zeebe

BPMN y Microservices Orchestration, Parte 1 de 2: Lenguajes de flujo, motores y patrones atemporales

por **Mike Winters** el 1 de agosto de 2018 en Inside Zeebe .

Un sincero agradecimiento a Bernd Rücker por sus comentarios durante la redacción de esta publicación de blog.

Esta es la parte 1 de 2 en una serie de publicaciones de blog de 2 partes. La parte 2 está disponible aquí .

Estamos construyendo Zeebe para que sea un motor de flujo de trabajo de próxima generación para casos de uso emergentes, como la orquestación de microservicios, casos de uso que pueden requerir un motor para manejar cientos de miles (o millones) de nuevas instancias de flujo de trabajo por segundo.

Y para hacer eso, estamos utilizando un estándar de modelado gráfico que existe desde hace casi 15 años: BPMN (Business Process Model and Notation).

A pesar de que BPMN es un estándar ISO probado en batalla, es posible que muchos de ustedes nunca lo hayan practicado o tal vez ni siquiera hayan oído hablar de él.

O peor, has oído hablar de BPMN, pero lo has descartado como una tecnología heredada que solo es relevante en un mundo monolítico o SOA.

El mes pasado, nos encontramos con esta cita de un comentarista de Hacker News (a quien le prometemos que no es un miembro incógnito del equipo de Zeebe):

zeebe

No podríamos estar más de acuerdo, y con ese espíritu, publicaremos una serie de blogs de dos partes sobre BPMN y la orquestación de microservicios, y más específicamente, por qué BPMN es ideal para casos de uso prometedores en el flujo de trabajo. mundo.

En esta parte 1, nosotros:

- Proporcionar una introducción rápida a BPMN
- Explique por qué un estándar bien establecido que prosperó en el pasado también puede prosperar en el futuro
- Revise los patrones de orquestación comunes compatibles con BPMN
- Discutir el estado actual y los planes futuros para BPMN en Zeebe

En la parte 2, nosotros:

- Sumérgete en los modelos gráficos de BPMN (y otras formas de definir flujos de trabajo)
- Mire ejemplos donde el uso de un modelo gráfico en lugar de un modelo basado en código simplifica enormemente la definición del flujo de trabajo

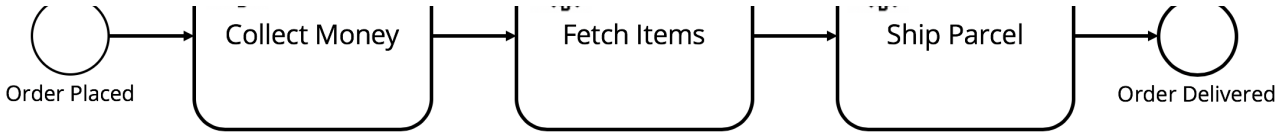
Una breve introducción a BPMN

BPMN es un estándar de modelado ampliamente utilizado para definir y ejecutar procesos de negocio. Lanzado por primera vez en 2004 (con la moderna especificación BPMN 2.0 que sigue en 2011, esto es lo que usa Zeebe), BPMN ha sido un estándar ISO desde 2013.

BPMN se utiliza para definir un modelo gráfico *y* la llamada semántica de ejecución. En otras palabras, el modelo visual se almacena como un archivo XML que se puede ejecutar directamente en un motor que mantiene el estado persistente de las instancias de flujo de trabajo en ejecución.

Para dar un ejemplo, el siguiente modelo se expresa con este XML .

