

A próxima grande novidade em IA: SYSTEM ONE

1 de out. de 2026 · 4 min de leitura · Por [Viktor Zaika](#)

Jev e Liquid d1 retornam escolhas e probabilidades em vez de texto gerado. Por que os modelos de decisão importam para produtos de IA.

Sistema Um

Jev

Modelos de decisão

Arquitetura IA

Líquido d1



Ok, esta é realmente interessante.

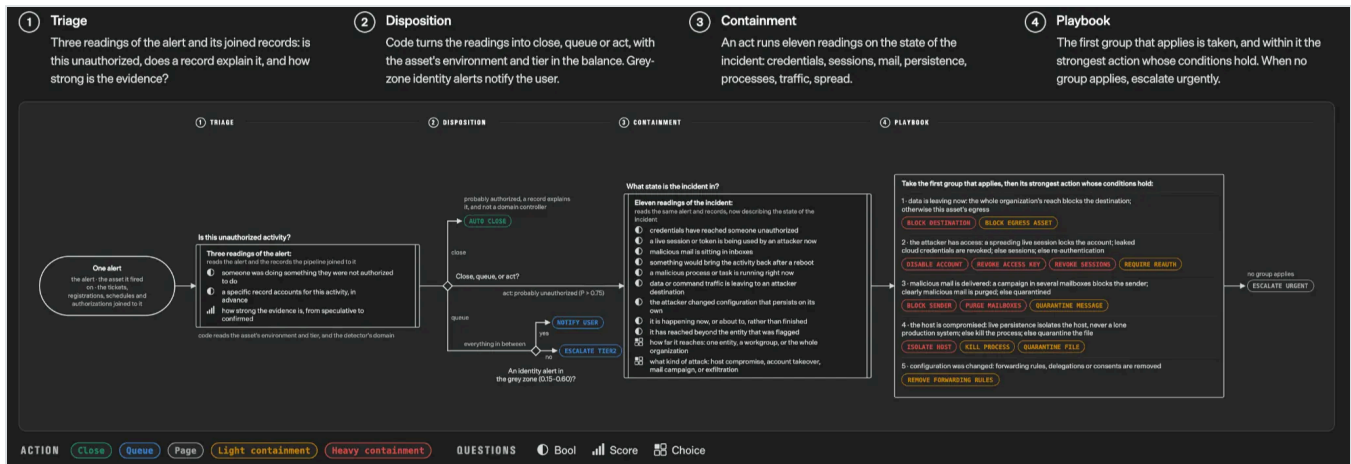
Nos últimos anos, praticamente todos os produtos de IA foram construídos em torno de LLMs. Você envia algo, o modelo gera algo de volta, talvez lhe dê JSON, então seu software usa o que saiu dele. O que funciona perfeitamente bem para muitas coisas, mas também significa que às vezes chamamos algum modelo maciço apenas para responder a uma pergunta ridículamente pequena como se algo é suspeito ou onde exatamente este pedido deve ir.

Agora existe o Jev, criado pelo TypeSafe, e eles chamam a esta nova classe de Modelos do Sistema Um. O Jev basicamente não fala com você. Você lhe dá algum estado e lhe pede para tomar uma certa decisão, e ele retorna a escolha juntamente com probabilidades/ confiança. Então imagine algo mais próximo de uma função de decisão extremamente inteligente em vez de outro chatbot. [Introdução do TipoSafetyçes ĩ](#)

Minha primeira reação foi honestamente um pouco "okay", não é basicamente um classificador??Bem, em algum sentido sim. Mas então comecei a pensar em quantas chamadas LLM estúpidamente caras em produtos reais são basicamente classificadores usando um terno de \$20.

Digamos que você tem alguns SaaS onde cada pedido entrante vai atualmente para um modelo de fronteira. Alguns pedidos são triviais. Alguns devem ir para um modelo mais barato. Alguns requerem revisão humana. Alguns podem ser suspeitos. O Jev pode tomar esta primeira decisão muito rápido e então o seu modelo realmente caro só acorda quando você realmente precisa dele.

request → Jev → decision → whatever actually needs to happen



TipoSafe é um exemplo de decisões dentro de um fluxo de trabalho de software maior. Este é um diagrama do fornecedor, não um teste independente. [Veja o original e a sua explicação](#)

Há um exemplo do Reddit que eu gostei exatamente porque é tão simples. Eles tinham um grande contexto entrando em um LLM basicamente para descobrir se havia algo de preocupante. Depois de mover esta primeira verificação para o Jev, eles relataram ir de 3 a 6 segundos para milissegundos, enquanto o LLM maior só é chamado se o Jev encontrar algo que valha a pena verificar mais. Esse é um relatório do usuário, não um benchmark controlado. [Leia a discussão](#) ¶

