

雲原生架構下的生產力工具

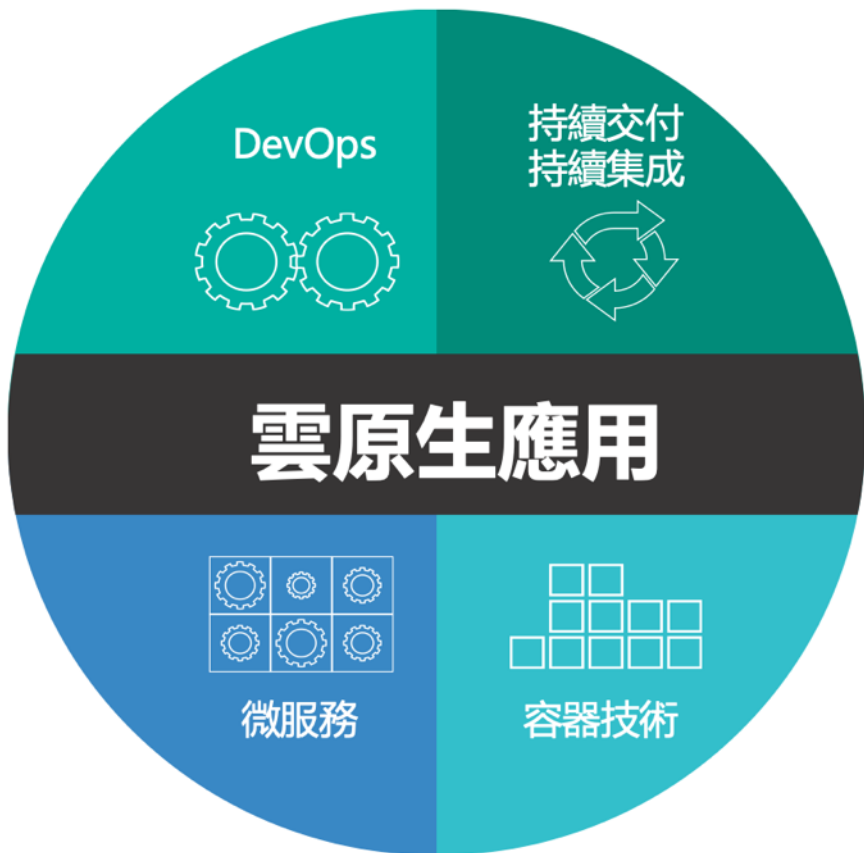
Spring Cloud & Tanzu Application Platform

Rex Wang

MAPBU 資深技術顧問

2022-10-18

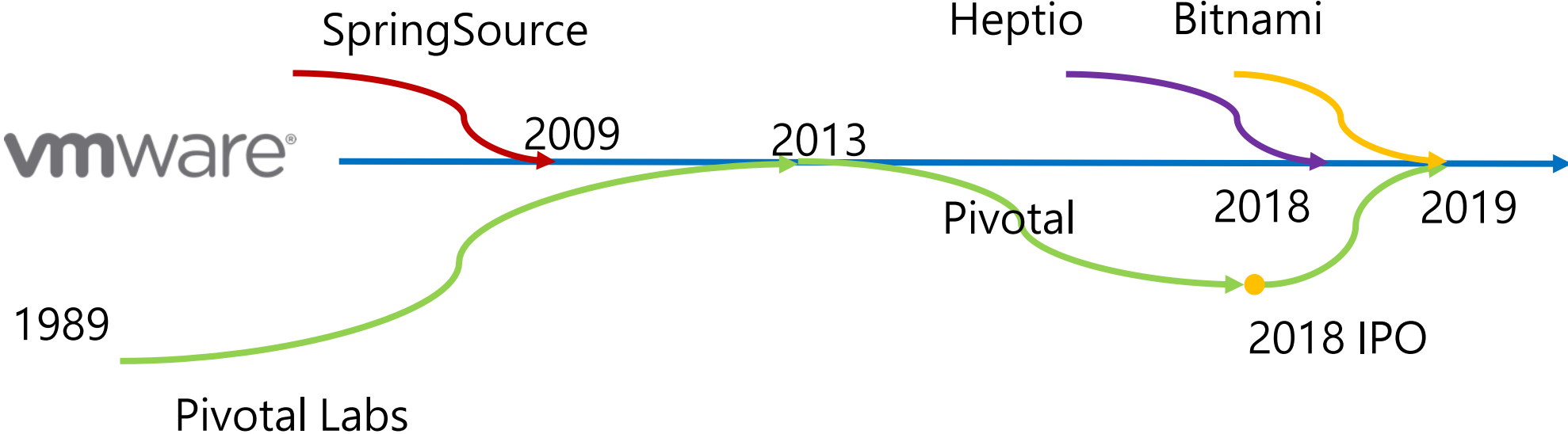
我們率先提出雲原生概念



利用**微服務、容器、DevOps、持續交付技術**

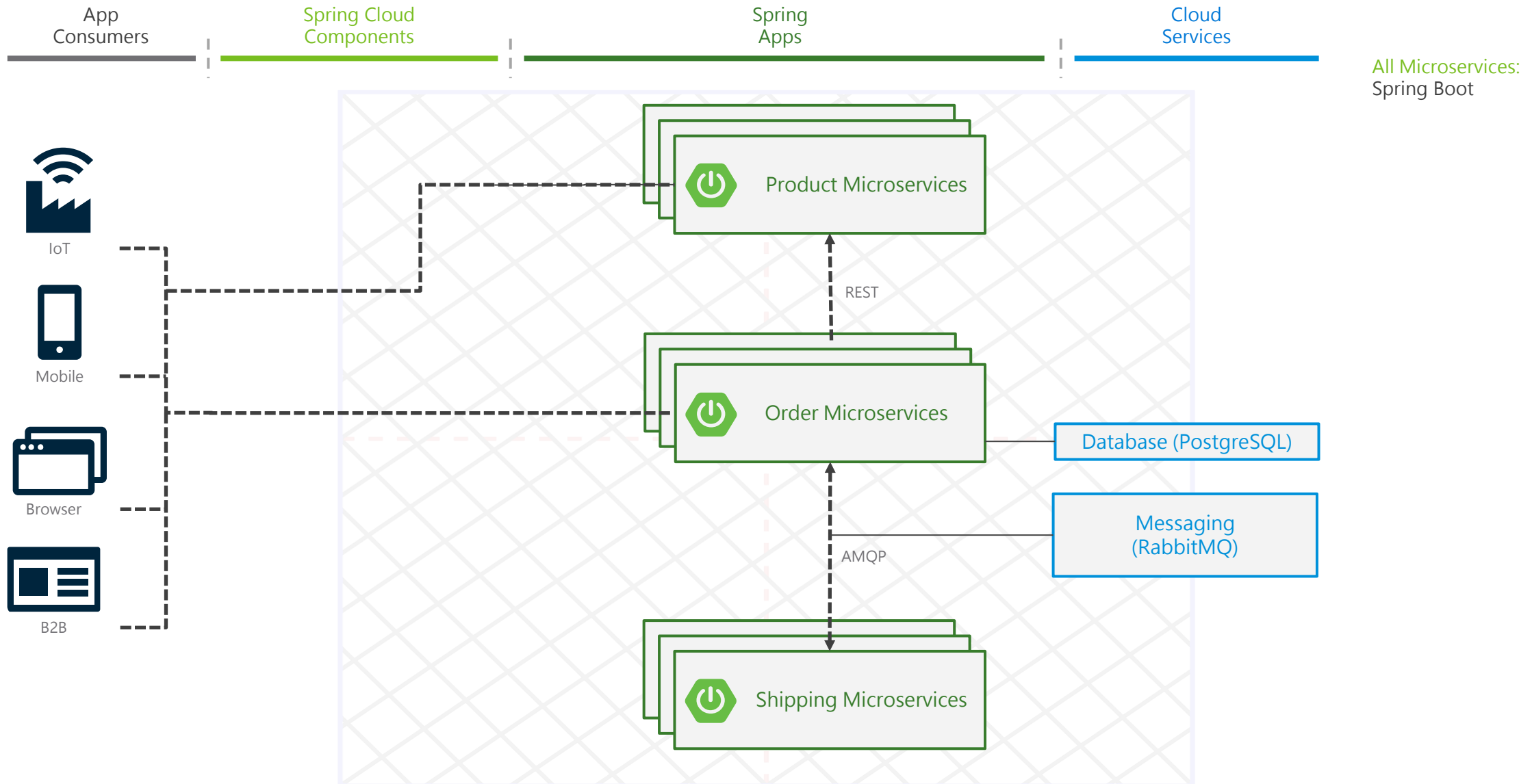
- 響應業務快速變化、快速上線
- 構建微服務應用
- 實現持續交付和部署生產系統

