



# Sistemas Distribuidos I (75.74)

## Arquitecturas Orientadas a Servicios y *Cloud*

Servicios. Cloud Computing. Platform as a Service (PaaS).  
AppEngine

### Docentes

- Pablo D. Roca
- Ezequiel Torres Feyuk
- Guido Albarello

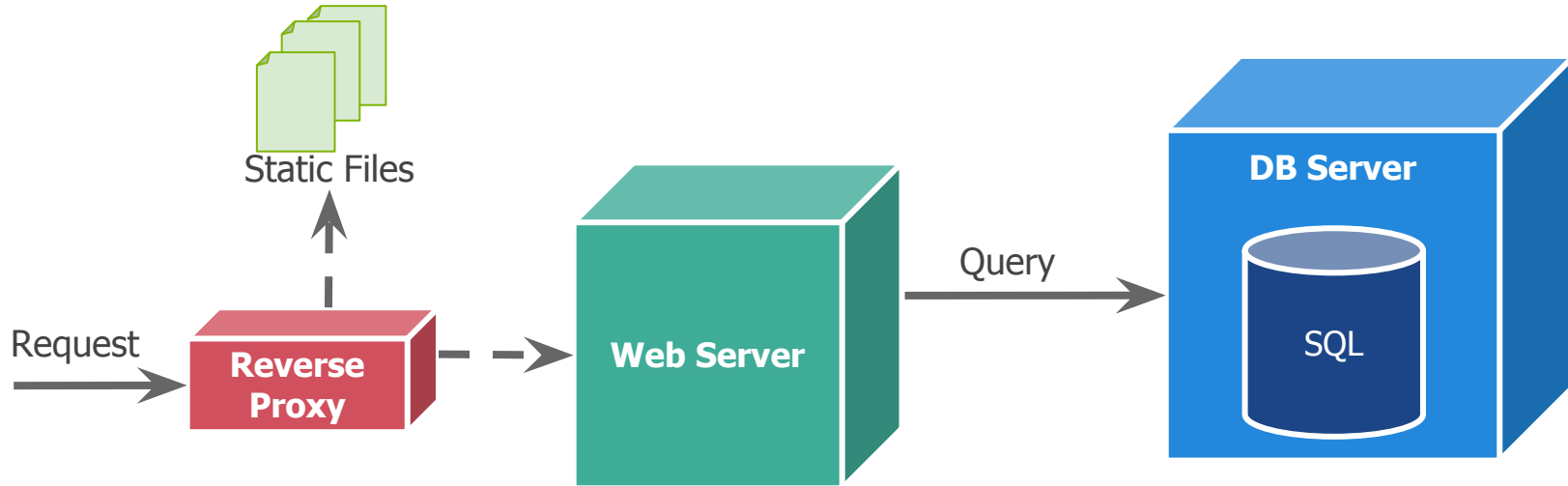
- Ana Czarnitzki
- Cristian Raña



## ● Arquitecturas Orientadas a Servicio

- Cloud
- Platform as a Service (PaaS)
- Caso de estudio: Google AppEngine y Datastore

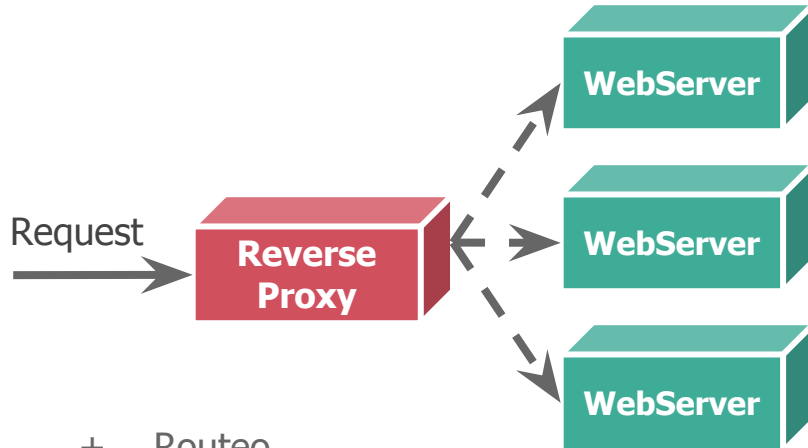
# Evolución en Arquitecturas | Monolíticas





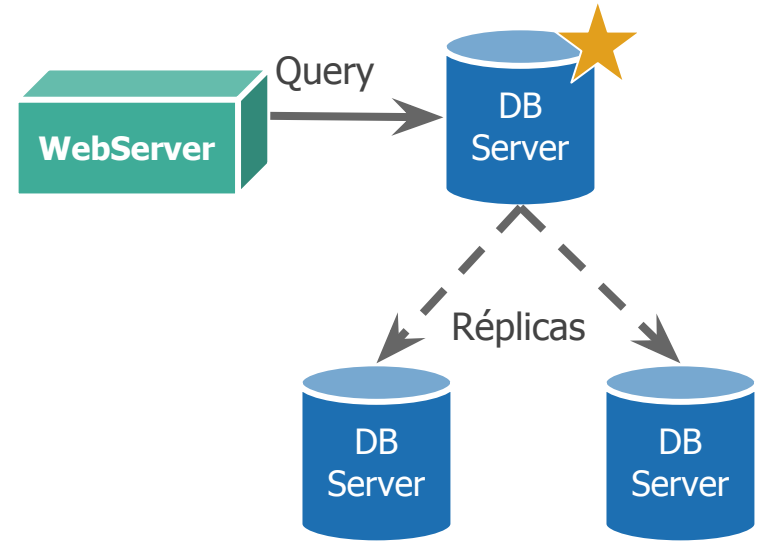
# Evolución en Arquitecturas | Monolíticas (Escalables)

Vista de *Web Requests*



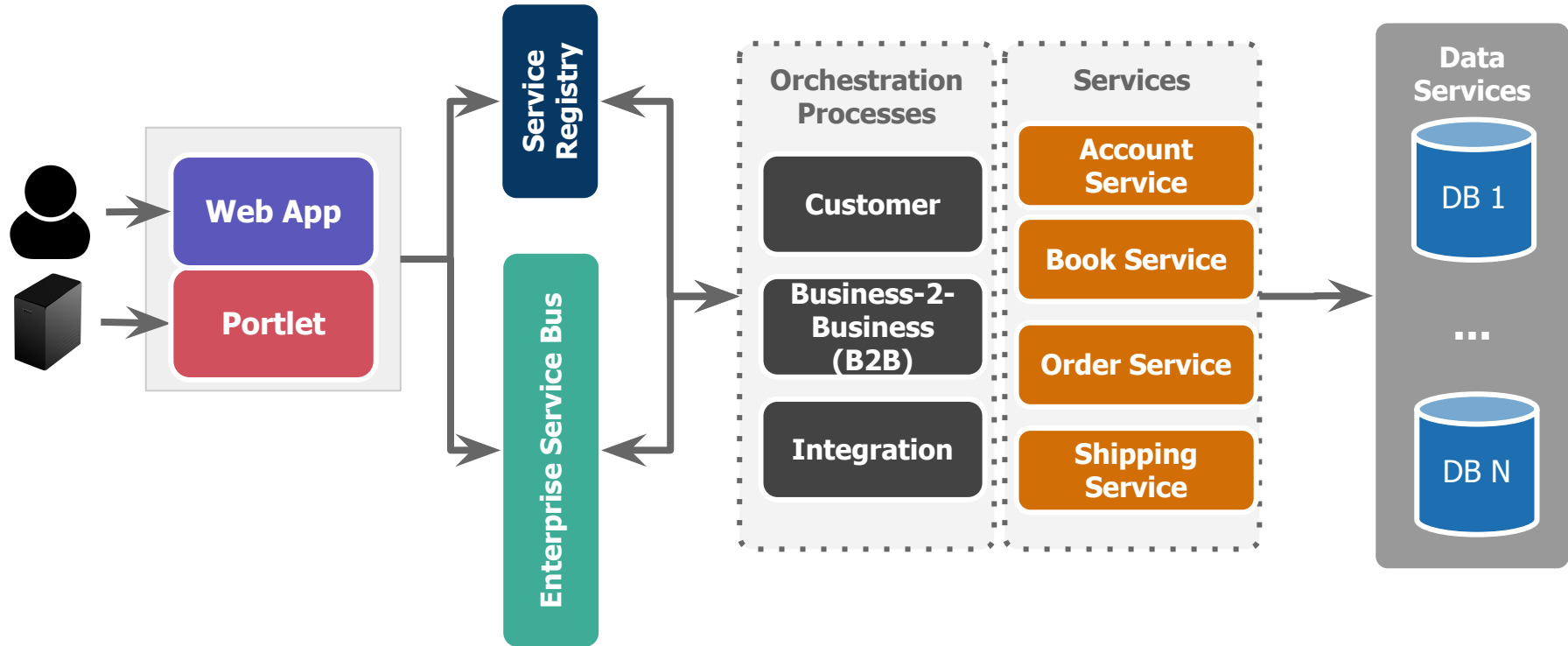
- + Routeo
- + Escalabilidad
- *Single point of failure*

