En Uber

La arquitectura de Uber es un poco diferente, basada en microservicios. Esta filosofía les ha permitido crecer muy rápido y escribir cada uno de estos microservicios en lenguajes muy diferentes. Uber utiliza Node.js, Python, Go, Java y Scala, lo que además les permite contratar ingenieros de forma más rápida al no verse limitados por el lenguaje.

Los microservicios permiten a las empresas crecer rápidamente, pero tienen un costo en términos de velocidad agregada. Brindan muchos beneficios, como equipos que poseen sus propios ciclos de lanzamiento y son responsables de su propio tiempo de actividad.

Adoptar microservicios le permite escribir su software en diferentes lenguajes de programación; para que pueda tener algunos escritos en Node.js, Python, Go y Java.

Pero el usar tantos lenguajes diferentes se convierte en una limitación cuando se tiene que solucionar el mismo problema tantas veces como lenguajes existen. La reutilización se convierte en algo muy complejo.

Debido a que cada equipo hace lo suyo, la velocidad agregada en muchos casos sería más lenta, la gente de Java tuvo que descubrir cómo comunicarse con el sistema de métricas, al igual que las personas de Node,js y Go.

Uber está moviendo gradualmente su desarrollo de mercado de Node.js a Go y Java.

Java se usa para los servicios de mapas. Se encuentra en muchos centros de datos diferentes en todo el mundo, para acercar los datos a los clientes y por razones de disponibilidad. La razón parece ser porque sus sistemas más críticos y que requieren un mayor rendimiento, como son los que calculan rutas y tiempos estimados, están escritos en java.

Luego tenemos el sistema de despacho Node.js, ahora llamado Marketplace, mientras Uber se expande hacia otros tipos de logística más allá del transporte. Marketplace se está moviendo gradualmente de Node.js a Go y Java.

La prueba de falla agresiva se usa ampliamente en Uber.

Algunas elecciones tempranas de diseño, como el uso de JSON sobre HTTP, hacen que la verificación formal sea básicamente imposible.

**Gestionar el crecimiento y la cultura del equipo**

Un crecimiento tan grande de equipo y producto no sería posible sin la posibilidad de contratar ingenieros realmente buenos. Sería mucho más eficiente si se contratara más despacio, pero la competencia es fuerte lo que demanda trabajar duro para no quedarse atrás.

Una de las cosas interesantes es que, debido a que estábamos agregando gente tan rápido, no hubiera sido posible si no hubiéramos creado cosas como muchos servicios pequeños.

Por lo que comenta Matt Ranney, nunca se habrían imaginado el coste que suponía el escribir los microservicios en lenguajes tan diferentes.