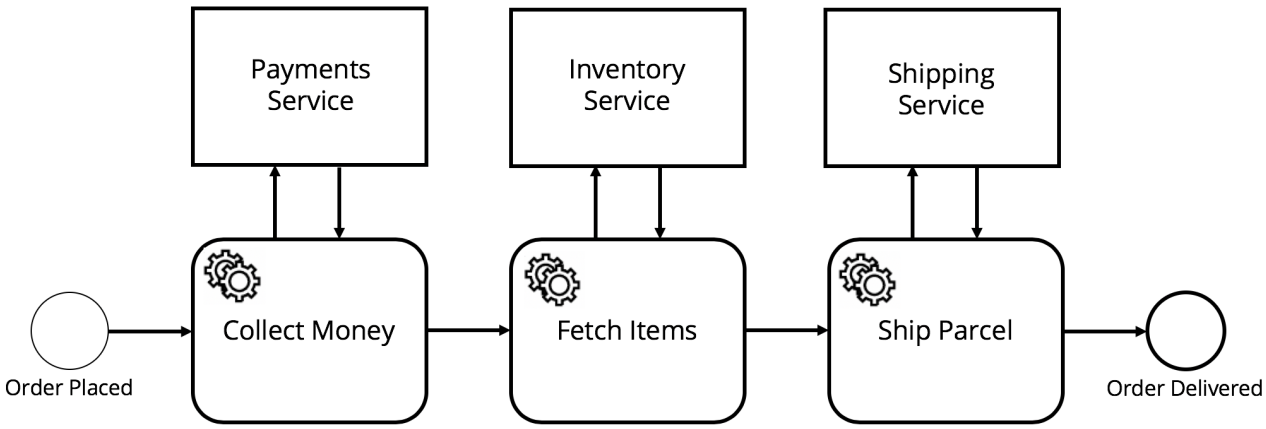
zeebe



¡Es importante decir que BPMN no implica generación de código ni transformación! El XML es en sí mismo el código fuente. Y BPMN solo se preocupa por el flujo: puede usar código normal para todos los demás aspectos de su solución.

Ese es un punto clave para la orquestación de microservicios, donde los trabajadores externos realizan las tareas en su flujo de trabajo. Cuando se combina con el motor adecuado, BPMN facilita la conexión de tareas en un flujo de trabajo a microservicios y hacerlo de una manera que no viole los principios de acoplamiento flexible e independencia de servicio.

Extendiendo el flujo de trabajo de pedidos de muestra anterior, podemos construir 3 microservicios distintos para manejar pagos, inventario y envío. El motor de flujo de trabajo es responsable de enviar el trabajo al servicio correcto en el punto correcto del proceso.



Por último, está la madurez de BPMN. BPMN es popular y está bien establecido, y ha demostrado su valor en muchos proyectos de automatización de flujo de trabajo en empresas grandes y pequeñas. Por esta razón, ya hay una gran cantidad de talentos experimentados de BPMN en el mercado, así como tutoriales y libros que facilitan a los recién llegados aprender el estándar.

zeebe

mi nueva arquitectura elegante?

Pasemos al modo metáfora por un momento.

Si bien los detalles sobre la historia del automóvil están en debate, muchos le dan crédito a Karl Benz por construir el primer automóvil y darle una vuelta por Mannheim, Alemania en 1886. En los últimos 130 años, las capacidades del automóvil, en este caso, literalmente el "motor", ha evolucionado bastante.

El "flujo", sin embargo, se ha mantenido bastante estático. Las carreteras, la señalización y las leyes que nos ayudan a llegar con seguridad desde el punto A al punto B siguen muchos de los mismos patrones que se implementaron hace muchas décadas. Esto podría cambiar eventualmente (ver: Hyperloop, drones), pero el punto es que un "patrón de flujo" dado puede soportar muchos avances significativos del "motor".

Entonces, cuando evaluamos un estándar bien establecido como BPMN, tenemos que distinguir entre "requisitos de flujo" y "requisitos del motor".

BPMN ha destilado una serie de patrones en torno a flujos que son intemporales. La realización de una serie de actividades en una secuencia o en paralelo se puede aplicar a un caso de uso BPMN más tradicional, como la gestión de tareas humanas, así como llamar a funciones sin servidor en AWS. Esperar una copia entrante de documentos impresos y firmados es comparable en cuanto a patrones para correlacionar múltiples mensajes en su arquitectura de transmisión de eventos.

Lo que sí cambia es el rendimiento (número de instancias de flujo de trabajo), así como los requisitos de rendimiento y escalabilidad. Estos son problemas que se pueden resolver con un *nuevo motor que* ejecuta el *mismo lenguaje de flujo,* y ese es el enfoque que estamos adoptando con Zeebe, que puede escalar a millones de nuevas instancias de flujo de trabajo por segundo.

La alternativa es construir un nuevo motor *e* inventar un nuevo lenguaje de flujo mientras lo hace. Pero con un nuevo lenguaje de flujo, inevitablemente pasará tiempo resolviendo problemas que ya se han resuelto en BPMN. Y es posible que

zeebe

Desarrollaremos esta idea en la parte 2 de esta serie, donde compararemos un patrón de flujo de trabajo complejo del mundo real como se expresa en BPMN versus Amazon States Language.