細說從頭VMware與現代化應用開發部署



使用Spring Boot 與 Spring Cloud 建構雲原生應用

典型的現代應用體系結構



Spring by VMware

The Standard for Cloud Native Java



Spring Boot

建構任何內容



Spring Cloud

協調任何內容

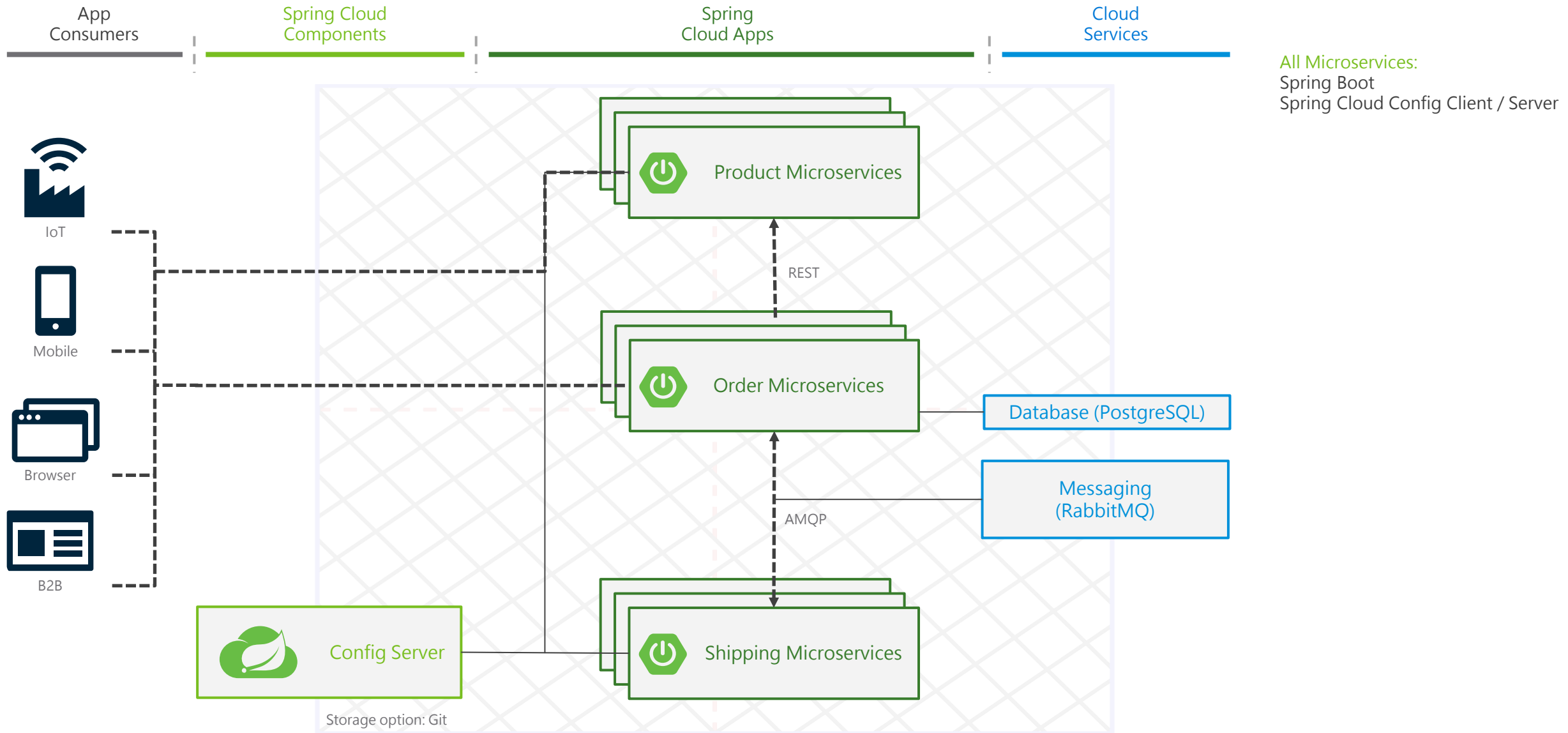


Spring Cloud

資料流
連接任何內容

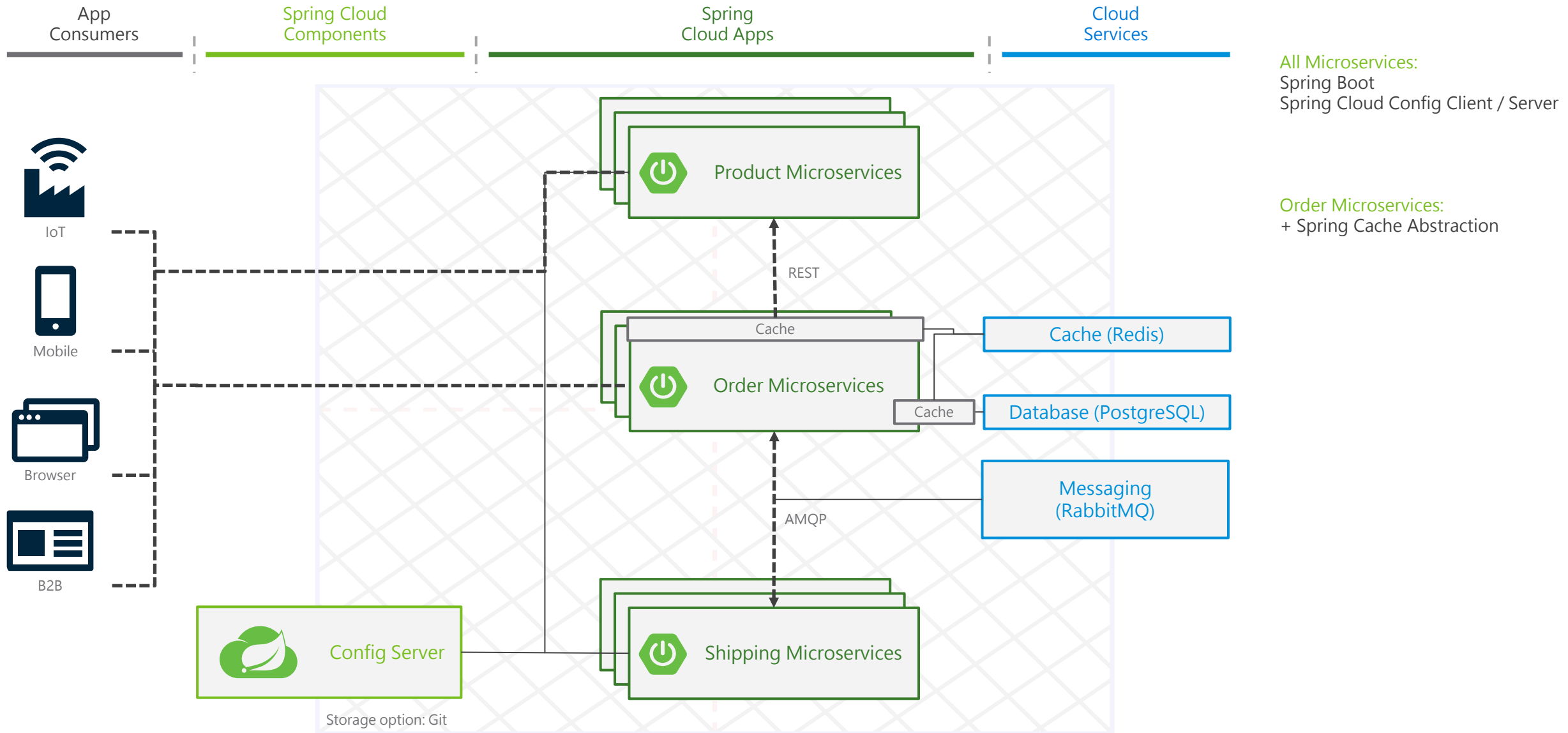
代碼清晰度| 降低複雜性| 減少技術債務|
專注於業務邏輯| 更好的測試覆蓋率| 更快的代碼完成

