

La próxima gran novedad en IA: SYSTEM ONE

1 oct 2026 · 4 min de lectura · Por [Viktor Zaika](#)

Jev y Liquid d1 devuelven opciones y probabilidades en vez de texto generado. Por qué importan los modelos de decisión en los productos de IA.

Sistema Uno

Jev

Modelos de decisión

Arquitectura AI

Líquido d1



Vale, esto sí que es interesante.

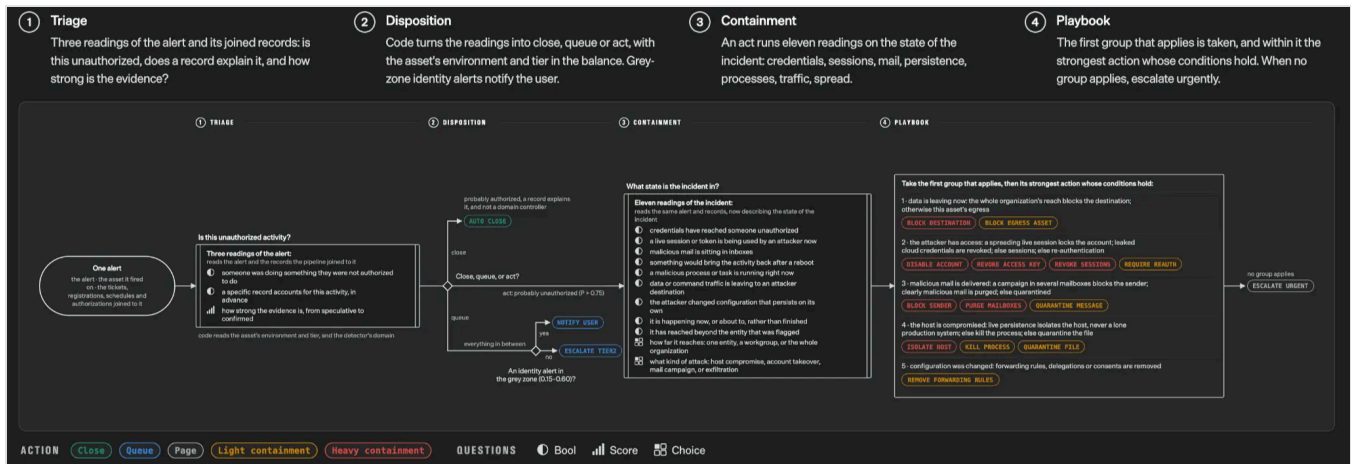
Durante los últimos años casi todos los productos de IA fueron contruidos alrededor de LLMs. Usted envía algo, modelo genera algo de vuelta, tal vez le da JSON, luego su software utiliza lo que salió de él. Lo que funciona perfectamente bien para muchas cosas, pero también significa que a veces llamamos a algún modelo masivo sólo para responder a una pregunta ridículamente pequeña como si algo es sospechoso o exactamente a dónde debería ir esta petición.

Ahora está Jev, creado por TypeSafe, y ellos llaman a esta nueva clase System One Models. Jev básicamente no habla con usted. Usted le da algún estado y le pide que tome una determinada decisión, y devuelve la elección junto con probabilidades / confianza. Imaginen algo más cercano a una función de decisión extremadamente inteligente en lugar de otro chatbot. [Introducción de TypeSafe](#)

Mi primera reacción fue honestamente un poco "bien, ¿no es esto básicamente un clasificador?". Y bien, en cierto sentido sí. Pero entonces empecé a pensar en cuántas llamadas LLM estúpidamente costosas en productos reales son básicamente clasificadores con un traje de \$20.

Digamos que tienes un poco de SaaS donde cada solicitud entrante va actualmente a un modelo de frontera. Algunas peticiones son triviales. Algunas deben ir a un modelo más barato. Algunas requieren revisión humana. Algunas pueden ser sospechosas. Jev puede tomar esta primera decisión muy rápido y luego su modelo realmente caro sólo se despierta cuando realmente lo necesita.

request → Jev → decision → whatever actually needs to happen



Ejemplo de decisiones de TypeSafe dentro de un flujo de trabajo de software más grande. Este es un diagrama de proveedor, no una prueba independiente. [Ver el original y su explicación](#)

Hay un ejemplo de Reddit que me gustó exactamente porque es tan simple. Tenían un gran contexto que va en un LLM básicamente para averiguar si había algo de preocupación. Después de mover este primer chequeo a Jev informaron de ir de 3 a 6 segundos a milisegundos, mientras que el LLM más grande sólo se llama si Jev encuentra algo que vale la pena comprobar más. Ese es un informe de un usuario, no un punto de referencia controlado. [Leer la discusión](#)