Vista de *Data Queries*



- + *Throughput* en Lectura
- *Throughput* en Escritura

# Evolución en Arquitecturas | *Service Oriented Arch. (SOA)*





- SOA no es únicamente la definición arquitecturas sino un paradigma orientado al ámbito corporativo.
- Se complementa con el concepto de Business process management (BPM):

*"... discipline involving any combination of modeling, automation, execution, control, measurement and optimization of business activity flows, in support of enterprise goals, spanning systems, employees, customers and partners within and beyond the enterprise boundaries"*

Palmer, Nathaniel. "What Is BPM". bpm.com. Retrieved 20 January 2018.

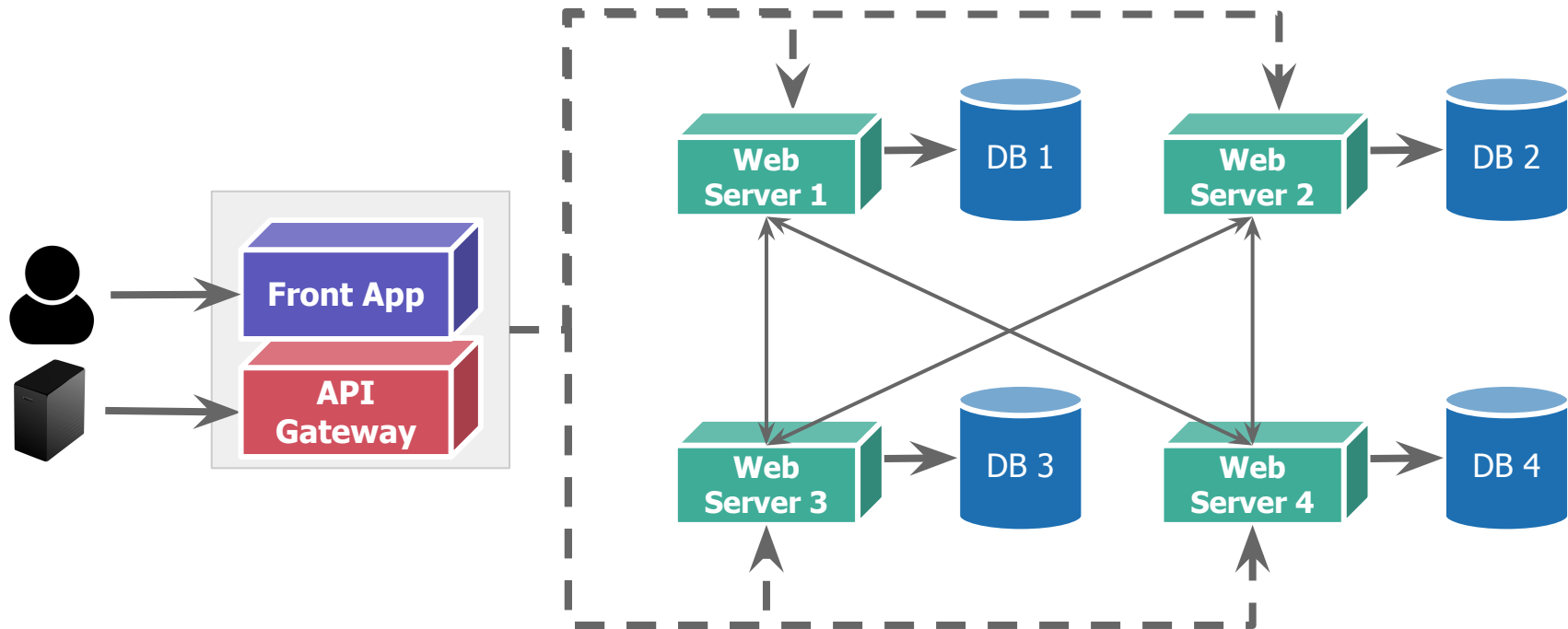


## Tecnologías

- WebServices = SOAP + HTTP
- ESB preponderante para eventos
- *Service Repository & Discovery* para comunicación punto a punto