典型的現代應用體系結構-配置



All Microservices:
Spring Boot
Spring Cloud Config Client / Server

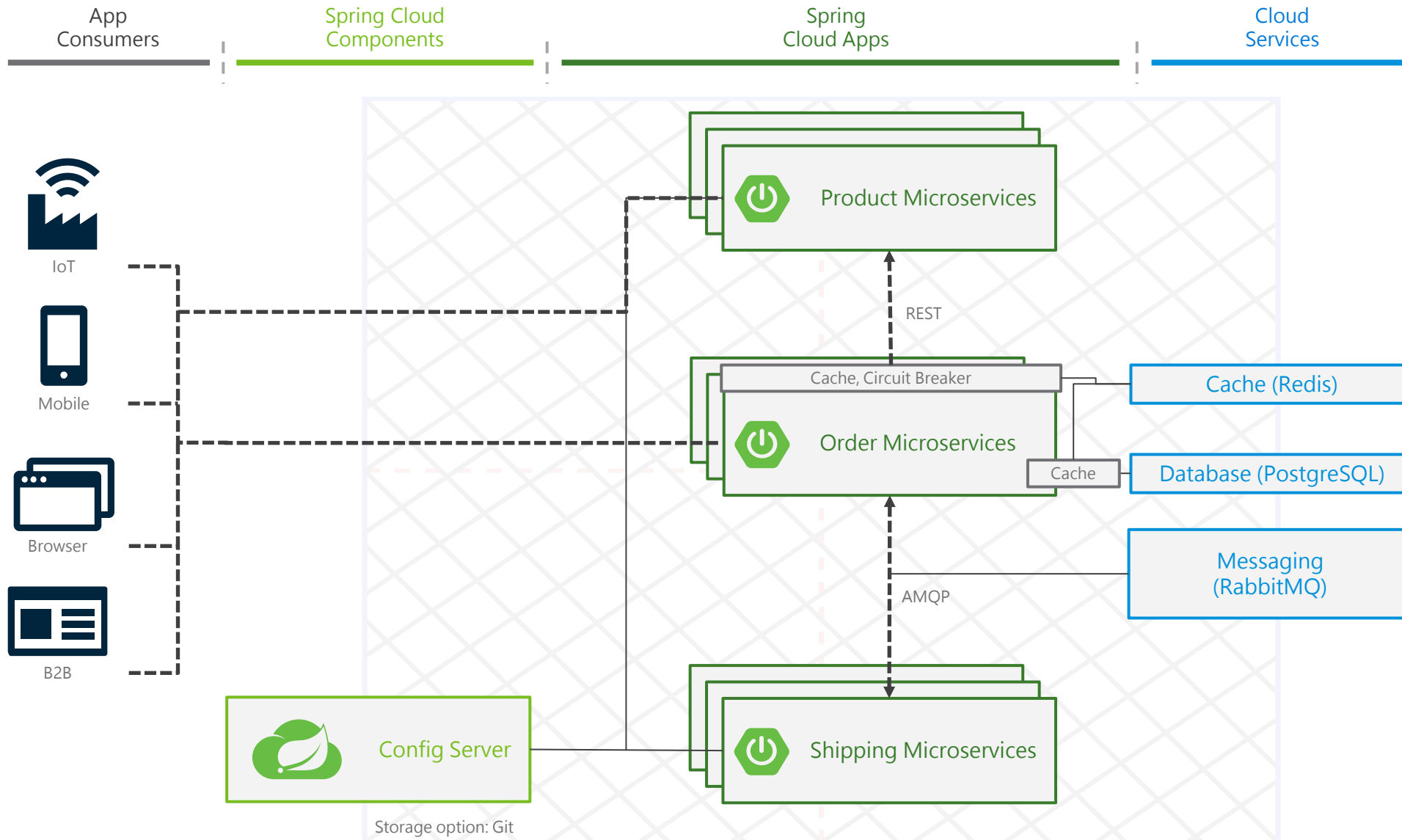
典型的現代應用體系結構-caching



All Microservices:
Spring Boot
Spring Cloud Config Client / Server

Order Microservices:
+ Spring Cache Abstraction

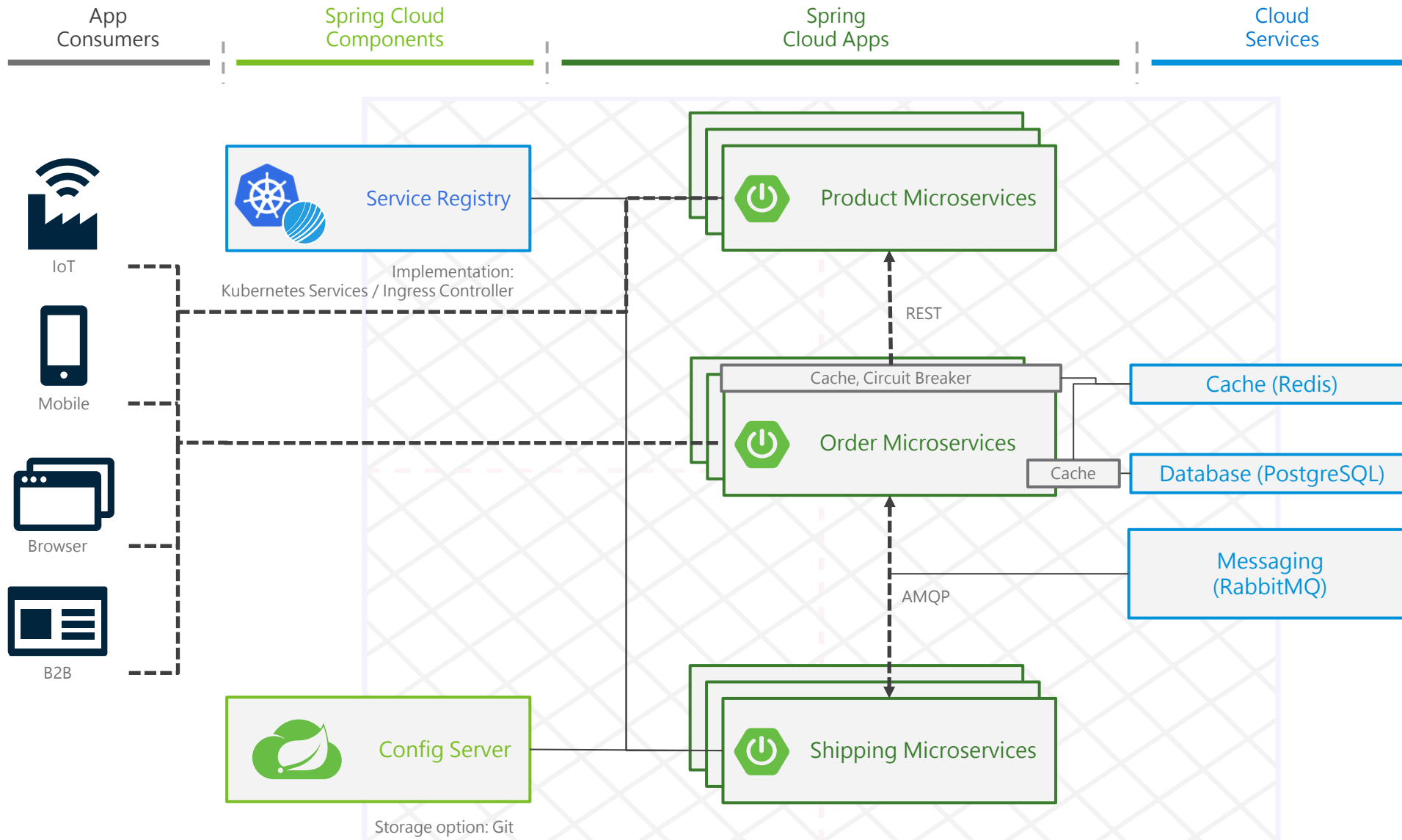
典型的現代應用體系結構-斷路器



All Microservices:
Spring Boot
Spring Cloud Config Client / Server

Order Microservices:
+ Spring Cache Abstraction
+ Spring Cloud Circuit Breaker

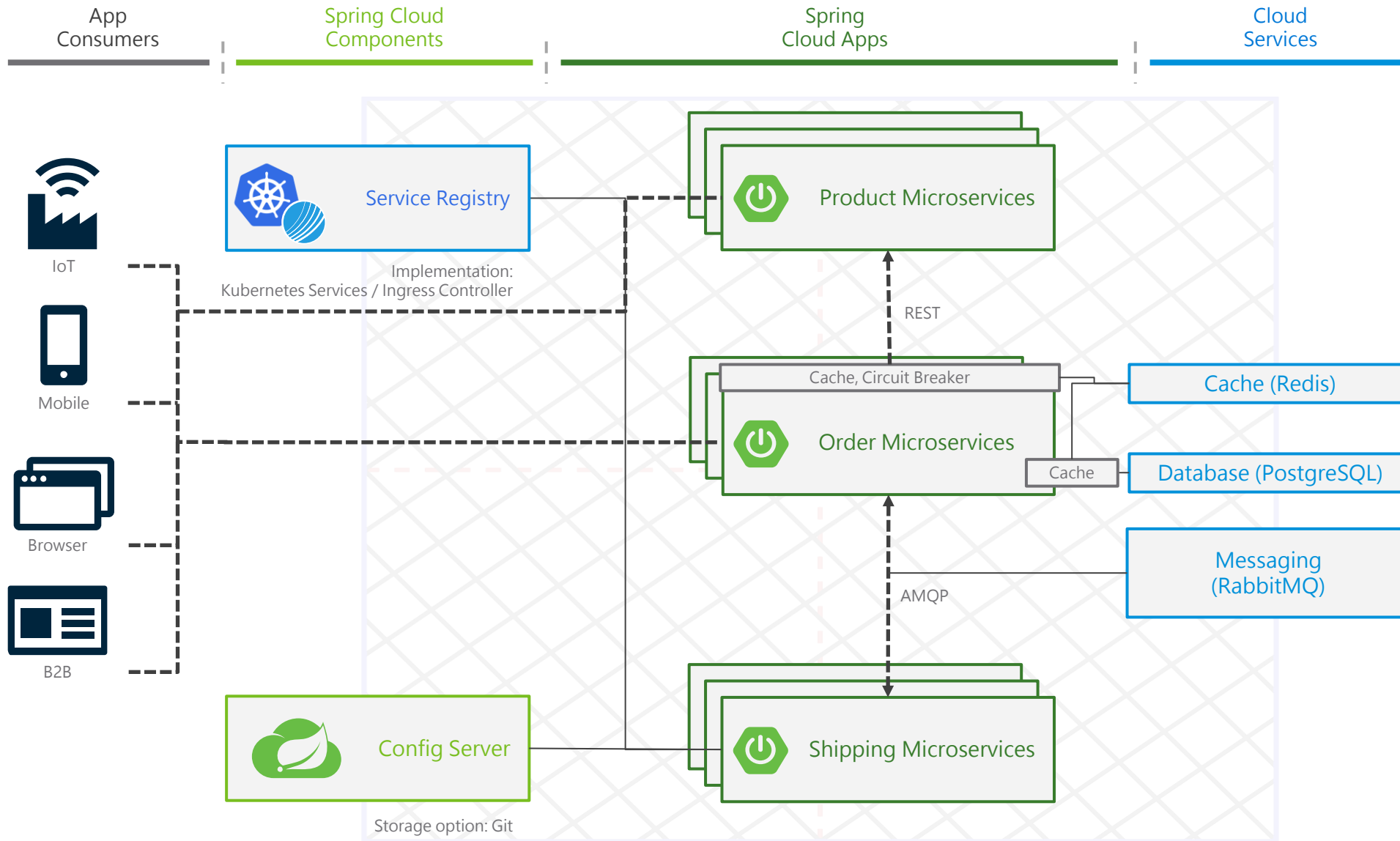
典型的現代應用體系結構-服務註冊發現



All Microservices:
Spring Boot
Spring Cloud Config Client / Server

Order Microservices:
+ Spring Cache Abstraction
+ Spring Cloud Circuit Breaker

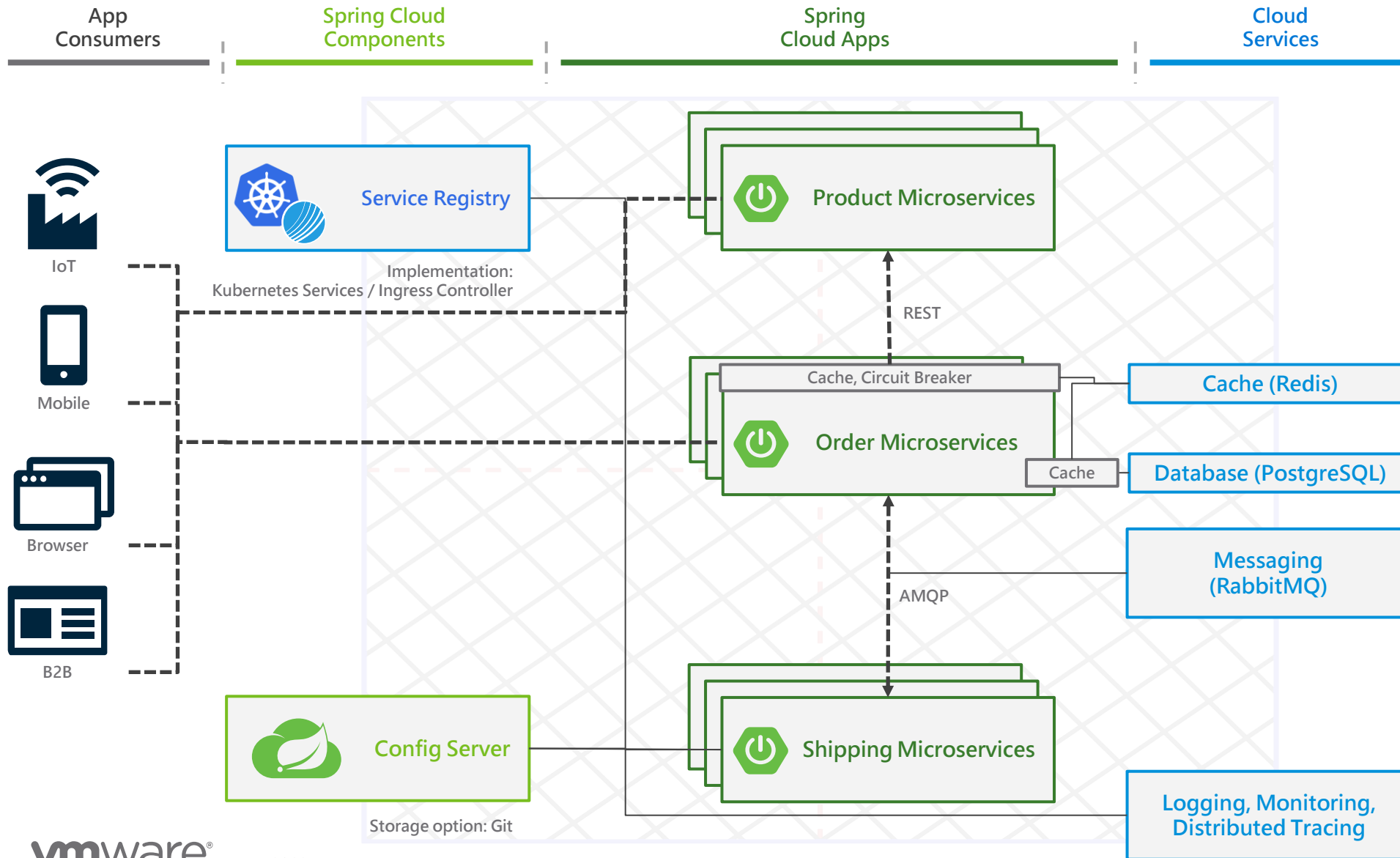
典型的現代應用體系結構-服務註冊發現



All Microservices:
Spring Boot
Spring Native
Spring Cloud Config Client / Server

