

Aqui estão os assuntos em destaque nessa edição::

The microservices style of architecture highlights rising abstractions in the developer world because of containerization and the emphasis on low coupling, offering a high level of operational isolation. Developers can think of a container as a self-contained process and the PaaS as the common deployment target, using the microservices architecture as the common style. Decoupling the architecture allows the the same for teams, cutting down on coordination cost among silos. Its attractiveness to both developers and DevOps has made this the de facto standard for new development in many organizations.

Temos visto realidade aumentada e realidade virtual (RA/RV) gerando interesse comercial, ambas tecnologias que já foram relegadas apenas a jogos e inovação. Enquanto perseguir desenhos animados virtuais trouxe a RA à atenção do público por meio de SDKs móveis, h' dam _____, f _____ _____ ? iz .

Decidimos trazer os
de volta do arquivo para essa edição, ainda
que tenhamos permitido que eles desvanecessem no
passado. O conceito não é novo, mas com a aceitação

múltiplos centros de processamento de dados e regiões na União Europeia. Portanto, recomendamos que as empresas, principalmente as que têm bases de usuários globais, avaliem a viabilidade de um refúgio seguro para os dados de seus usuários

. Depois que escrevemos sobre essa técnica no último Radar, lançamos um novo sistema interno que lida com informações sensíveis de todas as pessoas funcionárias da empresa, e optamos por hospedá-lo em um centro de dados localizado na União Europeia.

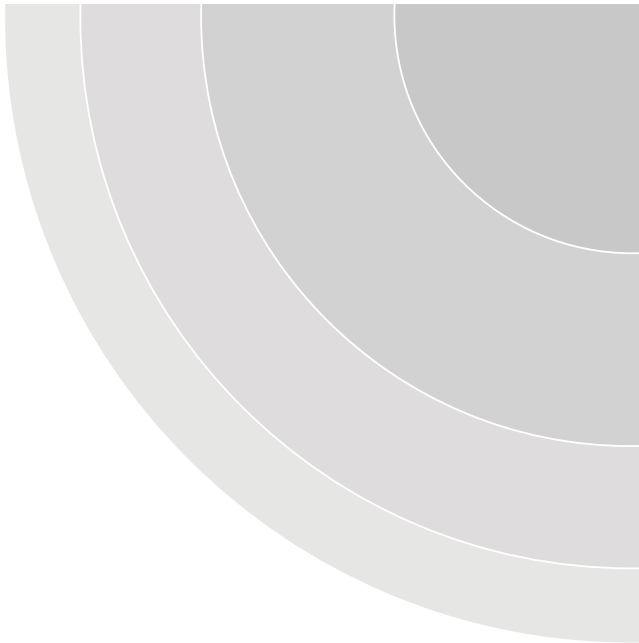
Embora muita documentação possa ser substituída por

Temos visto benefícios significativos com a introdução de arquiteturas de microsserviços, o que permitiu aos times escalar a entrega de serviços implantados e mantidos de maneira independente. Entretanto, os times têm frequentemente encontrado dificuldades para evitar a criação de monólitos de front-end—grandes aplicações de navegador que estão se alastrando e que são difíceis de manter e evoluir, assim como as aplicações monolíticas do lado do servidor que abandonamos. Temos visto emergir uma abordagem que nossos times chamam de arquitetura de front-ends distribuídos. Nessa abordagem, uma aplicação web é dividida por suas páginas e funcionalidades, com cada funcionalidade sendo assumida de ponta a ponta por um único time. Existem múltiplas técnicas para unir as funcionalidades da aplicação—algumas velhas e algumas novas—em uma experiência de usuário coesa, mas o objetivo continua sendo permitir que cada funcionalidade seja desenvolvida, testada e implantada de forma independente das demais. A abordagem BFF - back-end para front-ends funciona bem aqui, com cada time desenvolvendo um BFF para dar suporte a seu conjunto de funcionalidades da aplicação.