Por ahora, revisemos ejemplos de patrones de flujo de trabajo comunes para ayudar a demostrar por qué estamos tan seguros de que BPMN es el lenguaje de flujo correcto para la orquestación de microservicios y otros casos de uso de flujo de trabajo de próxima generación.

Definición de patrones de orquestación en BPMN

No vamos a proporcionar un tutorial completo de BPMN en esta publicación. Nuestro objetivo es que comprenda un subconjunto de los bloques de construcción a su disposición y brinde algunos ejemplos de cómo puede usarlos.

Eso no debería impedirle profundizar en BPMN si lo desea. El tutorial BPMN de Camunda que mencionamos anteriormente es un buen lugar para comenzar, como lo es nuestra referencia BPMN .

También puede comenzar a utilizar nuestra herramienta de modelado gráfico específica de Zeebe , y hablaremos mucho más sobre los modelos gráficos en la parte 2 de esta serie.

Ahora en los patrones.

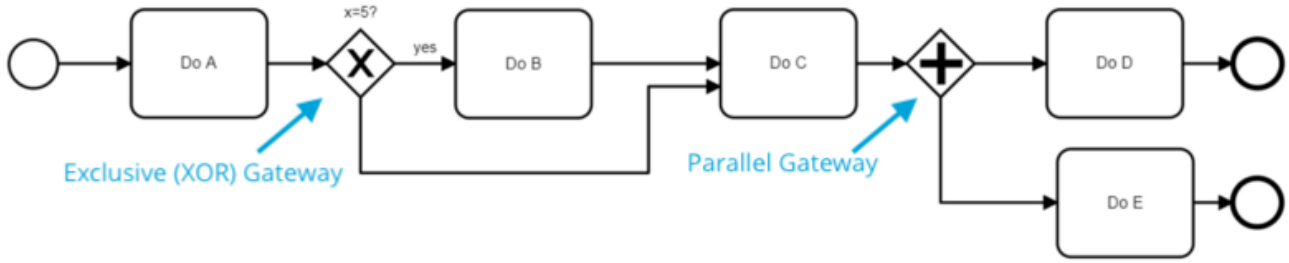
Flujo de secuencia, decisiones y procesamiento paralelo.

En el núcleo de BPMN se encuentra el flujo de secuencia, que define el orden en que se llevan a cabo los pasos en un flujo de trabajo.

Como puede imaginar, limitar un flujo de trabajo a una simple secuencia de tareas una tras otra deja mucha lógica empresarial del mundo real sin abordar. Además de las tareas (unidades de trabajo), los flujos de trabajo BPMN consisten en puertas de enlace (que dirigen el flujo) y eventos (que representan *cosas que*

zeebe

BPMN proporciona construcciones para enrutar una instancia de flujo de trabajo a un único flujo de secuencia basado en datos asociados (la puerta de enlace exclusiva) y también para uno o más flujos de secuencia que deben llevarse a cabo en paralelo (la puerta de enlace paralela).



Correlación de mensajes con tiempo de espera

La tarea de recepción de BPMN es una de las formas en que el estándar proporciona soporte para la correlación de mensajes, una característica muy poderosa que hace posible mover una instancia de flujo de trabajo en espera hacia adelante o tomar alguna otra acción *solo* cuando un mensaje puede coincidir correctamente ("correlacionarse") con un instancia de flujo de trabajo específica que lo espera con un identificador común.

Esta es una característica que sería especialmente difícil de construir desde cero y luego admitir a escala, y con BPMN combinado con el motor adecuado, lo obtienes de fábrica.

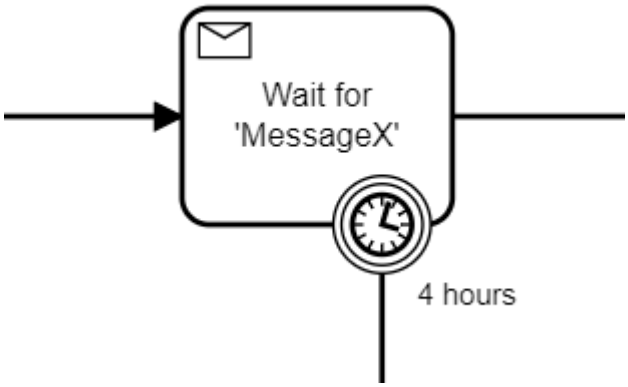
El soporte de BPMN para enviar y recibir mensajes significa que los modelos pueden integrarse perfectamente con una arquitectura basada en mensajes, una arquitectura que es particularmente común en el mundo de los microservicios.

zeebe

instancia de flujo de trabajo puede finalizar en función de un mensaje recibido. Por ejemplo, una instancia de flujo de trabajo en vuelo, como un proceso de cumplimiento de pedido en una empresa de comercio electrónico, se puede finalizar en respuesta a un mensaje de cancelación de pedido entrante asociado con un pedido específico.