Order Microservices:
+ Spring Cache Abstraction
+ Spring Cloud Circuit Breaker

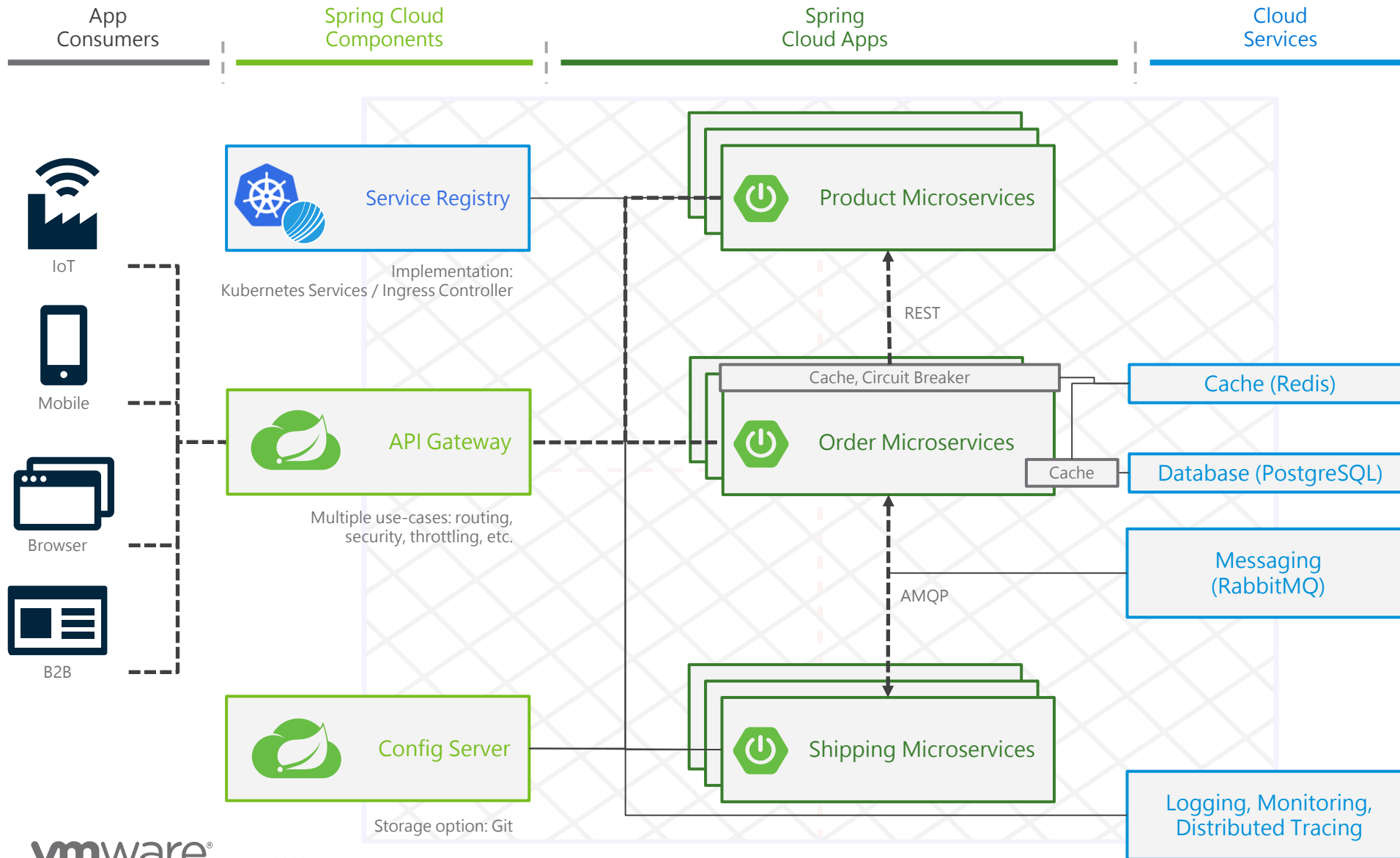
典型的現代應用體系結構-分散式追蹤



All Microservices:
 Spring Boot
 Spring Native
 Spring Cloud Config Client / Server
 Spring Cloud Sleuth

Order Microservices:
 + Spring Cache Abstraction
 + Spring Cloud Circuit Breaker

典型的現代應用體系結構-Gateway

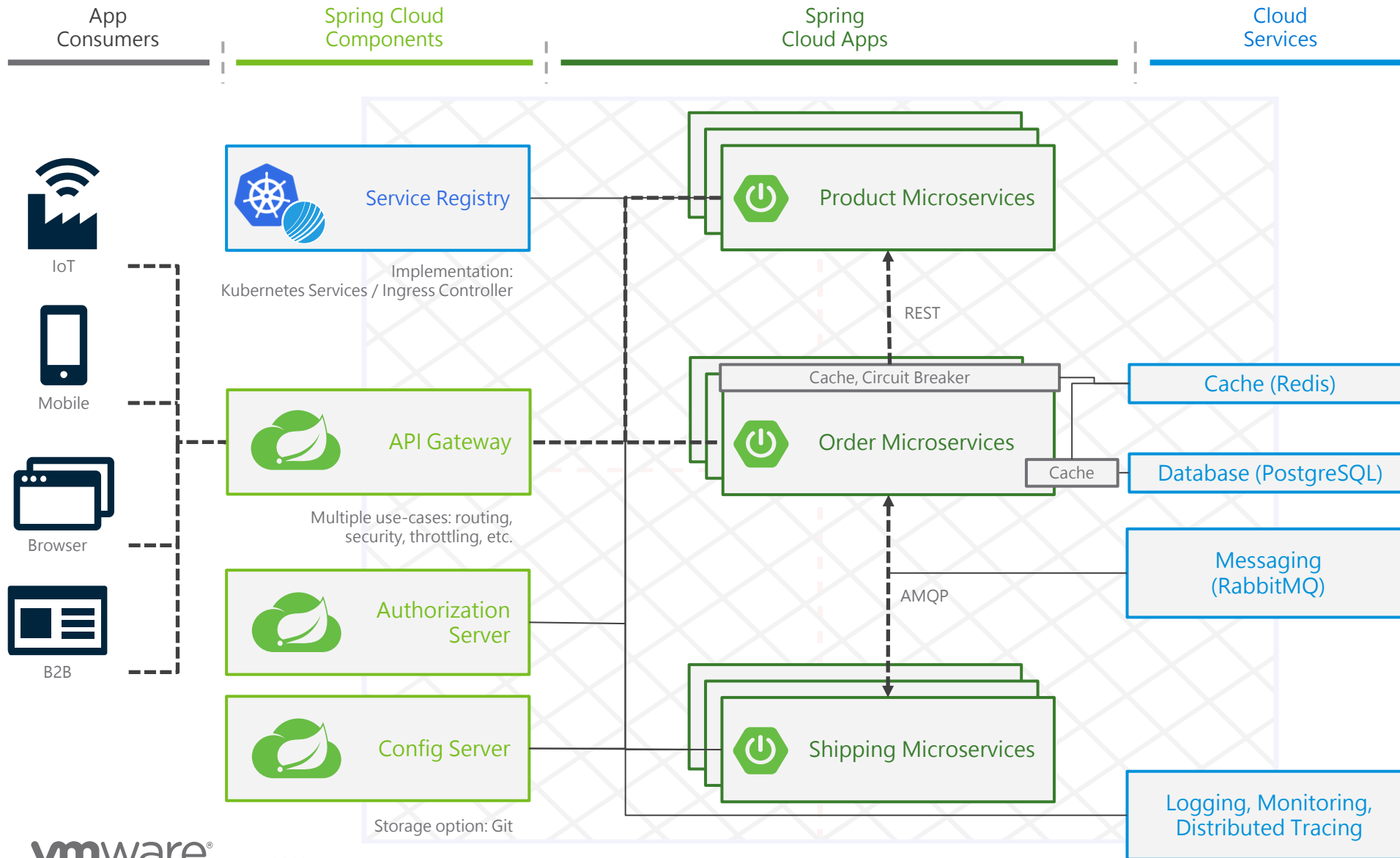


All Microservices:
 Spring Boot
 Spring Native
 Spring Cloud Config Client / Server
 Spring Cloud Sleuth

Order Microservices:
 + Spring Cache Abstraction
 + Spring Cloud Circuit Breaker

Gateway
 + Spring Cloud Gateway

典型的現代應用體系結構-Security

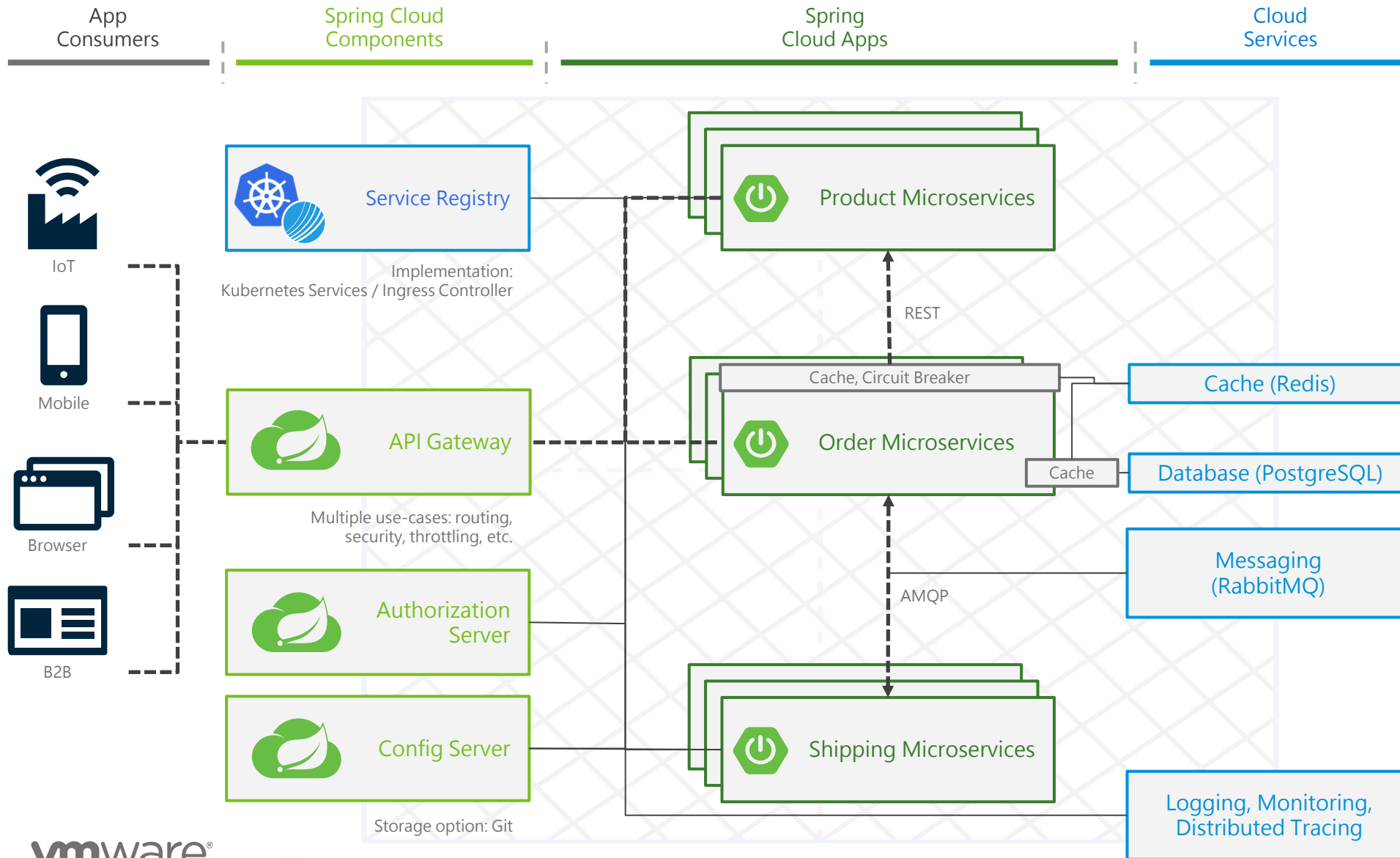


All Microservices:
 Spring Boot
 Spring Native
 Spring Security (OAuth 2 Client)
 Spring Cloud Config Client / Server
 Spring Cloud Sleuth