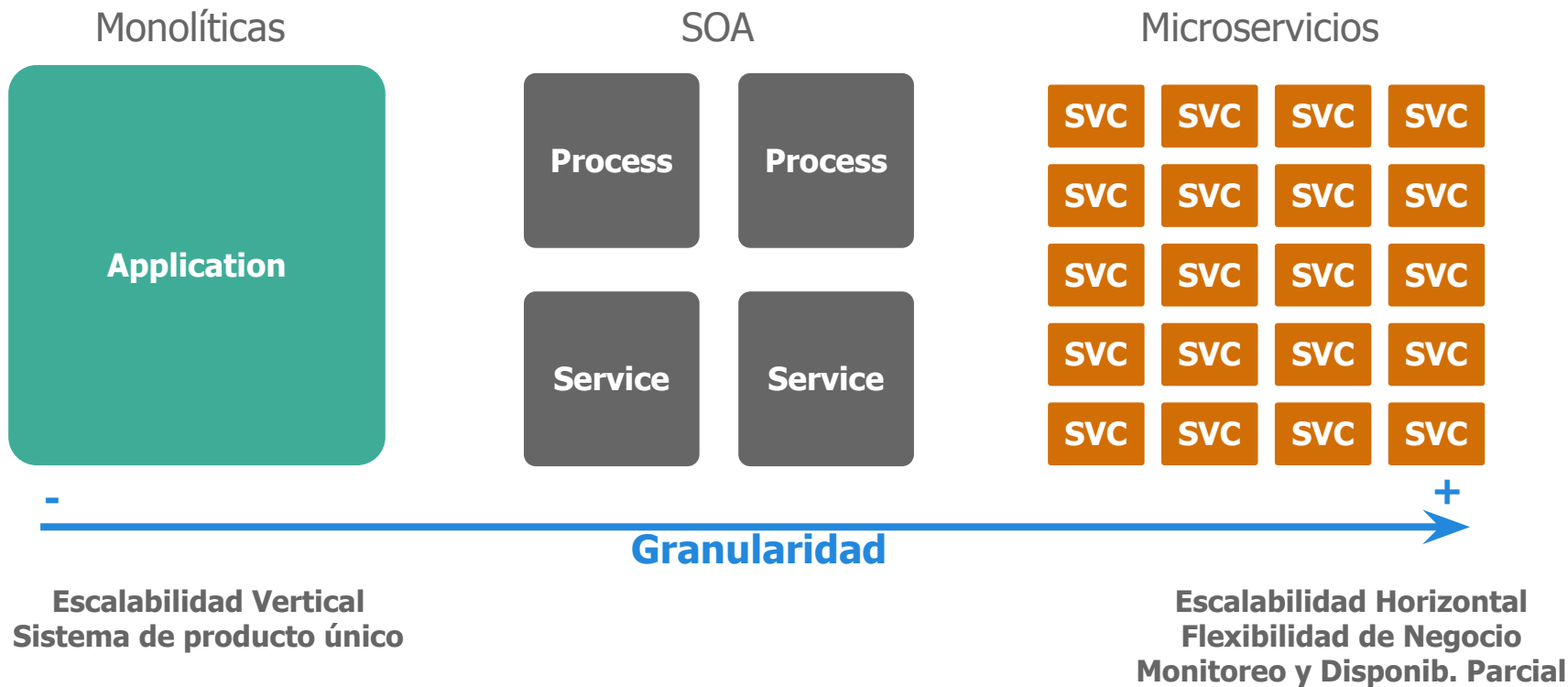
## Procesos y Servicios

- Contract
- Interface
- Implementation: Business Logic + Data Management



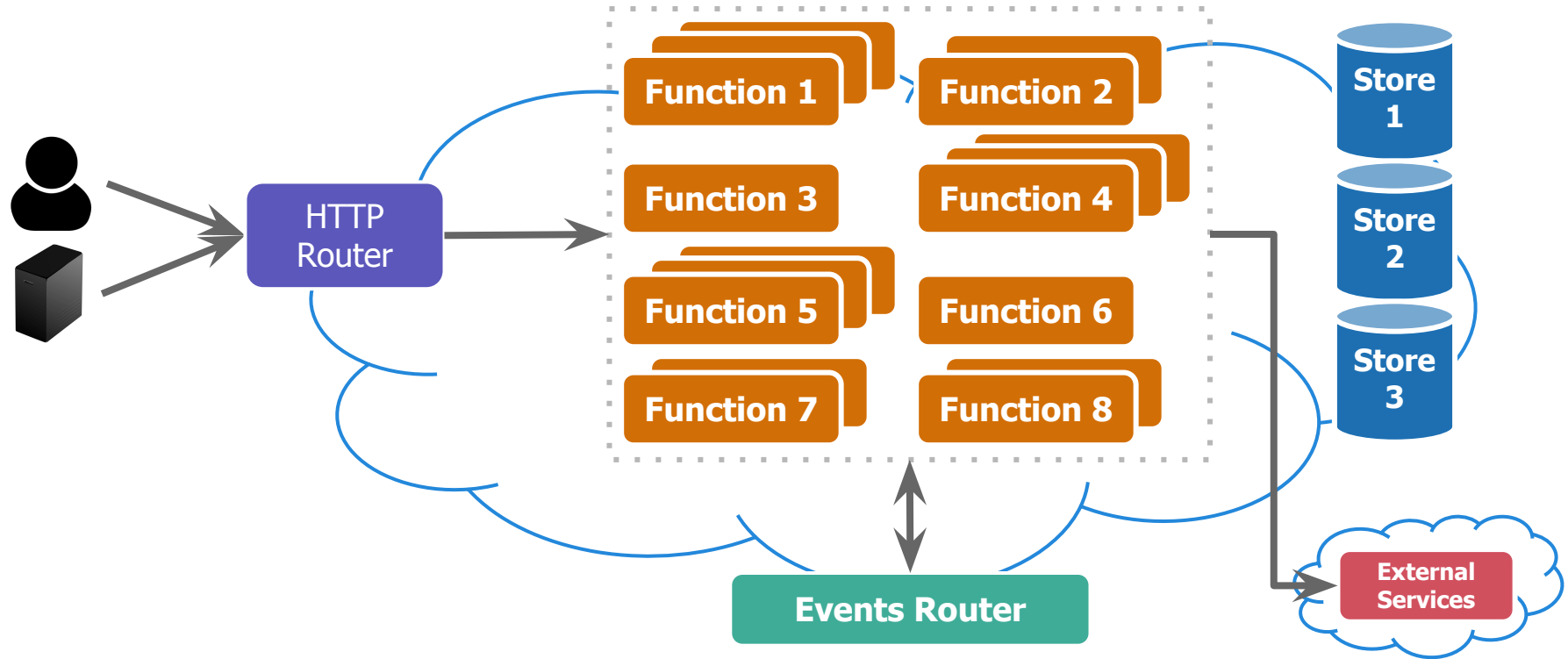


# Evolución en Arquitecturas | Transición entre Arquitecturas





# Evolución en Arquitecturas | *Serverless*





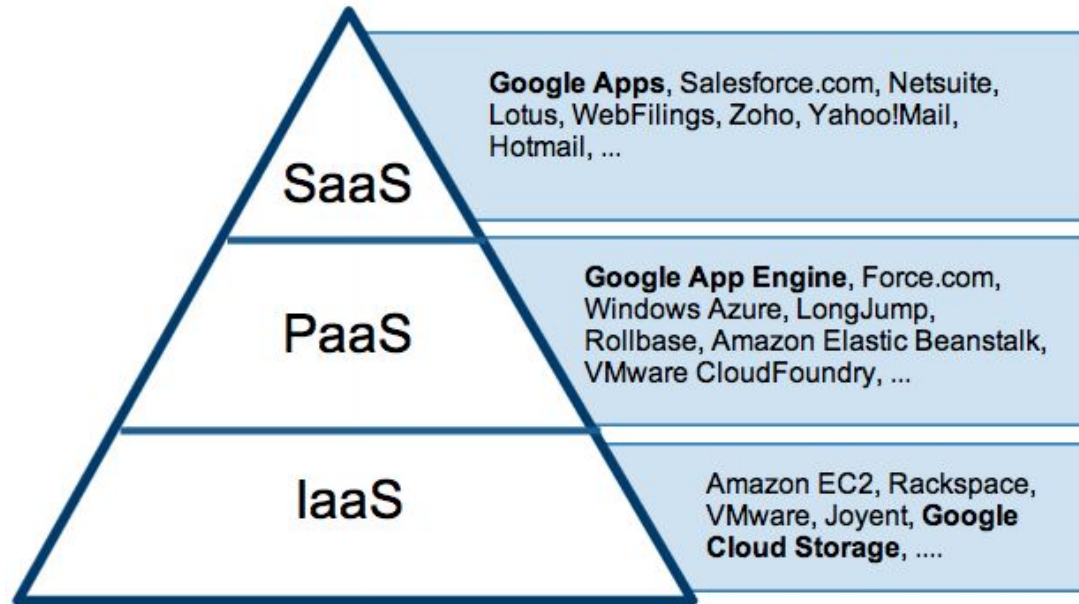
# Agenda



- Arquitecturas Orientadas a Servicio
- **Cloud**
- Platform as a Service (PaaS)
- Caso de estudio: Google AppEngine y Datastore