Como verá (y repetiremos con frecuencia), la facilidad con la que se pueden combinar diferentes elementos es lo que hace que BPMN sea tan poderoso.

Por ejemplo, la tarea de recepción se puede combinar con un evento de temporizador para que si el mensaje requerido no llega dentro de las 4 horas, la tarea se "agota" y la instancia de flujo de trabajo sigue una ruta diferente.

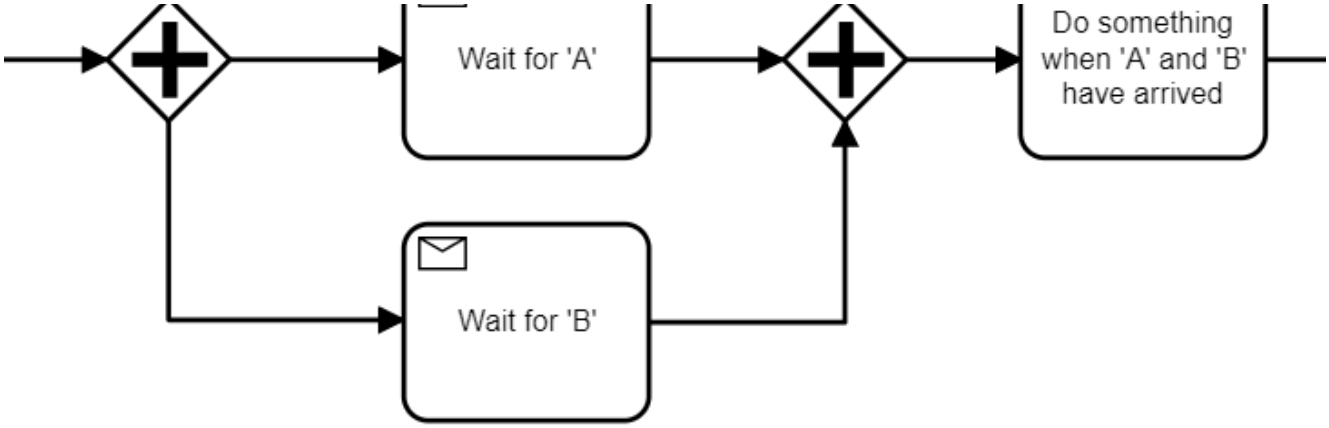


Correlación de múltiples mensajes.

Correlacionar un mensaje con una instancia de flujo de trabajo es útil, pero ¿qué sucede si necesita correlacionar dos, tres o diez?