Order Microservices:
 + Spring Cache Abstraction
 + Spring Cloud Circuit Breaker

Gateway
 + Spring Cloud Gateway

典型的現代應用體系結構-資料流



All Microservices:

- Spring Boot
- Spring Native
- Spring Security (OAuth 2 Client)
- Spring Cloud Config Client / Server
- Spring Cloud Sleuth

Order Microservices:

- + Spring Cache Abstraction
- + Spring Cloud Circuit Breaker

Shipping Microservices:

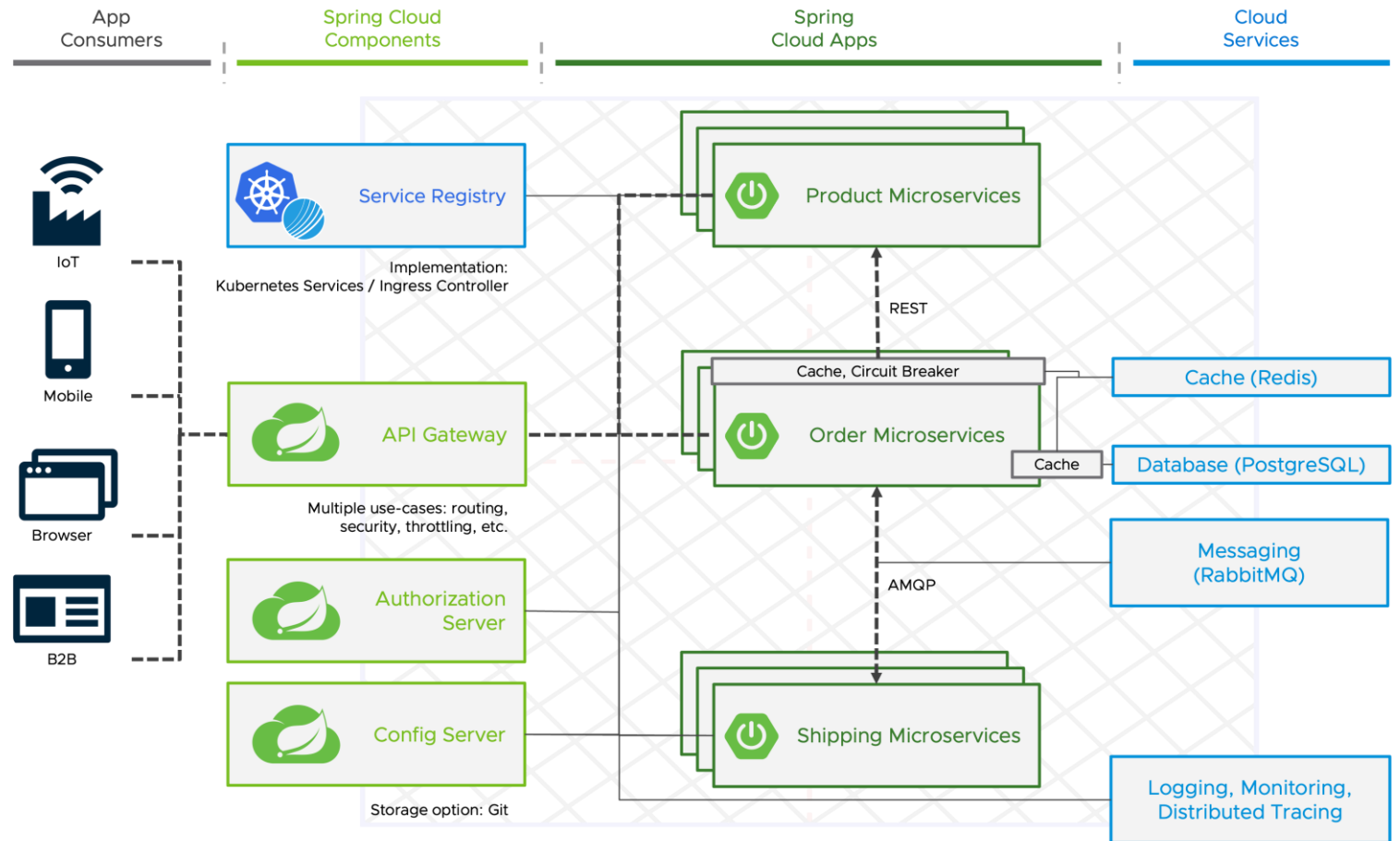
- + Spring Cloud Stream
- + Spring Cloud Function