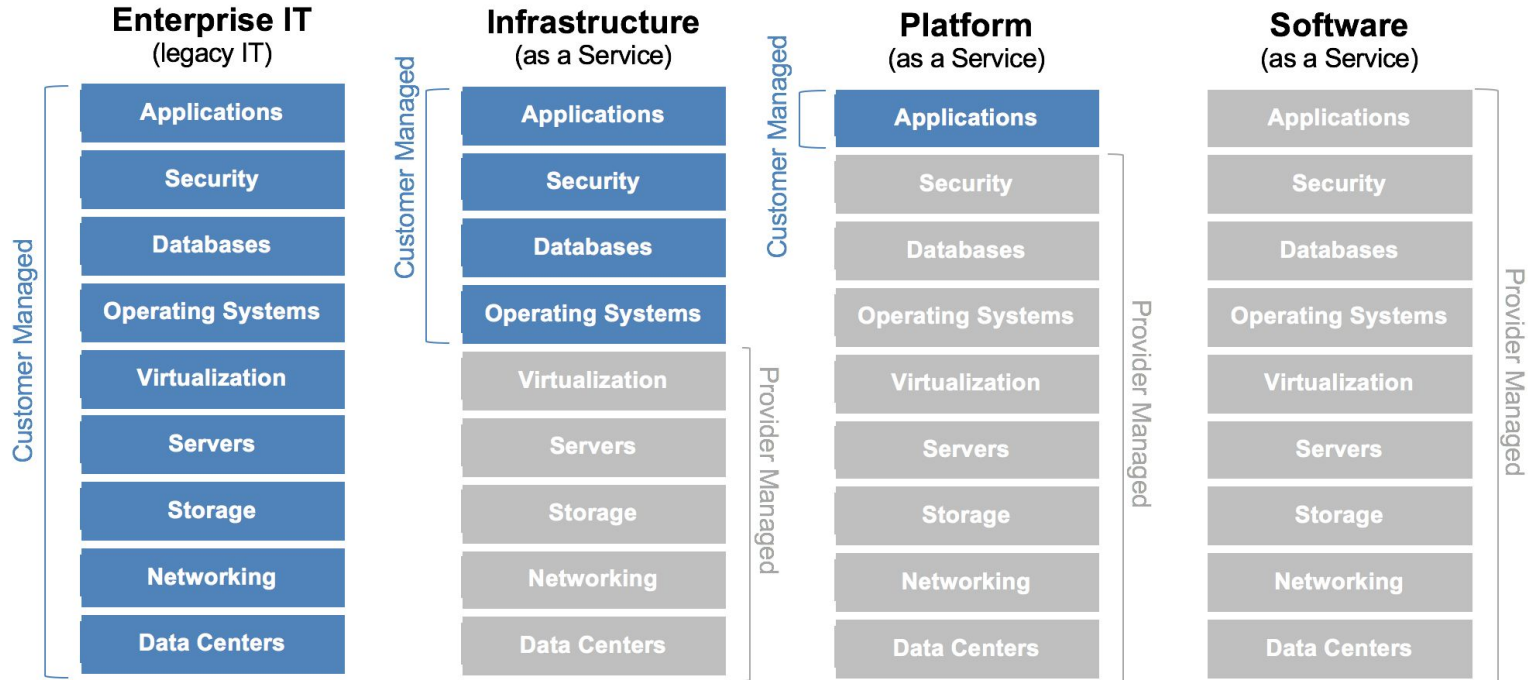
- Metáfora para internet y todo el contenido que ofrece.
- “Todo lo que se pueda consumir más allá del firewall...”
- Una forma de ofrecer recursos de IT, no necesariamente una nueva tecnología.
- Networking + Infraestructura + Nuevas Plataformas + Servicios



Source: Gartner AADI Summit Dec 2009



- **Infrastructure as a Service (IaaS)**
  - Almacenamiento y virtualización de equipos.
  - Definir redes y técnicas de adaptación para capacidad frente a cargas.
- **Platform as a Service (PaaS)**
  - Frameworks y plataformas para desarrollar aplicaciones *Cloud ready*.
  - Recursos expuestos como servicios para el desarrollo y el manejo del ciclo de vida de las aplicaciones (logs, *monitoring*, etc).
- **Software as a Service (SaaS)**
  - Software a demanda. Alquiler de servicios.
  - Ámplia variedad de soluciones genéricas y adaptables.
  - Protocolos abiertos y arquitecturas pensadas para integración.





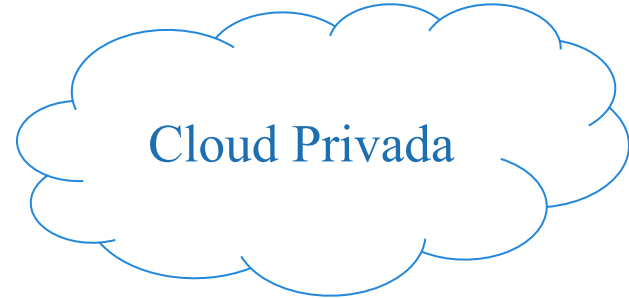


# Cloud | Principales Beneficios

- **Accesibilidad:** *Access anywhere.* Movilidad y visibilidad constante de los recursos.
- **Time-to-Market:** Disponibilidad instantánea de los recursos.
- **Escalabilidad:** Capacidades 'ilimitadas' de recursos para manejar volúmenes: almacenamiento, ancho de banda, cómputo, memoria, etc.
- **Costos:**
  - Pago a demanda (*Pay as you go*)
  - Control del gasto dependiendo del uso
  - Accesibilidad + escalabilidad + confiabilidad realmente barata



- Servicios públicos
- Servidores compartidos con otros usuarios
- Disponibilidad de recursos garantizados con SLAs
- Costos variables. *'Pay as you go'*
- Se accede mediante internet



- Servicios privados
- Datacenter propio de la empresa
- Recursos dedicados
- Costos fijos de mantenimiento y expansión
- Se accede mediante intranet