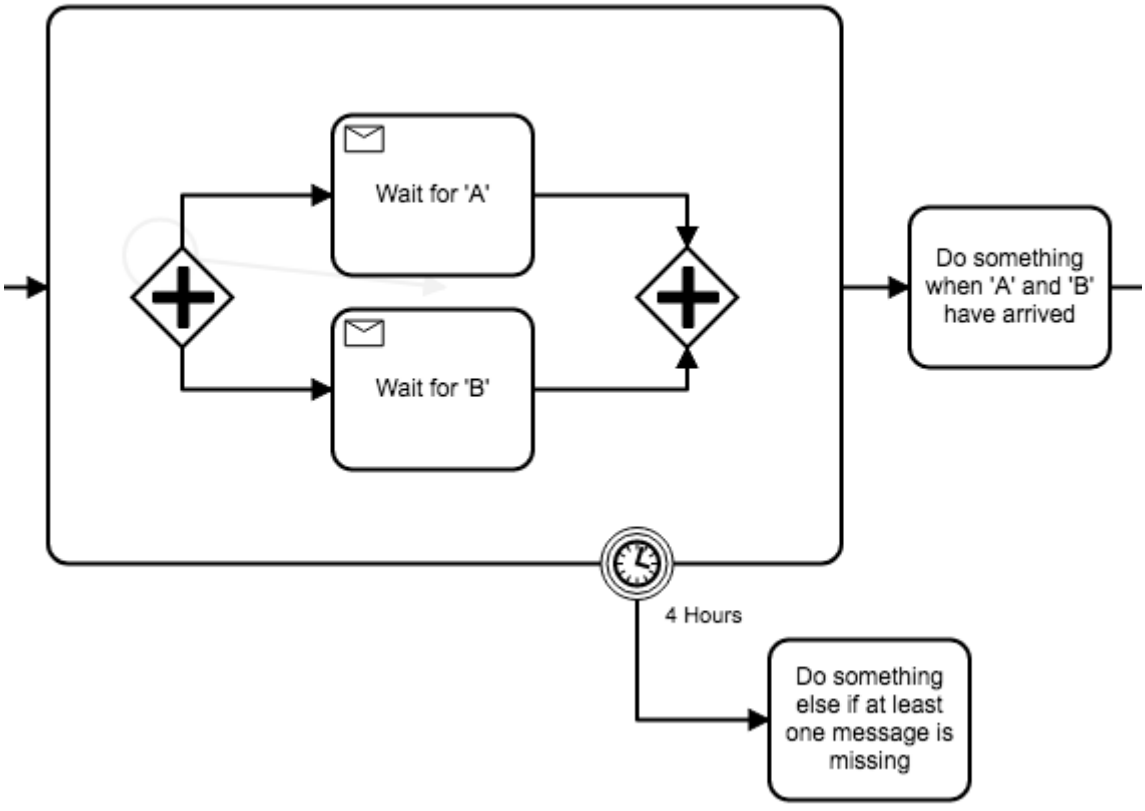
BPMN también tiene cubierto este patrón. Puede esperar a que dos o más mensajes se sincronicen y fusionen sus cargas antes de mover una instancia de flujo de trabajo hacia adelante combinando la tarea de recepción con la puerta de enlace paralela.

zeebe



Avancemos un paso más y combinemos *este* patrón con un tiempo de espera.

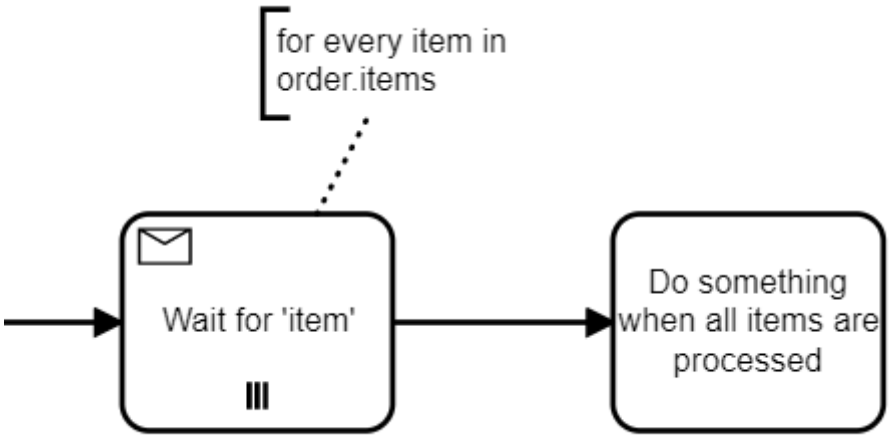
Nuevamente, la adición del temporizador y el subproceso en el diagrama a continuación es solo un ejemplo de cómo se pueden combinar diferentes elementos BPMN para expresar un flujo complejo; Si bien, por supuesto, hay ciertas combinaciones que no tienen sentido lógico, esencialmente no hay límite para cómo puede conectar símbolos BPMN para definir flujos de trabajo.



Esperando un número arbitrario de mensajes

zeebe

Considere un ejemplo en el que necesitamos recibir un `itemAvailable` mensaje para cada artículo en un pedido antes de avanzar con el flujo de trabajo. El número de elementos en cada orden puede variar ampliamente, y podemos dar cuenta de eso en nuestro modelo utilizando la actividad de instancias múltiples de BPMN .

