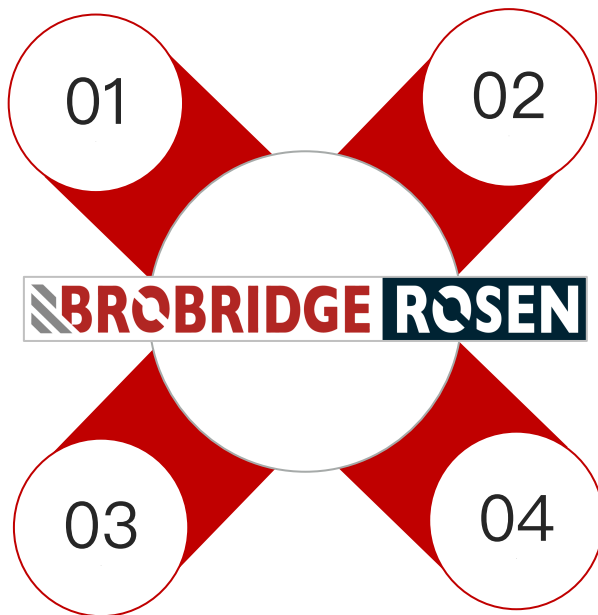
BROBRIDGE PROFESSIONAL SERVICE

## 專業客製化服務

客制化服務團隊訓練與管理機制，並擁有著堅強的研發設計與創新技術能力，是值得信賴的團隊。

## 產品技術服務

提供產品問題服務請求及進度追蹤，並跟蹤用戶需求和故障問題、支援遠程技術支援服務



## 豐富多元化教育訓練課程

提供專業的產品知識和經過驗證的技能，可幫助您簡化部署和改進 K8與微服務平台的運營。

## 技術評估協助機制

產品相關的評估與研析服務，提供知識傳遞和最佳實踐分享，於容器平台的問題解決諮詢，上版的檢視與建議。



#### 關於寬橋有限公司

Brobridge (“寬橋”) 泛指寬橋有限公司 (一家根據台灣法律組成的有限公司)。請參閱 [www.brobridge.com](http://www.brobridge.com) 中有關寬橋有限公司的詳細描述。寬橋有限公司為各行各業之上市及非上市客戶提供資訊技術諮詢、應用程式開發、IT基礎架構設計、顧問服務及系統維運服務。寬橋有限公司網羅業界最專業人士，致力於追求卓越，樹立典範。

#### 關於寬橋有限公司

寬橋 (Brobridge) 係指寬橋有限公司 (Brobridge Co., Ltd)。寬橋有限公司以卓越的客戶服務、優秀的人才、完善的訓練及嚴謹的開發工藝於業界享有良好聲譽。透過寬橋有限公司之資源，提供客戶資訊系統全方位的服務，包括虛擬桌面系統、多雲整合、IT基礎架構規劃設計、先進前沿程式開發等。

本出版物係依一般性資訊編寫而成，僅供讀者參考之用。寬橋有限公司不因本出版物而被視為對任何人提供專業意見或服務。對信賴本出版物而導致損失之任何人，寬橋有限公司之任一個體均不對其損失負任何責任。