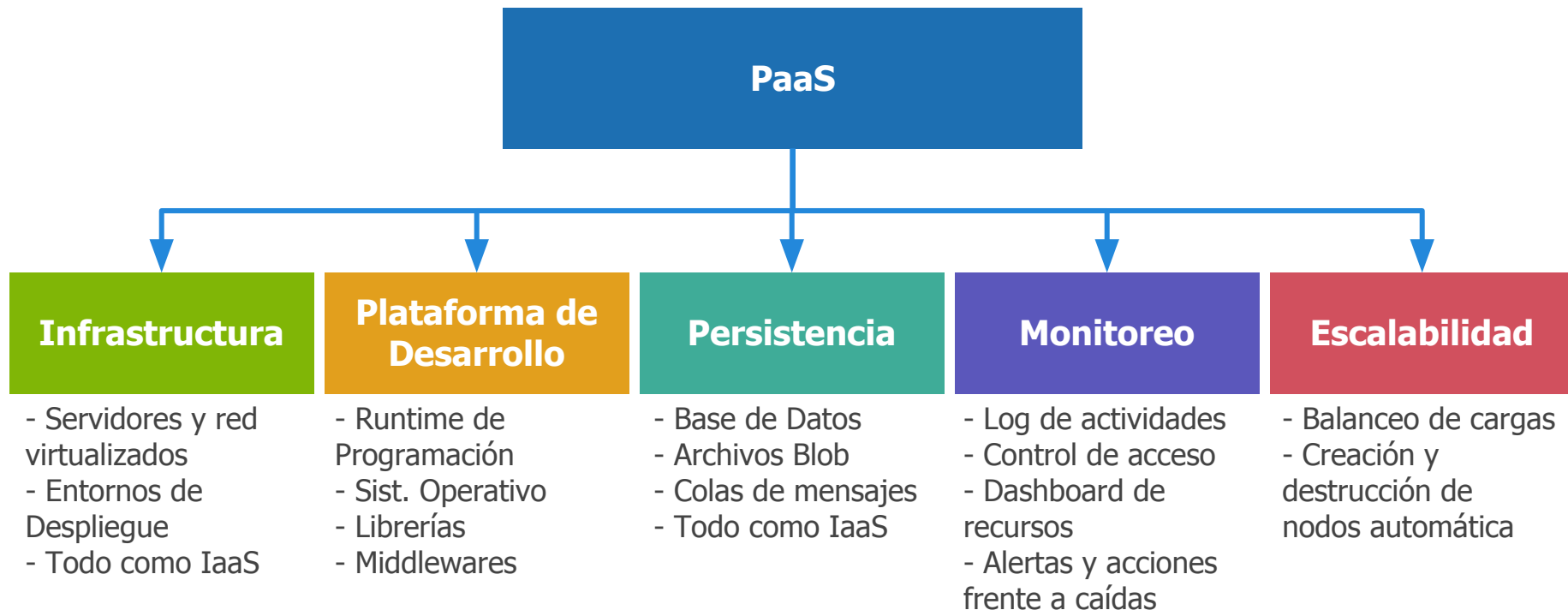
- Factores Políticos
  - Licenciamiento, jurisdicción y pérdida de gobernabilidad sobre los datos
  - Incapacidad de influir sobre la toma de decisiones que afecta al hardware
- Factores Técnicos
  - Costos monetarios y de tiempo para migraciones de datos y software
  - Alta exposición de datos sensibles
  - Dificultades en la construcción de los sistemas por esta “nueva” arquitectura

# Agenda

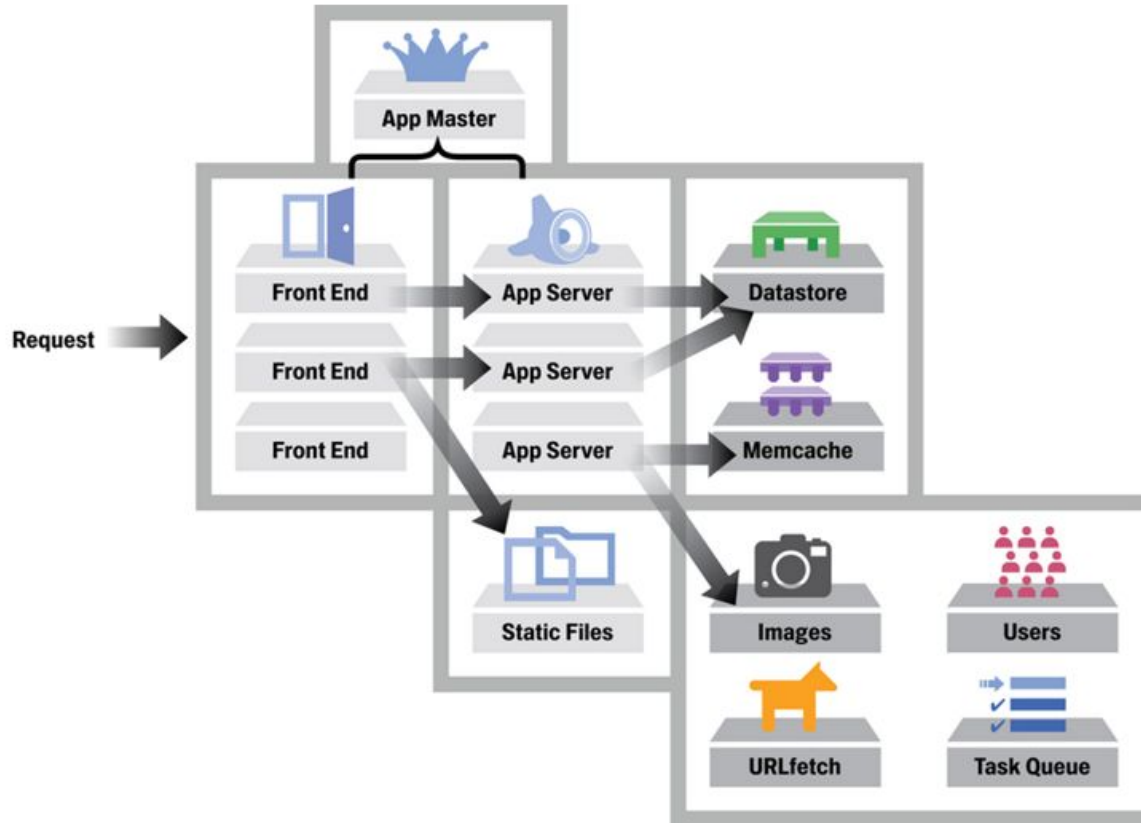


- Arquitecturas Orientadas a Servicio
- Cloud
- **Platform as a Service (PaaS)**
- Caso de estudio: Google AppEngine y Datastore

# Platform as a Service (PaaS)



# Infraestructuras Escalables | Modelo AppEngine



# Agenda



- Cloud
- Platform as a Service (PaaS)
- **Caso de estudio: Google AppEngine y Datastore**



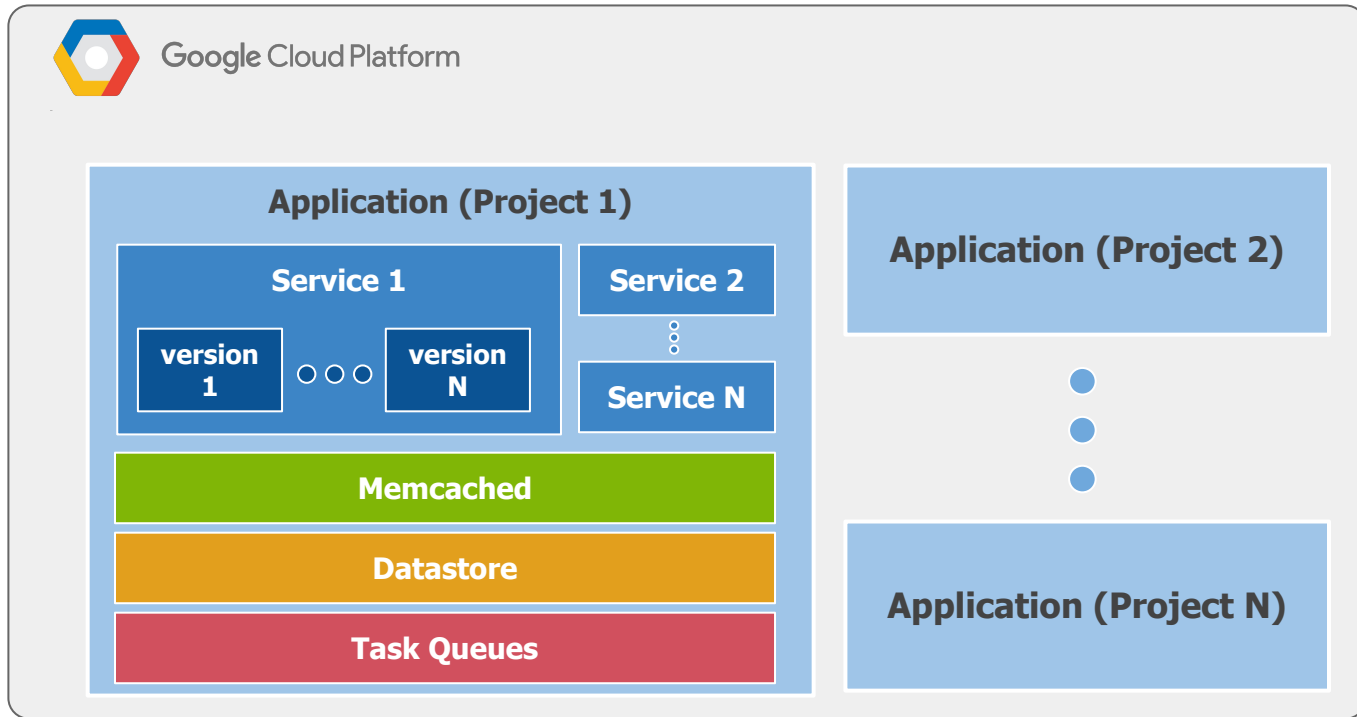
# Google AppEngine | Objetivos de Diseño