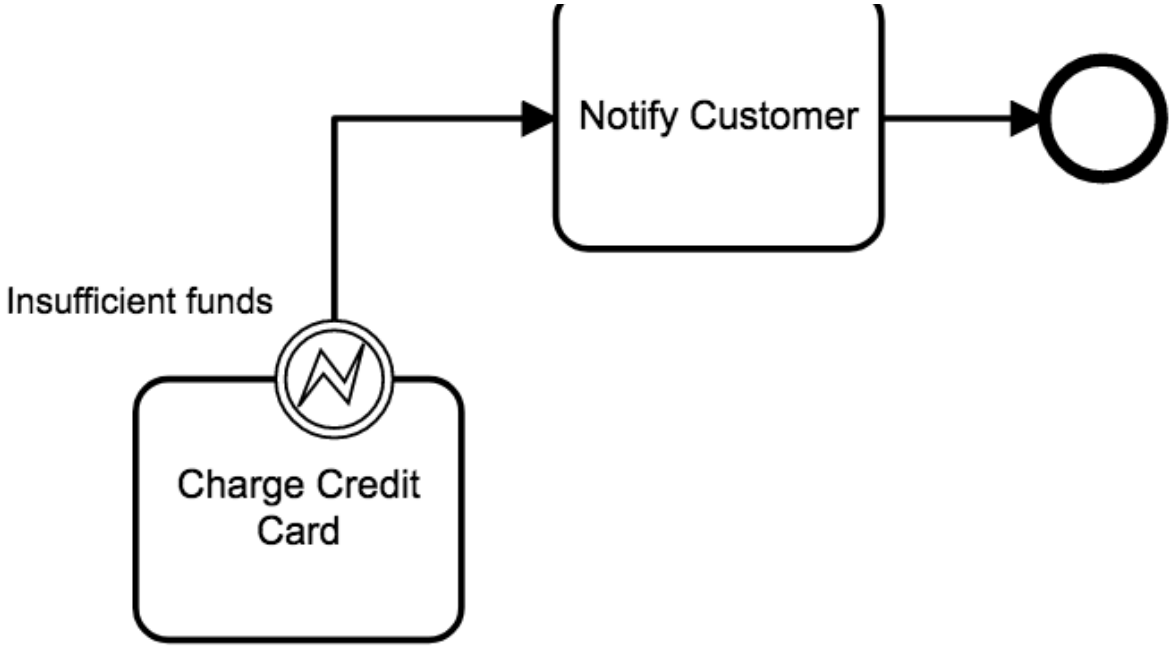


Manejo de errores

Puede haber ciertos "errores de lógica de negocios" que necesitará diseñar en sus flujos de trabajo. Aquí, no estamos hablando de errores en los que un servicio falla por una razón técnica, sino de casos en los que un flujo de trabajo no puede continuar debido a un problema comercial que podemos planificar con anticipación. El evento de límite de error de BPMN fue diseñado para este caso particular.

En este ejemplo, intentamos cargar la tarjeta de crédito de un cliente. Si el cargo se rechaza debido a fondos insuficientes en la cuenta del cliente, se lo notificaremos al cliente sobre el problema.

zeebe



Podríamos seguir yendo ...

Cuando se trata de patrones compatibles con BPMN, solo hemos arañado la superficie. Esperamos que tenga una idea de cuántos casos de uso diferentes se pueden expresar solo con estos símbolos BPMN.

Lo que mostramos en estos ejemplos no pretende ser prescriptivo o decirle exactamente cómo debe usar BPMN. Más bien, nuestro objetivo es despertar tu imaginación sobre los tipos de modelos que puedes construir.

El estado de BPMN en Zeebe

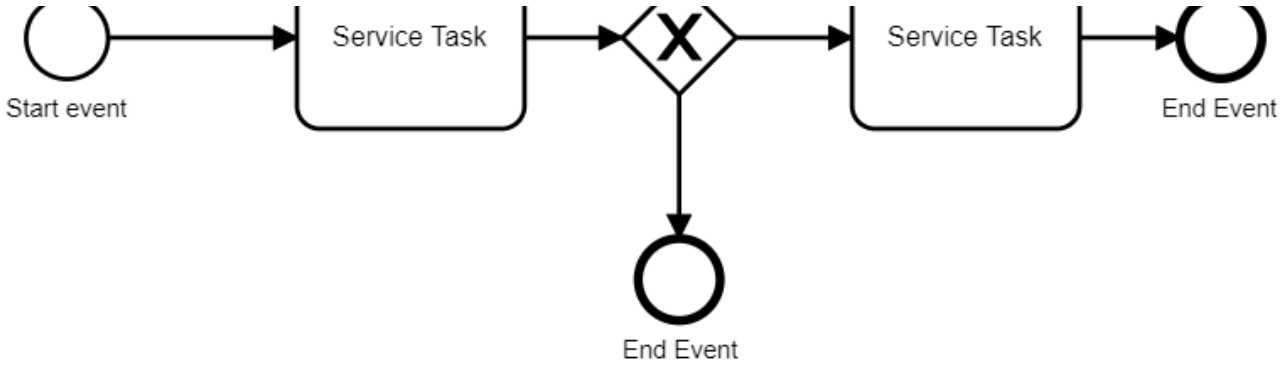
Con suerte, ha llegado a este punto en la publicación con una comprensión de las posibilidades de BPMN a la hora de definir y ejecutar flujos de trabajo complejos.