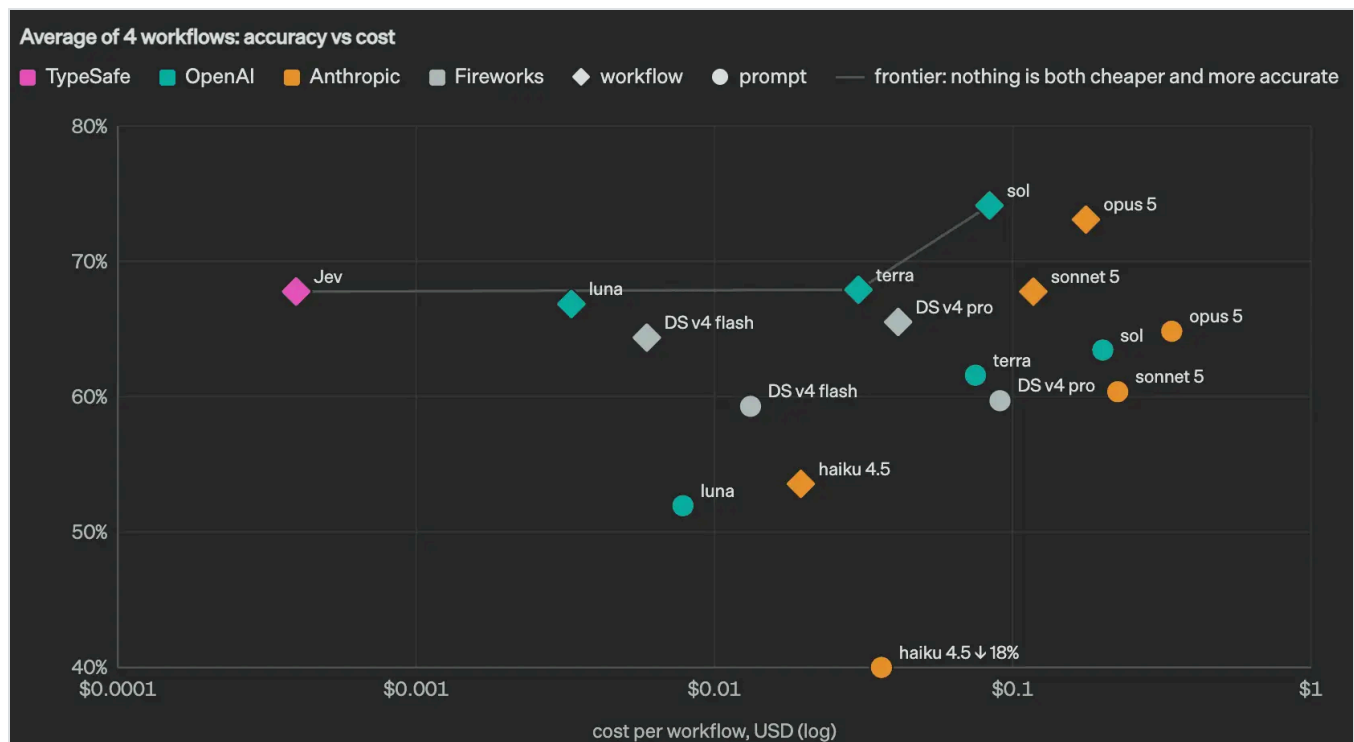
E honestamente, para as pessoas do produto, isso pode ser mais interessante do que alguns novos modelos com 2% maior em outro ponto de referência. O produto pode se tornar mais rápido. Sua conta de inferência pode ser menor. Você também pode ter essas pequenas decisões em todo lugar sem se sentir como se estivesse enviando todas as pequenas coisas para o Godzilla.

Você não precisa necessariamente do modelo gigante mais inteligente para cada pequena decisão no produto. Às vezes, o que você precisa é basicamente uma instrução muito inteligente se.

Agora também existe o Liquid d1, feito para esse tipo de tarefa de decisão. Então já começamos a ver mais de uma implementação da ideia, e estou curioso para saber aonde essa categoria vai chegar. A documentação mostra a mesma estrutura básica: entra um estado; sai uma resposta Choice, Score ou Noul, de sim ou não. [Leia a documentação do Liquid d1](#) ¶

Os LLMs ainda fazem todo o sentido quando preciso gerar, explicar, transformar ou resumir alguma coisa. Mas, se em algum ponto da minha arquitetura há uma chamada a um LLM cujo único resultado útil é literalmente true, "sales" ou 0.83, eu começaria a questionar se ela deveria continuar ali. Um artigo

da comunidade do Hugging Face sobre Jev faz basicamente essa distinção entre respostas limitadas e decisões dentro do software. [Leia o artigo da comunidade](#) ã



Avaliação de precisão- versus- custo do próprio TypeSafe, em quatro fluxos de trabalho selecionados. É útil inspecionar, mas não validação independente. [Leia a metodologia e as advertências](#)

Agora obviamente não reconstrua toda a empresa em torno deste amanhã. O Jev é novo, muitos dos números de desempenho impressionantes vêm do próprio TypeSafe, e o fio Reddit tem pessoas dizendo que isso é classificado glorificado e/ ou fortemente sobrehiped. Bastante justo. Precisamos de testes mais independentes antes de declarar uma era completamente nova da arquitetura da IA. [Resultados do tipo Segurança](#) ã · [Discussão com Reddit](#)

Mas, se você está criando produtos de IA, eu definitivamente testaria.

Especialmente se você olhar para o seu uso de LLM e de repente perceber que metade dele existe apenas para tomar decisões minúsculas.