- Las buenas prácticas son forzadas
  - Sistemas granulares
  - Escalamiento horizontal
  - Requests breves con posibilidad de encolar Requests largos
  - Independencia del SO y Hardware
- Servicios de aplicación ya integrados:
  - Cache
  - Colas de mensajes
  - Elasticidad
  - Versionado
  - Herramientas de Log, *Debugging* y Monitoreo
  - Modelos No-Relacionales con Datastore/BigTable



**AppEngine**

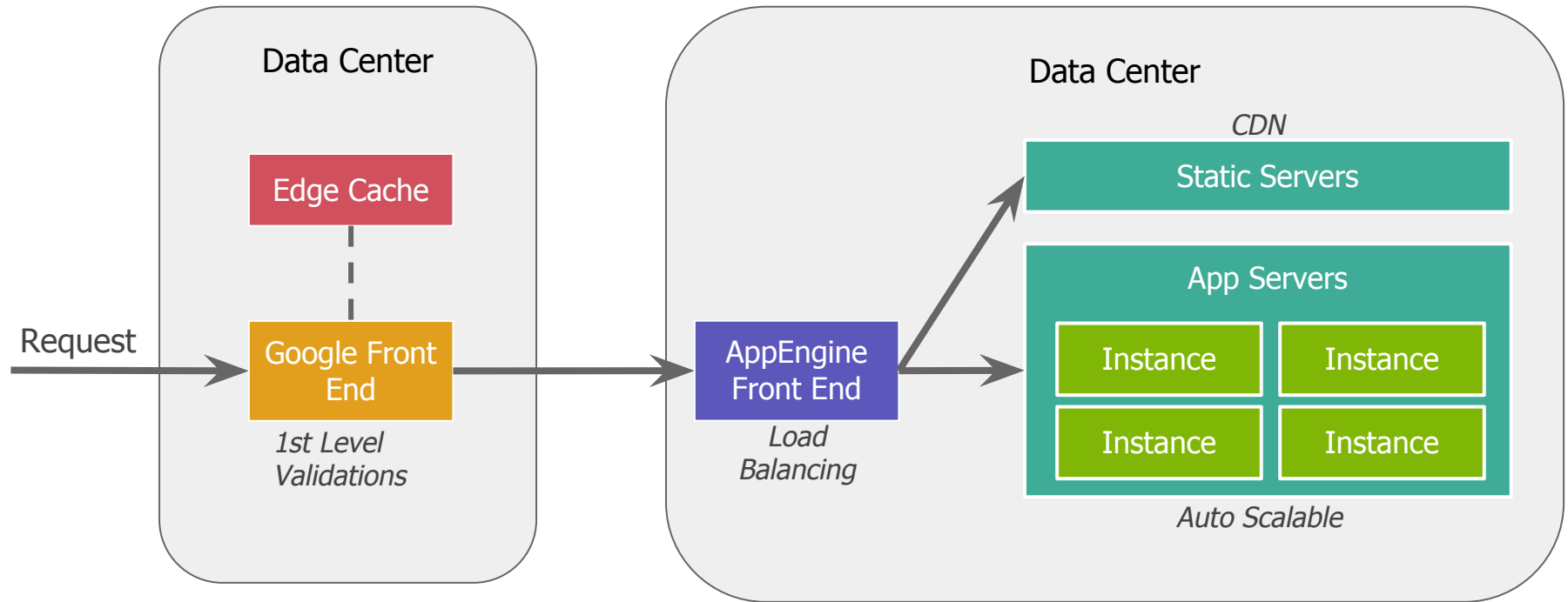






- Servicios = Módulos
  - Permite mantener unidad entre las operaciones soportadas.
  - Pueden desplegarse distintas versiones
  - Para cada versión puede existir una o más instancias
- Instancias = AppServers = Backend Servers
  - Son la unidad de procesamiento
  - Se las clasifica en Dinámicas o Residentes
    - Dinámicas: Son creadas por los requests
    - Residentes: Escaladas de forma manual

# Google AppEngine | Arquitectura Interna





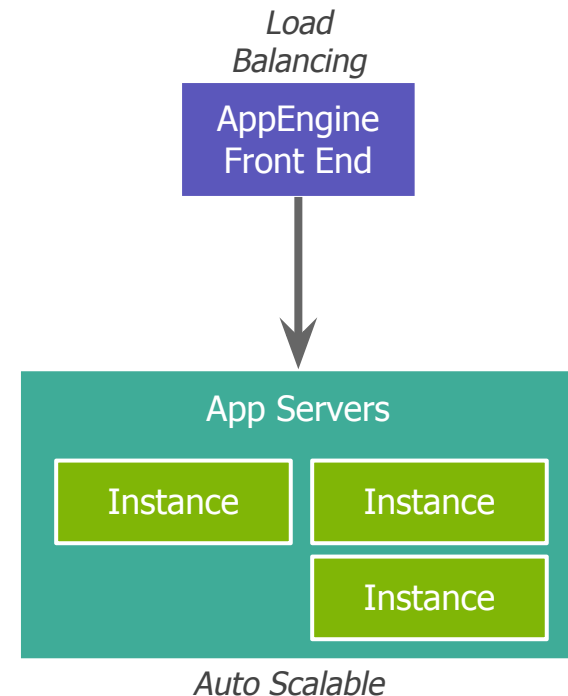
# Google AppEngine | Instancias de Procesamiento

## Instancias Dinámicas

- Se crean dinámicamente
- Procesan request pequeños
- Fuerzan respuestas rápida y manejo sin estado (stateless)
- Pueden aceptar requests externos (GET) e internos (mensajes en colas 'push')