Gateway

- + Spring Cloud Gateway

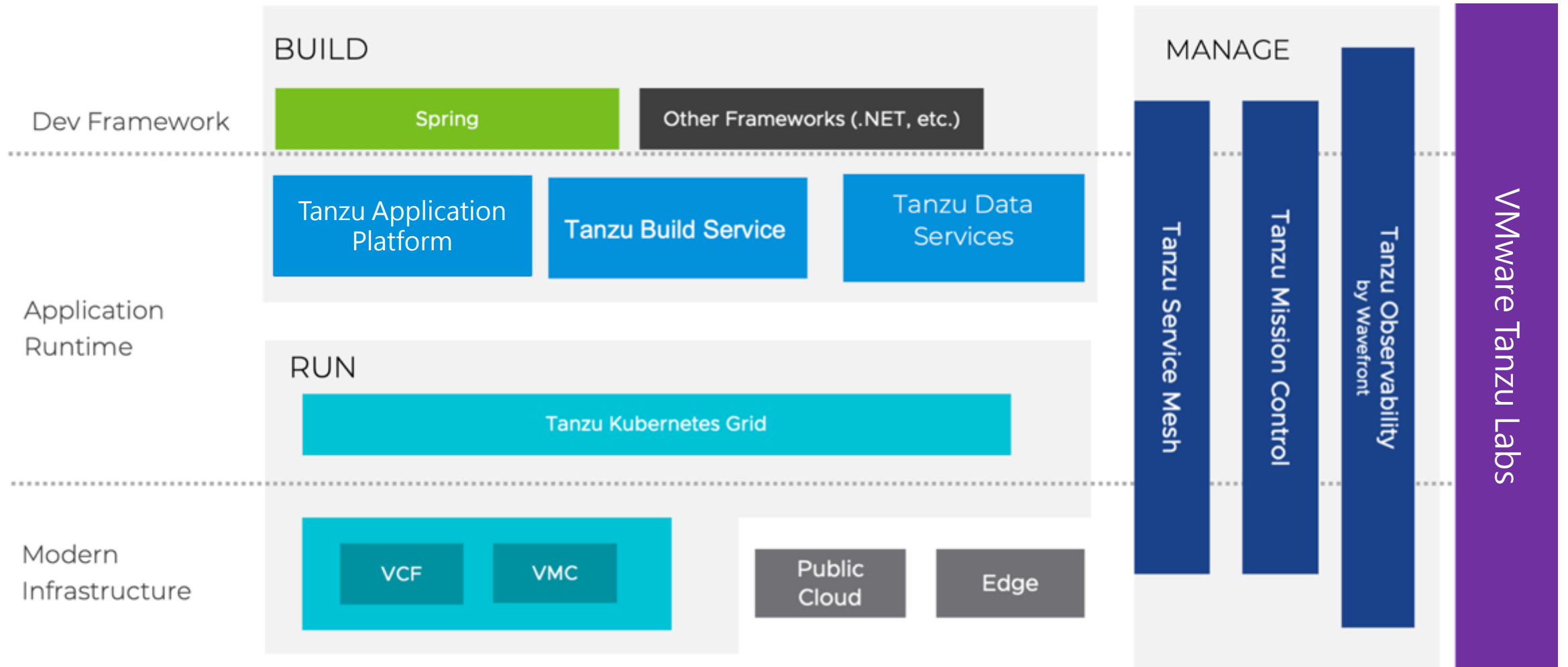
常見的困難與挑戰

- 管理微服務的雲基礎架構需要付出巨大努力
- 應用程式生命周期難以管理
- 解決應用程式問題的痛苦

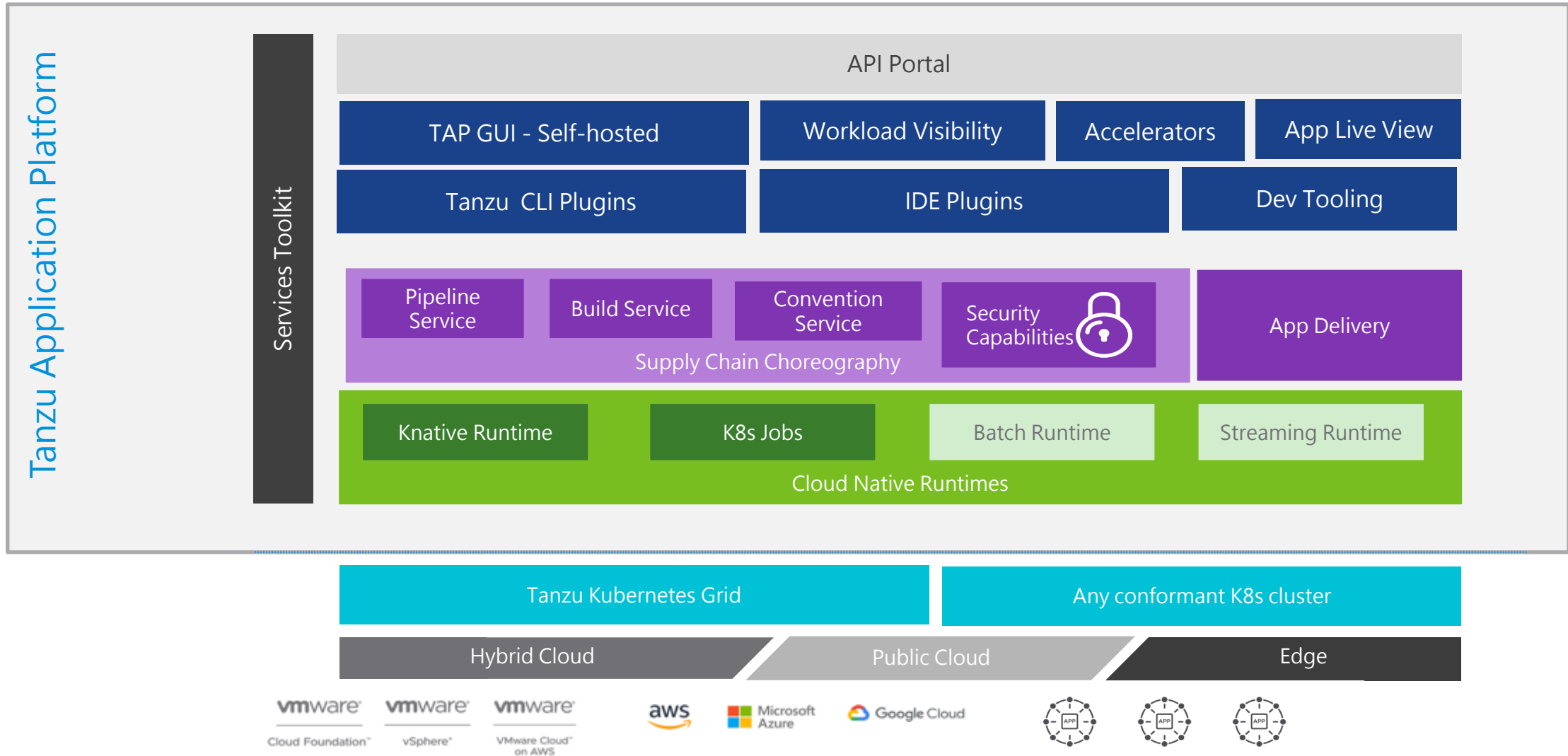


打造新世代 PaaS

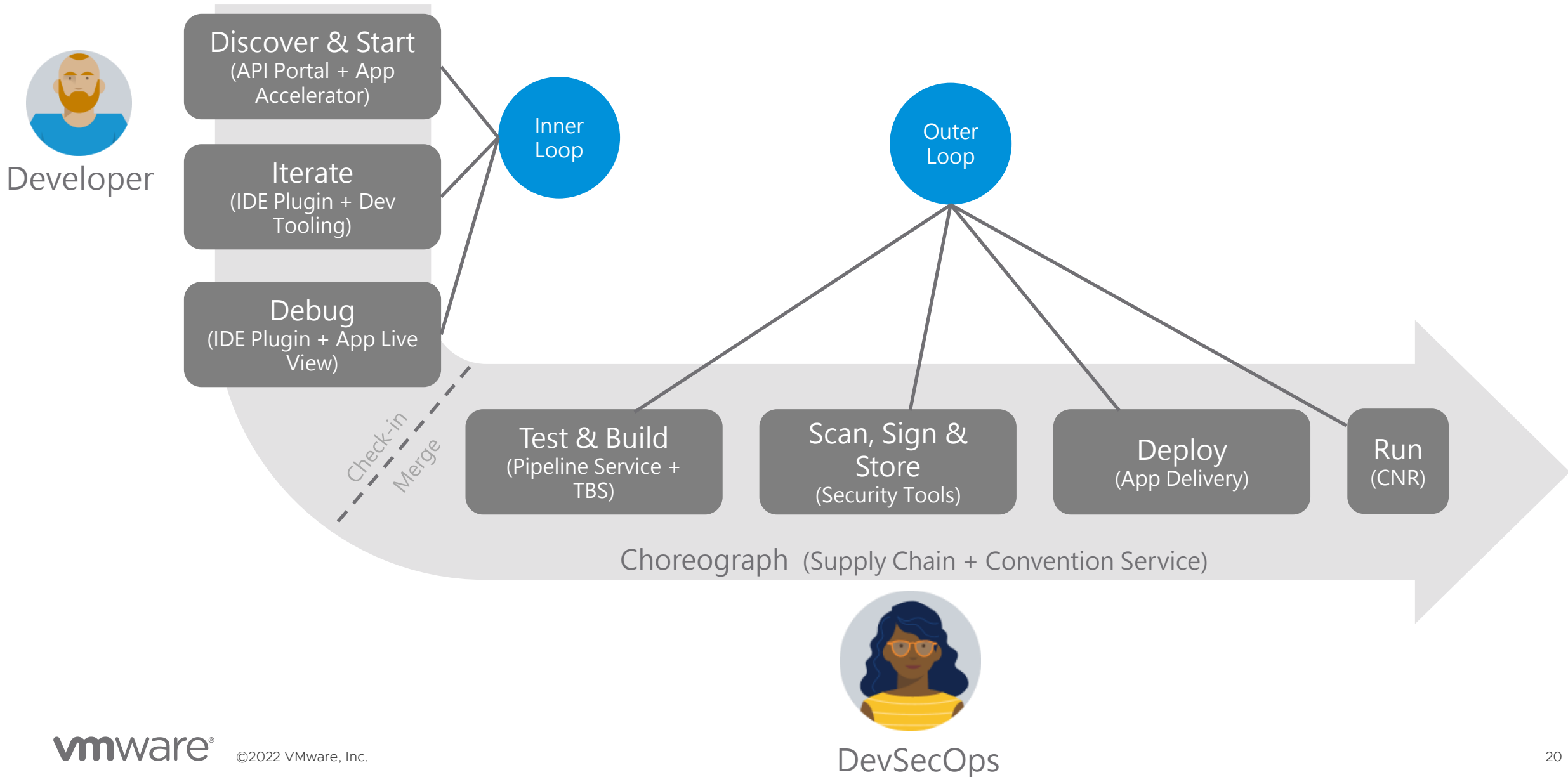
現代化應用所需的各項元件與服務



下一代PaaS : Tanzu Application Platform

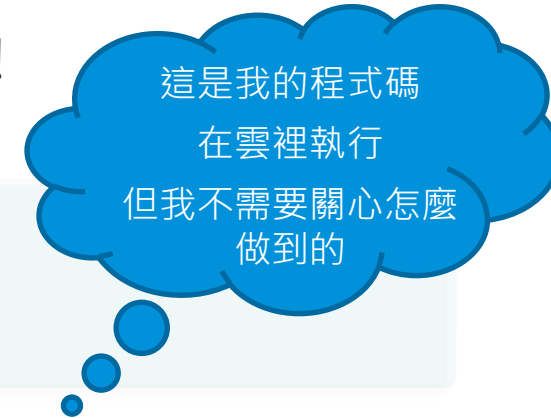


從開發到維運，全方位的支援：Tanzu Application Platform



業務應用的開發和維運在雲原生背景下所面臨的挑戰

開發者效率直接關係企業的業務上線速度和迭代，關乎企業增長、盈利和創新！



DEV

開發人員花了太多時間在YAML編寫和管理 - YAML泥沼

我有多個 Pipeline 需要管理

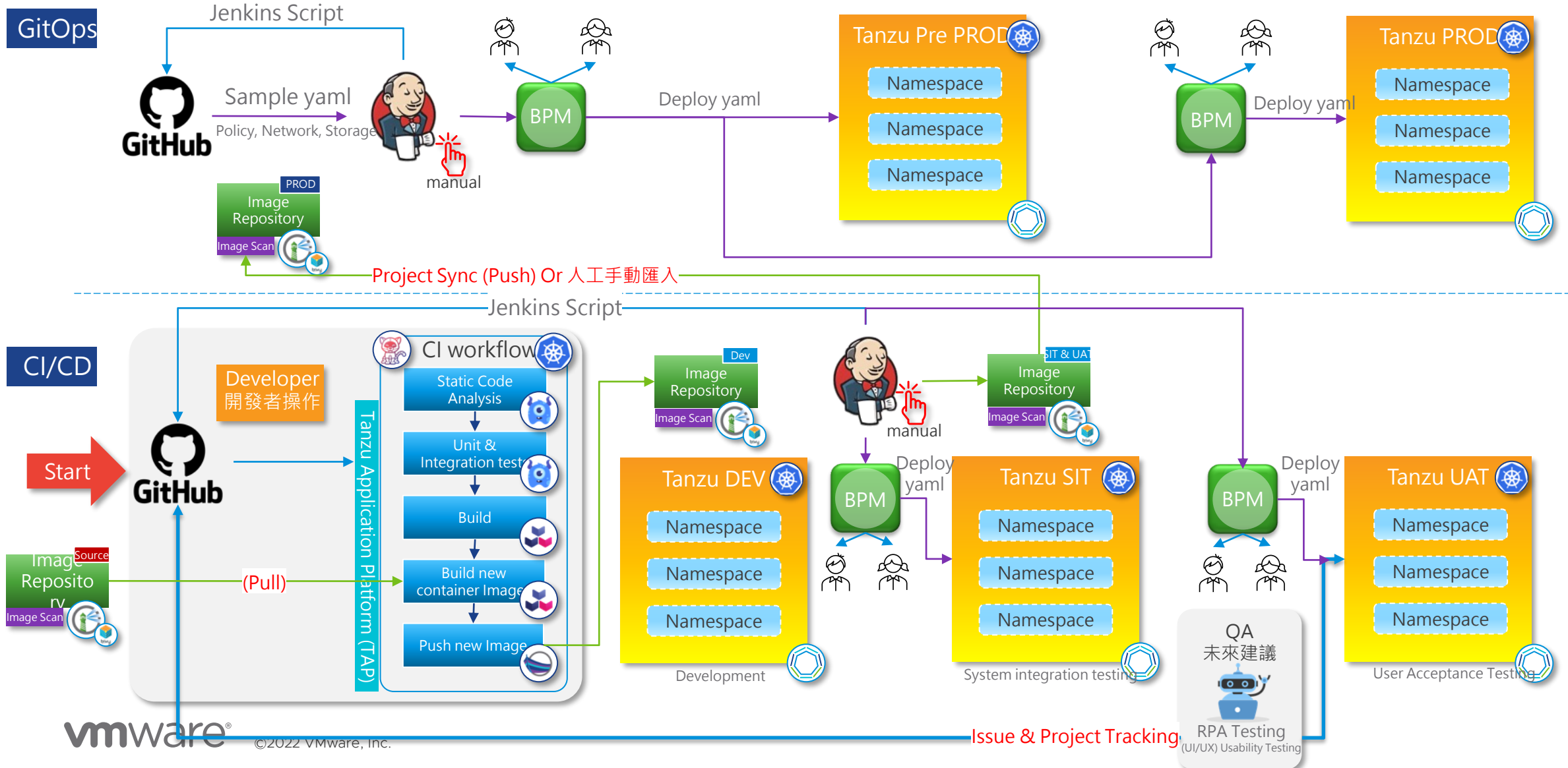
我需要整合現有的DevSecOps & Security Practices

我們的應用程式在多雲、混合雲環境中運行 – 跨雲，跨 K8s



OPS

DevSecOps 常見規劃



CNCF 的安全軟體供應鏈



CLOUD NATIVE COMPUTING FOUNDATION About Projects Training Community Blog & News

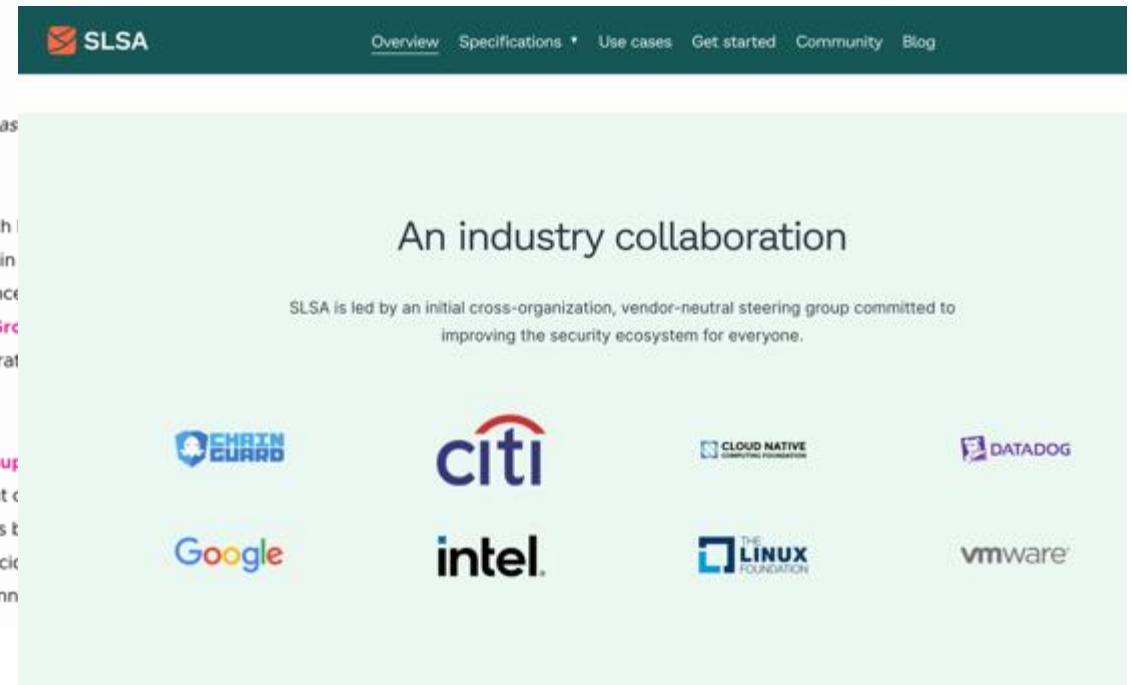
CNCF paper defines best practices for supply chain security