Com a crescente popularidade do padrão BFF - back-end para front-ends e do uso de frameworks de conexão de dados em sentido único como React.js, notamos uma reação negativa às arquiteturas REST. As pessoas que criticam acusam o REST de provocar interações ineficientes e verbosas entre sistemas e de não se adaptar à medida que as necessidades do cliente evoluem. Elas oferecem frameworks como GraphQL ou Falcor como mecanismos de busca de dados alternativos que permitem ao cliente especificar o formato dos dados retornados. Mas em nossa experiência, o REST não é a causa desses problemas. Ao contrário disso, eles acontecem quando se deixa de modelar adequadamente o domínio como um conjunto de recursos. Desenvolver ingenuamente serviços que simplesmente expõem modelos de dados hierárquicos estáticos por meio de URLs com templates resultam em uma implementação de APIs REST. Em um domínio ricamente modelado, o REST deve habilitar mais que uma simples busca repetitiva de dados. Em uma arquitetura RESTful completamente evoluída, eventos de negócio e conceitos abstratos também são modelados como recursos, e a implementação deve fazer uso efetivo de hipertexto, relações de link e tipos de mídia para maximizar o desacoplamento entre serviços. Esse antipadrão tem relação direta com o padrão do Modelo de Domínio Anêmico e resulta em serviços com baixas classificações no Modelo de Maturidade de Richardson. Temos mais sugestões para a concepção de APIs REST efetivas nesse artigo do Insights.

Continuamos vendo organizações perseguindo tecnologias “da moda”, assumindo complexidade e risco desnecessários quando uma escolha mais simples funcionaria melhor. Um tema específico é o uso de sistemas distribuídos de Big Data para conjuntos relativamente pequenos de dados. Esse comportamento nos leva a trazer a

Segurança de transporte HTTP estrito é atualmente uma política amplamente suportada, que permite que websites se protejam de ataques downgrade. Um ataque downgrade, no contexto do HTTPS, é aquele que pode direcionar usuários do seu site para utilizar HTTP ao invés de HTTPS, permitindo outros ataques, como ataques man-in-the-middle. Com HSTS, o servidor envia um cabeçalho informando ao navegador que ele deve usar apenas HTTPS para acessar o site. O suporte dos navegadores é hoje difundido o suficiente para que essa funcionalidade simples de implementar seja adicionada por qualquer site usando HTTPS. O Observatory do Mozilla pode ajudar a identificar esse e outros cabeçalhos e opções de configuração úteis para melhorar a segurança e a privacidade. Ao implementar



continuação

código de autenticação. O gerenciamento de identidade terceirizado acelera a entrega, reduz erros e tende a possibilitar uma resposta mais rápida a vulnerabilidades recém descobertas. _____ particularmente tem nos impressionado nesse campo por sua facilidade de integração, variedade de protocolos e conectores suportados e API rica de gerenciamento.

Nossos times continuam gostando de usar _____ e estão começando a usar o serviço para experimentar com arquiteturas sem servidor, combinando Lambda com o API Gateway. Recomendamos que as funções Lambda contenham apenas uma quantidade moderada de código. Assegurar a qualidade de uma solução baseada em um emaranhado de muitas funções Lambda é difícil, e tal solução pode não ser um bom custo-benefício. Para necessidades mais complexas, implantações baseadas em contêineres ou VMs ainda são preferíveis. Além disso, encontramos problemas significativos usando Java para funções Lambda, com latências irregulares de vários segundos quando o

grupos de forma independente. As checagens de saúde realizadas pelos balanceadores de carga estão bastante

continuação

um primeiro momento, ele se limita a frases próximas às usadas para treiná-lo, mas com o tempo ele pode começar a identificar o significado de frases diferentes. Embora ainda esteja em versão beta, a acurácia nas primeiras explorações tem sido convincente, e o produto final é uma opção a se observar quanto a formulários de aplicação que podem se beneficiar da interação com o usuário sem usar as mãos—incluindo os campos móvel, IoT, RA, RV e interativo.

_____ é o SDK subjacente para fazer muitos dos head-mounted displays (HMDs) de RV funcionarem com Unity e provavelmente vai continuar a crescer em importância. Boa parte do trabalho de RV na ThoughtWorks foi construído com base no OpenVR, porque ele pode ser executado em qualquer HDM, ao contrário de outros SDKs. Embora não tenha código aberto, é gratuito por meio de sua licença. O SDK do Oculus é mais restritivo em seu processo de licenciamento e só funciona em dispositivos Oculus. OSVR, embora tenha verdadeiramente código aberto, não parece ter grande adoção ainda. Se você for desenvolver uma aplicação de RV para o maior número de dispositivos possível—e sem usar Unity ou Unreal para desenvolvê-la—OpenVR é a solução mais concreta e pragmática no momento.