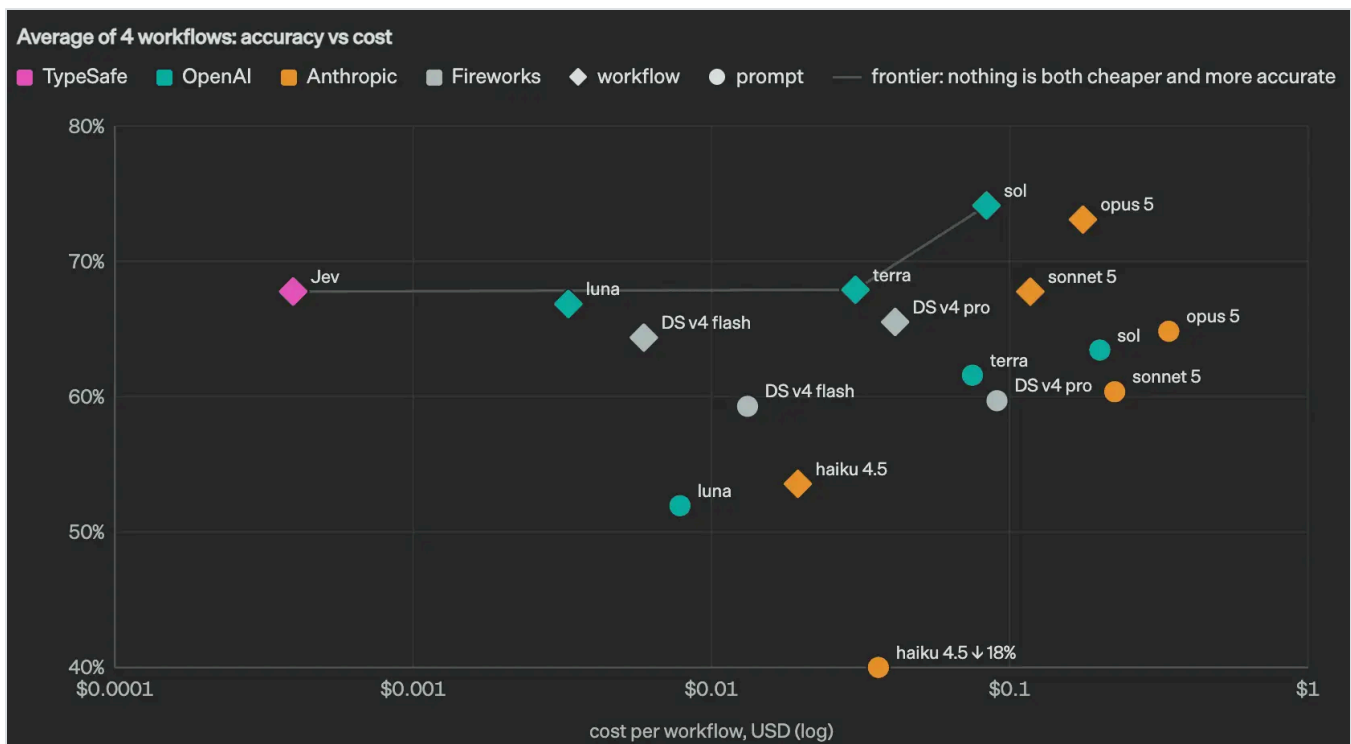
Y honestamente, para la gente del producto esto podría ser más interesante que algún nuevo modelo que anote 2% más alto en otro punto de referencia. Producto puede convertirse en más rápido. Su factura de inferencia puede volverse más pequeña. También puede tener estas pequeñas decisiones en todas partes sin sentir que está enviando cada cosa pequeña a Godzilla.

Usted no necesariamente necesita el modelo gigante más inteligente para cada pequeña decisión en el producto. A veces lo que usted necesita es básicamente una declaración muy inteligente si.

Ahora también está Liquid d1, creado para este tipo de tareas de decisión. Así que al menos empezamos a ver más de una implementación de la idea, y me da curiosidad saber hacia dónde va toda esta categoría. Su documentación muestra la misma estructura básica: entra un estado y sale una respuesta Choice, Score o una respuesta de sí/no llamada Noul. [Lea la documentación de Liquid](#)

Los LLM siguen teniendo todo el sentido del mundo cuando necesito generar, explicar, transformar o resumir algo. Pero si en alguna parte de mi arquitectura hay una llamada a un LLM cuyo único resultado útil es literalmente true, «sales» o 0.83, empezaría a preguntarme si esa llamada debería seguir ahí. Un

artículo de la comunidad de Hugging Face sobre Jev plantea básicamente esta distinción entre respuestas acotadas y decisiones dentro del software. [Leer el artículo de la comunidad](#)



Evaluación de TypeSafe de precisión contra costo, a través de cuatro flujos de trabajo seleccionados. Útil para inspeccionar, pero no para validación independiente. [Lea la metodología y las advertencias](#)

Ahora obviamente no reconstruyas toda la compañía alrededor de este mañana. Jev es nuevo, muchos de los números de rendimiento impresionante vienen de TypeSafe ellos mismos, y el hilo de Reddit tiene gente diciendo que esta es una clasificación glorificada y / o fuertemente sobre-hiper. Necesitamos pruebas más independientes antes de declarar una era completamente nueva de la arquitectura de la IA. [Resultados de TypeSafe](#) · [Discusión de Reddit](#)

Pero si estás creando productos de IA, yo sin duda lo probaría.

Especialmente si usted mira su uso de LLM y de repente se da cuenta de que la mitad de ella existe sólo para tomar pequeñas decisiones.