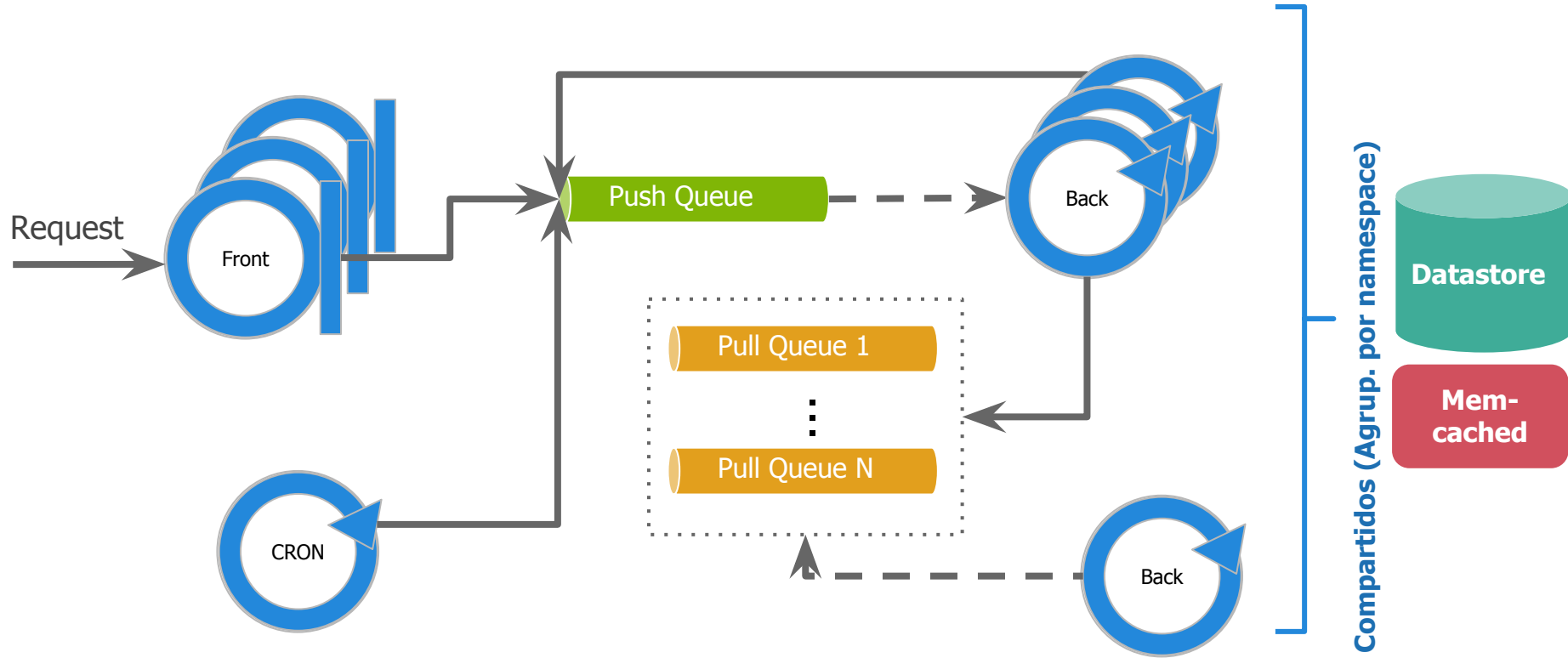
## Instancias Residentes

- Creadas de forma manual mediante configuración.
- No existen límites para su empleo y se pueden elegir su capacidad de cómputo
- Procesan requests largos, especialmente en *batches* con o sin estado.





# Google AppEngine | Comunicación Interna





## Push Queue



- Nombre
- URL
- Headers

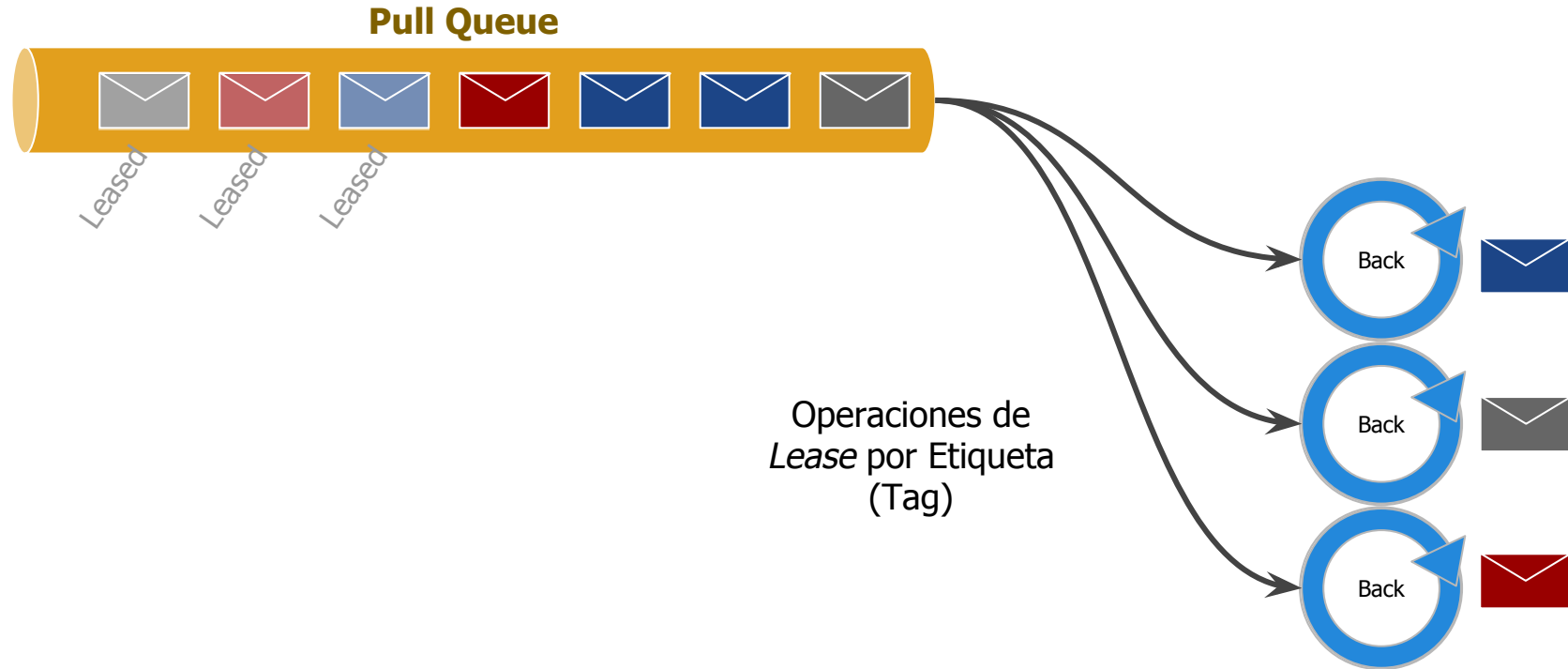
- Envían los requests a instancias activas
- La URL define la instancia, servicio y versión p/atender el mensaje
- El payload está dado por los argumentos en la URL y headers

## Pull Queue



- Nombre
- *Payload*
- Etiqueta

- Permiten encolar tareas a ser consumidas de forma controlada.
- Es el enfoque más usual de colas de tareas y se puede reemplazar por colas PubSub





- Base no-SQL administrada por Google y accesible en modalidad PaaS
  - Modelo de objetos con atributos: *Entities*
  - Tipos sin esquema con atributos indexables: *Kinds*
- Altamente escalable (volumen de datos)
  - Basada en BigTable (que es soportada por un Google FS: Colossus)
  - Soporta entidades de hasta 1MB
  - Soporta consultas por Key o por atributos indexados.
  - Millones de escrituras por segundo. No es bueno para consultas.
  - Operaciones ACID incluso entre entidades.