_____ é uma solução NoSQL de código aberto que combina base de dados e cache e fornece APIs para escrever lógica de aplicação em Lua. Tanto mecanismos em memória quanto baseados em disco são suportados, e os usuários podem criar múltiplos índices (HASH, TREE, RTREE, BITSET) baseados em seus casos de uso. Os dados em si são armazenados em formato `MessagePack` e usam o mesmo protocolo para comunicação entre clientes e servidor. Tarantool suporta logs write-ahead, transações e replicação master-master assíncrona. Estamos felizes com a decisão arquitetural de adotar a política single-writer e multitarefa cooperativa para lidar com conexões simultâneas.

A excitação em torno da inteligência de máquina atingiu um pico, mas, assim como no caso de Big Data, frameworks e ferramentas úteis estão esperando ser descobertos no meio disso tudo. Uma dessas ferramentas é _____, uma plataforma SaaS que permite a quem desenvolve criar interfaces de conversação usando processamento de linguagem natural (PLN). Wit funciona com entradas tanto de texto quanto

de voz, ajuda quem desenvolve a gerenciar intenção de conversação e permite que lógica de negócios personalizada seja implementada usando JavaScript. O sistema é gratuito para uso comercial e não-comercial e incentiva a criação de aplicações abertas. Esteja ciente de que você precisa aceitar que o Wit use seus dados para melhorar o serviço e para análise própria, então leia os [termos e condições](#) com cuidado. Outro candidato nesse campo é o [Microsoft Bot Framework](#), mas ele está disponível apenas em forma de preview limitada até o momento dessa publicação. Assim como a maioria das coisas da Microsoft, esperamos que o Bot Framework evolua com rapidez, então vale a pena observar de perto.

Temos visto muitas organizações encontrarem problemas na tentativa de usar seu _____ para entrega de aplicações digitais grandes e complexas. Isso é com frequência motivado pela esperança alimentada por fornecedores de ignorar as organizações de TI que não respondem e permitir que o negócio arraste e solte alterações diretamente em produção. Ainda que apoiemos firmemente o fornecimento de ferramentas e fluxos de trabalho certos para produtores de conteúdo, para aplicações com lógica de negócio complexa tendemos a recomendar tratar seu CMS como um componente da sua plataforma (muitas vezes em modo híbrido ou headless) cooperando de forma limpa com outros serviços, ao invés de tentar implementar toda a sua funcionalidade no CMS em si.

Uma das nossas reclamações comuns é sobre colocar inteligência de negócio em middleware, o que resulta software de transporte que ambicionam executar uma lógica de aplicação crítica. Fornecedores no competitivo mercado de API Gateways continuam adicionando funcionalidades que diferenciam seus produtos. Isso tem como resultado produtos com _____, cuja funcionalidade—baseada essencialmente em um proxy reverso—incentiva designs que são difíceis de testar e implantar. API Gateways podem ser úteis para lidar com algumas preocupações genéricas – por exemplo, autenticação e limitação de carga – mas quaisquer inteligências de domínio como transformação de dados ou processamento de regras devem residir em aplicações ou serviços nos quais eles possam ser controlados por times de produto trabalhando próximos aos domínios que eles suportam.

_____ se tornou o compilador padrão para escrever JavaScript da próxima geração. Seu ecossistema está realmente decolando, graças a seu sistema de plugins reestruturado. Ele permite a quem desenvolve escrever código ES6 (e até mesmo ES7) que é executado no navegador e no servidor sem sacrificar a compatibilidade com versões anteriores para navegadores antigos, e com pouquíssima configuração. Possui suporte de primeira classe para diferentes sistemas construa-e-teste, o que torna simples a integração com qualquer fluxo de trabalho em uso. É um ótimo software que se tornou o principal motivador da adoção e da inovação do ES6 (e do ES7).