Pero la verdadera pregunta es: ¿qué cantidad de BPMN admitimos en Zeebe?

A largo plazo, Zeebe admitirá todos los símbolos que tengan sentido para la automatización del flujo de trabajo, al igual que lo hemos hecho con el motor de flujo de trabajo Camunda BPMN .

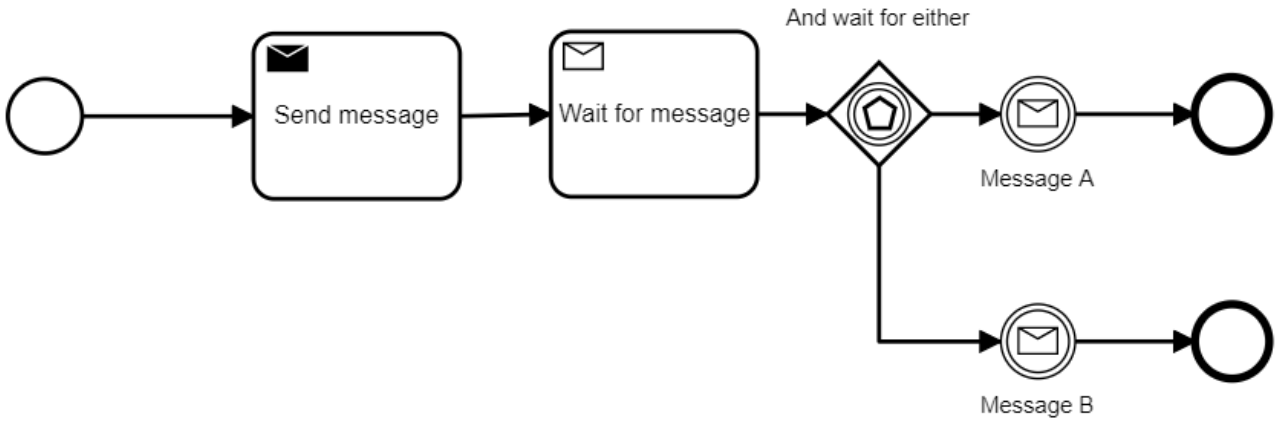
En cuanto a ahora, Zeebe 0.11 (la versión más reciente) admite:

zeebe



Por supuesto, ese es un alcance limitado, y hasta este punto, nos hemos centrado principalmente en el motor de Zeebe, es decir, asegurarnos de que Zeebe tenga la escalabilidad y el rendimiento para manejar casos de uso de alto rendimiento .

Pero hemos estado invirtiendo fuertemente en soporte BPMN para Zeebe este trimestre, y estaremos listos para soportar la correlación de mensajes en el futuro cercano:



A medida que preparamos a Zeebe para la producción en 2018, planeamos agregar soporte para más símbolos como:

- Temporizadores,
- Ámbitos (subprocesos) y
- Ejecución paralela

En 2019, ampliaremos la compatibilidad de símbolos en función de los comentarios de los usuarios y de lo que sabemos sobre los casos de uso que abordará Zeebe.

zeebe

hecho de que los modelos pueden definirse gráficamente y luego ejecutarse directamente por un motor.

Así que volveremos pronto con la parte 2, donde veremos los muchos beneficios de los modelos gráficos en BPMN, particularmente cuando se compara con otros lenguajes de flujo creados para el caso de uso de orquestación. También abordaremos las preocupaciones de los desarrolladores que han tenido experiencias negativas con los modelos gráficos.

Si tiene alguna pregunta o comentario sobre esta publicación, nos encantaría saber de usted. Encuéntranos en Twitter (@ZeebeHQ) o en el foro de Zeebe y la comunidad de Slack .

zeebe

Camunda