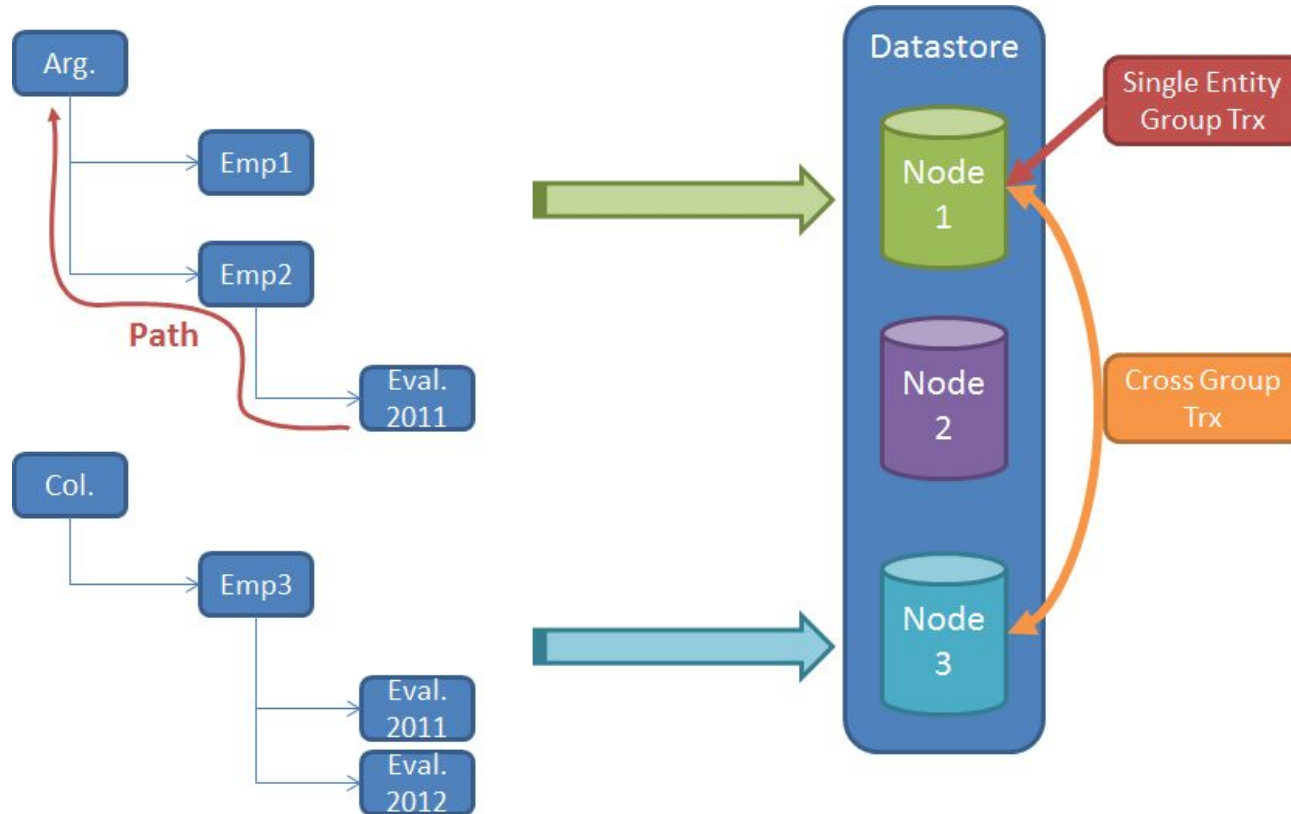


**Datastore**





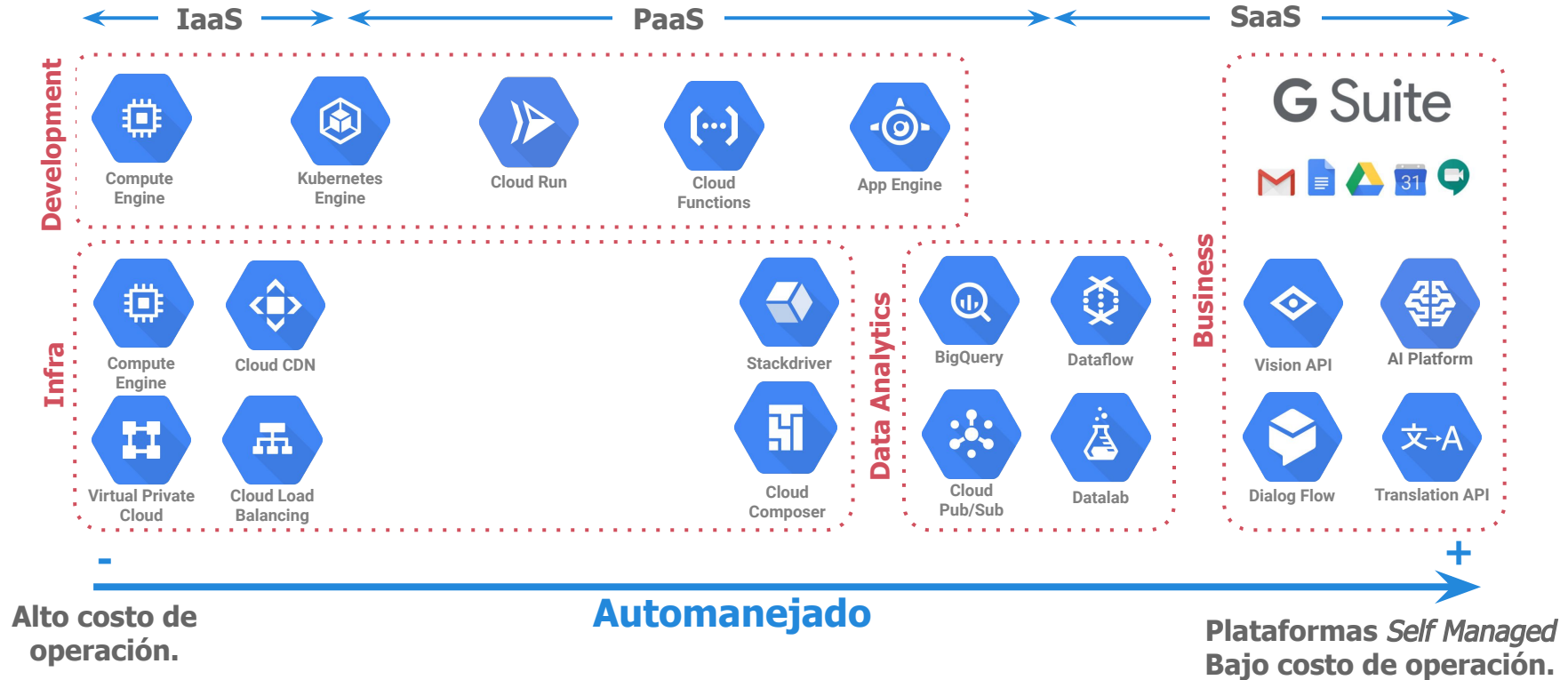
# Google AppEngine | Particionamiento del Almacenamiento





# Google Ecosystem

Target Audience





# Bibliografía

- Google I/O 2011: Scaling App Engine Applications, Justin Haugh, Guido van Rossum, 2011
  - <https://www.youtube.com/watch?v=rP-kjrx9CRE&t=1589s>
- Writing Infinitely Scalable and High Performance Apps with AppEngine, Karan Goel, 2017
  - <https://www.youtube.com/watch?v=WpWPMMC0q3E>
- Google Cloud Datastore Inside-out, Etsuji Nakai, 2017:
  - <https://www.slideshare.net/enakai/google-cloud-datastore-insideout>
- App Engine datastore tip: monotonically increasing values are bad, Ikay Lan, 2011:
  - <https://ikaisays.com/2011/01/25/app-engine-datastore-tip-monotonically-increasing-values-are-bad/>