New paper demonstrates an actionable approach to architecting a secure supply chain amidst an increase in attacks

SAN FRANCISCO, Calif. – May 14, 2021 – The Cloud Native Computing Foundation* (CNCF*), which sustains sustainable ecosystems for cloud native software, today announced a new paper, Software Supply Chain Practices, designed to provide a holistic approach to supply chain security by highlighting the importance of defensive practices. The paper was compiled by members of the **CNCF Security Technical Advisory Group**, which produces resources that enable secure access, policy control, and safety for operators, administrators, developers, and end users across the cloud native ecosystem.

"The Security TAG has been focused on supply chain security for a few years, first with the **catalog of supply chain compromises** and now the paper," said Emily Fox, co-chair of the CNCF Security TAG. "It is critical that cloud native and open source communities seriously consider not only what their software does but the mechanisms that it comes to be. As security practitioners, we recognize the difficulty in rapidly pivoting from incident to incident. It is the time to thoughtfully consider a better, more secure end-to-end architecture responsible for our innovation."


<https://slsa.dev/>



SLSA Overview Specifications Use cases Get started Community Blog

An industry collaboration

SLSA is led by an initial cross-organization, vendor-neutral steering group committed to improving the security ecosystem for everyone.



場景1



我們這個專案好像和其它專案挺類似的，我不想每次都從零開始，可不可以站在前人的肩膀上，“參考”其他類似專案，以它們為模版和基礎，進行後續開發呢？

我們已經整理了典型的成功專案，作為 **template**，並且發佈出來作為 **應用加速器**，這樣也便於快速推廣和複製，你們的專案做得好，也別忘了再貢獻回來給別人參考啊

架構師



App Accelerator

Accelerators

All accelerators

Search for accelerators...

TAGS 18

- accelerator
- c#
- cloud
- dotnet
- express
- f#
- function
- java
- jpa
- node
- serverless
- spring
- sql
- steeltoe
- tanzu
- web
- yaml
- ytt

hello-fun



Hello Fun

java spring cloud function serverless

A simple Spring Cloud Function serverless app -- includes option for native build

REPOSITORY

hello-ytt



Hello ytt

yaml ytt

Demonstrates how to use the YTT transform to generate YAML

REPOSITORY

new-accelerator



New Accelerator

tanzu accelerator

Generate yaml for a new Accelerator resource.

REPOSITORY

node-express



Node Express

node express

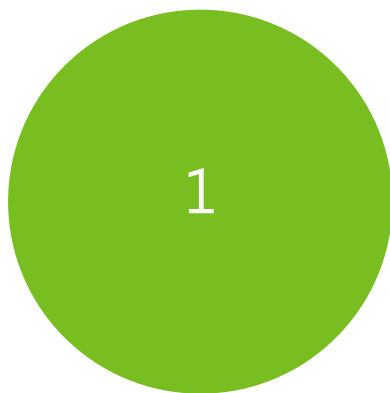
A Node.js sample app using Express

REPOSITORY

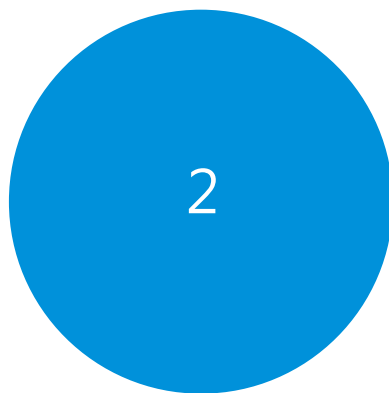
企業按照自己的技術標準定製/擴展，讓架構規範落地

架構師
準備/定義

程式碼與設定 = Accelerators

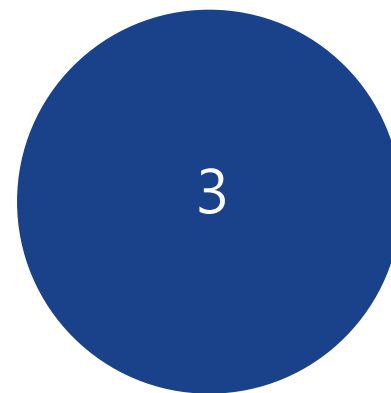


開發人員
選擇Accelerator

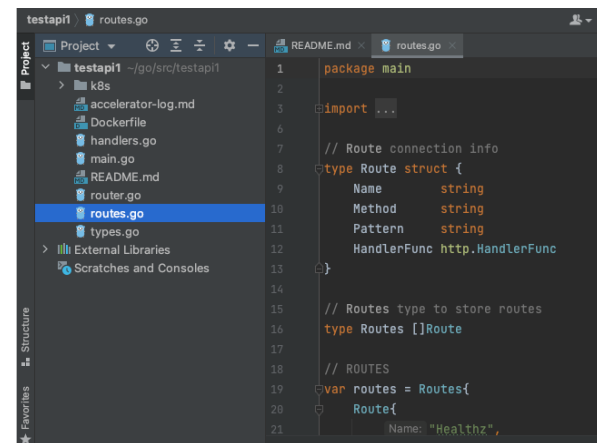
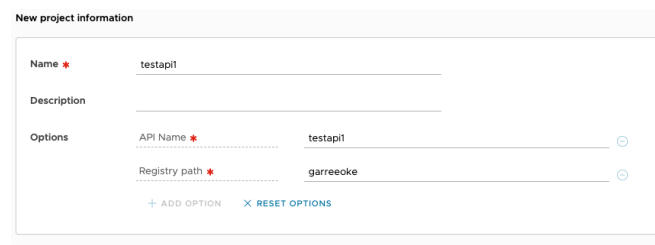
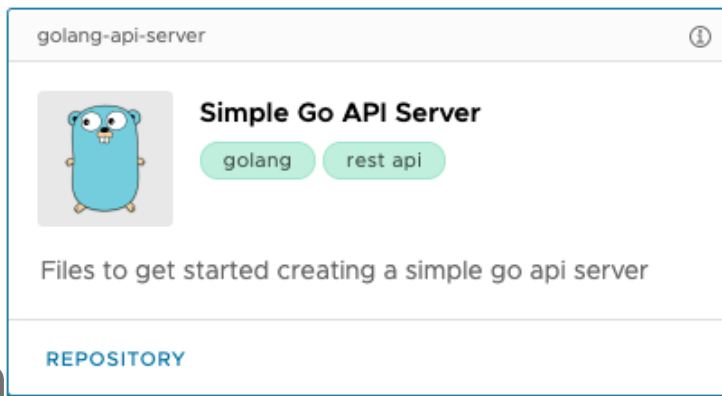


提供必要的設定與參數

.....
文件處理



下載壓縮檔 ZIP



場景2



開發

應用打包成鏡像也要我來寫？好吧，那寫Dockerfile要注意哪些地方？聽說有不少要留意的坑.....，還有用到的基礎鏡像如果有更新了，誰來負責更新到我們環境啊？

不用寫Dockerfile了，直接部署程式碼和應用，我們的**鏡像建構服務**會自動生成鏡像，無論是你的程式碼有更新，還是基礎鏡像有更新，都會自動觸發構建新鏡像，並保存到鏡像倉庫

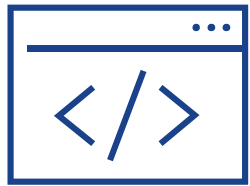
架構師



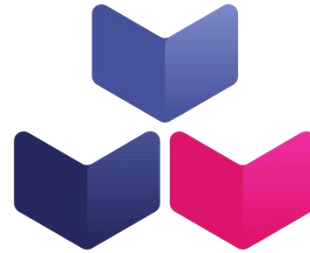
Cloud Native Buildpacks

code to images

Source Code



Cloud Native Buildpacks



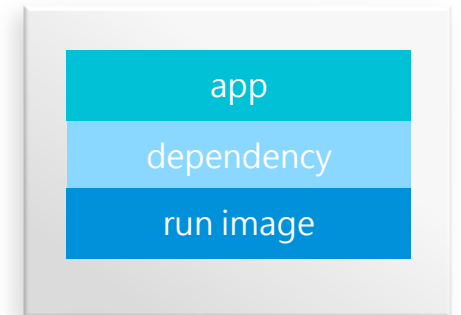
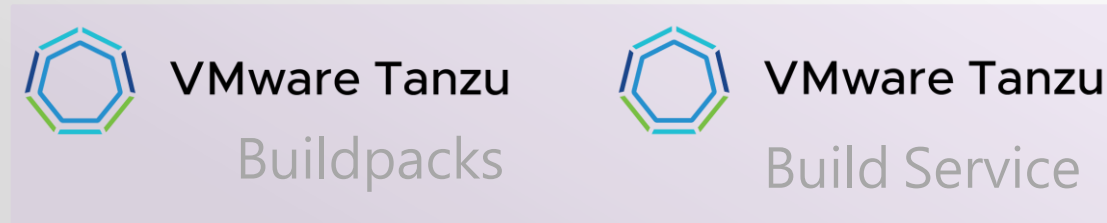
OCI Compliant Containers



Open Source Buildpacks + Build Platform



Enterprise Ready Buildpacks + Build Platform



Kubernetes



Public Clouds

場景3

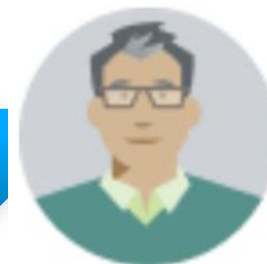


開發

我猜那麼多 K8s yaml 文件大概也要我來寫吧..... K8s還是太複雜了，我會用，但學藝不精，我怕用我寫的上生產環境會有問題.....