Quando combinam técnicas e estilos de arquitetura modernos, como microsserviços, DevOps e QA em produção, times de desenvolvimento precisam de monitoramento cada vez mais sofisticado. Simplesmente olhar para gráficos de uso do disco e de utilização da CPU não é mais suficiente, e muitos times coletam métricas da aplicação e especificas do negócio

hC&P P0pA0 PP00P@P0A P 0 000

us cmedrnt PevOdoré si oosO ò eto camoleso diÁ

continuação

_____ é nossa ferramenta favorita para automatizar a maioria das atividades entediadas envolvidas na criação, testes, documentação e provisionamento de aplicativos móveis para iOS e Android. Configuração simples, uma variedade de ferramentas e múltiplas pipelines fazem dela um ingrediente chave para entrega contínua para aplicações móveis.

Testar se layout e estilo de websites responsivos estão funcionando como esperado em vários dispositivos diferentes pode ser um processo lento e frequentemente manual. _____ ajuda a amenizar esse problema oferecendo uma linguagem simples, executada em Selenium, que permite que você especifique expectativas para a aparência do seu website em vários tamanhos de tela. Embora Galen sofra com problemas de fragilidade e velocidade típicos de qualquer abordagem de teste de ponta a ponta, encontramos benefícios em ter rápido feedback sobre questões de design.

Ter uma forma de armazenar segredos com segurança tem se tornado uma questão cada vez mais importante para projetos. A velha prática de manter segredos em um arquivo ou variáveis de ambiente está se tornando difícil de se administrar, principalmente em ambientes com múltiplas aplicações e grande número de microsserviços. _____ lida com o problema fornecendo mecanismos para acessar segredos com segurança através de uma interface unificada. A ferramenta nos atendeu bem em muitos projetos, e nossos times gostaram do.180030 do 5n Bbie e rorp Vacom

_____ é uma adaptação do projeto código aberto Android para plataformas x86. O projeto começou hospedando vários patches da comunidade para suporte para x86, mas em seguida criou sua própria base de código para oferecer suporte para diferentes plataformas x86. Observamos economia de tempo significativa com o uso de Android-x86 em nossos servidores de integração contínua no lugar de emuladores para testes herméticos de interface do usuário. Entretanto, para testes específicos de interface do usuário direcionados a uma resolução de dispositivo específica—simulando baixa quantidade de memória, largura de banda e bateria—é melhor ficar com os emuladores.

Nossos times têm tido sucesso com _____, um cliente HTTP baseado em promessas em JavaScript que eles descrevem como “melhor que Fetch”. O projeto tem muitos endossos e atividade no GitHub, e tem nossa aprovação.

Com o aumento do interesse em arquiteturas de fluxo contínuo de dados e os lagos de dados derivados que elas alimentam, temos visto um aumento de confiança em ferramentas que alteram a captura de dados para conectar bancos de dados transacionais a sistemas de processamento de fluxo. _____ é uma adição bem-vinda a esse campo, convertendo alterações no log write-ahead do PostgreSQL em eventos Kafka. Uma desvantagem dessa abordagem, entretanto, é que você se prende a eventos do banco de dados de baixo nível ao invés de eventos de negócios de maior nível, que recomendamos como base para uma arquitetura orientada a eventos.

Um dos debates perpétuos envolvendo desenvolvedores envolve a tipagem em linguagens de programação: Exatamente quanto é a medida certa? Clojure, o Lisp funcional dinamicamente tipado na JVM, adicionou uma nova contribuição nessa discussão que torna as coisas mais confusas. _____ é um novo utilitário

A ThoughtWorks é uma empresa de software e uma comunidade de pessoas apaixonadas e guiadas por propósitos, especialistas em consultoria, entrega e produtos de software. Pensamos de forma disruptiva para entregar tecnologia que atenda aos maiores desafios de clientes, ao mesmo tempo que buscamos revolucionar a indústria de TI e promover mudanças sociais positivas. Criamos ferramentas pioneiras para times de desenvolvimento que aspiram a ser grandiosos.

Nossos produtos ajudam organizações a melhorar constantemente e a entregar software de qualidade para suas necessidades mais críticas. Fundada há 25 anos, a ThoughtWorks cresceu de um pequeno escritório em Chicago para uma empresa de mais de 40 escritórios espalhadas em 40 escritórios e em 14 países: Brasil, Alemanha, Austrália, Brasil, Canadá, China, Espanha, Estados Unidos, Índia, Itália, Reino Unido, Singapura e Turquia.³²⁵front-end JavaScript,

Thought

©