不用自己寫那麼多K8s yaml啦！只要寫一個**workload yaml**就行了，這個是更簡單的應用抽象，只要聲明式的配置應用，最關鍵是指定應用所屬的類型

架構師



應用抽象：Workload

Workload manifest：

- 名稱
- 類型
- 標籤與註釋
- 程式碼位置
- 相依的服務
- 環境變數
- 參數
- 資源需求

```
workload.yaml
! workload.yaml x
Users > bhale > Desktop > ! workload.yaml > ...
1  apiVersion: apps.tanzu.vmware.com/v1alpha1
2  kind: Workload
3  metadata:
4    name: sample-application
5    labels:
6      apps.tanzu.vmware.com/workload-type: web
7  spec:
8    source:
9      git:
10     url: https://gitlab.eng.vmware.com/bhale/sample-application.git
11   serviceClaims:
12     - name: database
13     ref:
14       apiVersion: services.tanzu.vmware.com/v1alpha1
15       kind: PostgreSQL
16       name: my-prod-db
17   env:
18     - name: SPRING_PROFILES_ACTIVE
19     value: postgresql
20   resources:
21     requests:
22       memory: 1Gi
23       cpu: "0.1"
24     limits:
25       memory: 4Gi
26       cpu: "4"
27
```

場景4



開發

Spring應用的常用actuator指標比如health, memory, metrics等，怎麼才能看到？

平台已經提供了Spring Boot Convention, 自動就開啟了Actuator, 通過App Live View 界面就能看到那些可觀測性指標

架構師



Workload執行畫面

COMPONENT — SERVICE

my-java-web-app ☆

Owner: default-team Lifecycle: experimental

OVERVIEW DEPENDENCIES **RUNTIME RESOURCES**

myapp-00001-deployment-85c7c85879-tvhvg ✓ Ready

v1/Pod REFRESH Last refreshed: a few seconds ago

AGE: 4 days ago

CLUSTER: host

NAMESPACE: default

[.YAML](#)

Status Conditions

Condition Type	Status	Last Transition
Ready	✓ True	4 days
ContainersReady	✓ True	4 days
Initialized	✓ True	4 days

3 rows |< < 1-3 of 4 > >|

Ownership

- my-java-web-app (Component)
 - ✓ myapp (serving.knative.dev/v1/Service)
 - ✓ myapp (serving.knative.dev/v1/Configuration)
 - ✓ myapp-00001 (serving.knative.dev/v1/Revision)
 - ✓ myapp-00001-deployment (apps/v1/Deployment)
 - ✓ myapp-00001-deployment-85c7c85879 (apps/v1/ReplicaSet)
 - ✓ myapp-00001-deployment-85c7c85879-tvhvg (v1/Pod)

場景5



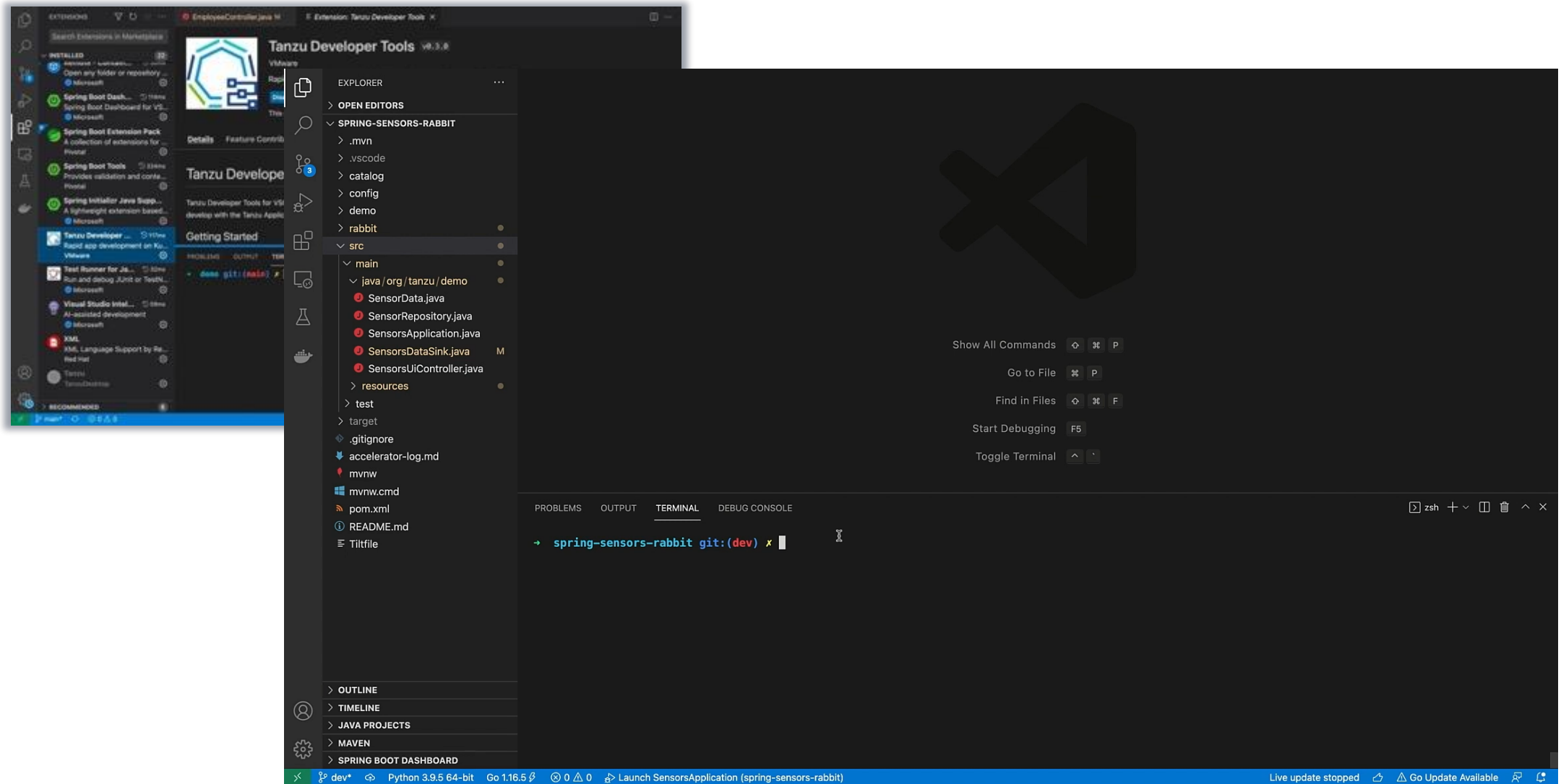
公司電腦規格好不好？這些機器資源本地端開發夠不夠？速度快不快？有沒有遠端的個人開發環境？

平台提供遠端的K8s環境，每個開發都有一個自己的namespace，可以部署自己的程式碼和服務，本地端可以連到這個開發環境，透過**IDE plugin**和**Tilt**，實現遠程更新和調校

架構師

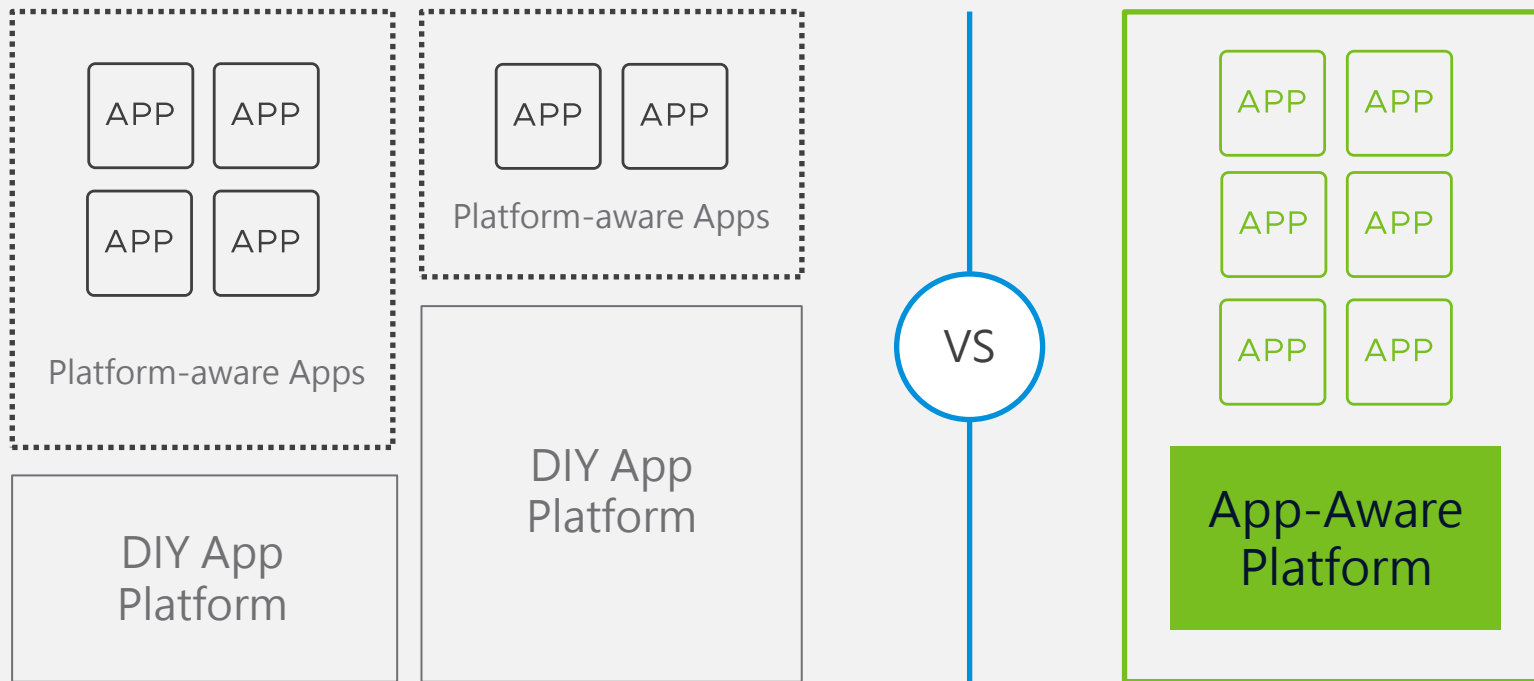


本機上IDE整合



應用感知平台的優勢

Removing the burden from the developers



開發人員專注於定義應用的要求

啟用左移模式，不會給開發人員帶來負擔

從開發到生產的完全便攜性

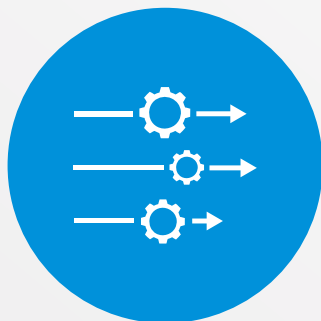
自動應用平臺最佳實踐

Tanzu Application Platform

在Kubernetes上，
提供卓越的多雲開發人員體驗



提高開發人員的工作效率



建立快速而可持續的生產途徑



協調開發、安全、維運的工作

Thanks

ModernApps Learning



<https://modernapps.ninja/>

Hands-On Workshops



[https://tanzu.vmware.com/
developer/workshops/](https://tanzu.vmware.com/developer/workshops/)