

DA LATÊNCIA D -7 AO REAL -TIME: A CONVERGÊNCIA DE DATA MESH E ARQUITETURAS ORIENTADAS A EVENTOS (EDA) NA ORQUESTRAÇÃO DE ECOSISTEMAS FINANCEIROS ESCALÁVEIS

FROM D -7 LATENCY TO REAL -TIME: THE CONVERGENCE OF DATA MESH AND EVENT -DRIVEN ARCHITECTURES (EDA) IN THE ORCHESTRATION OF SCALABLE FINANCIAL ECOSYSTEMS

Autor: Willian Gouveia de Aguiar

Bacharel em Sistemas de Informação pela Universidade Bandeirante

Product Management – PM3

RESUMO

O presente artigo propõe um modelo arquitetural robusto voltado para grandes corporações financeiras que enfrentam o desafio da migração de Data Warehouses monolíticos e processos em batch para ecossistemas federados baseados no paradigma de Data Mesh . O foco central da análise reside na drástica redução de latência na ingestão e consumo de dados, abordando a transição de janelas de processamento de sete dias (D -7) para o tempo real (Online). Através de uma abordagem analítica e fundamentada na engenharia de software e teoria de sistemas distribuídos, investiga -se como a implementação de Arquiteturas Orientadas a Eventos (EDA) atua como o tecido conectivo necessário para viabilizar a descentralização proposta pelo Data Mesh . O estudo de caso apresenta a engenharia reversa de uma transformação digital real, discutindo os desafios técnicos da consistência eventual, a governança computacional em malhas de dados e a centralização estratégica do consumo de eventos como catalisador da visão Customer 360° . A pesquisa conclui que a integração entre domínios de dados autônomos e fluxos contínuos de informação é a única via sustentável para a competitividade no setor financeiro moderno.

Palavras -chave: Data Mesh. Arquitetura Orientada a Eventos. Sistemas Financeiros. Latência de Dados. Engenharia de Dados. Real -Time.

ABSTRACT

This article proposes a robust architectural model for large financial corporations facing the challenge of migrating from monolithic Data Warehouses and batch processes to federated ecosystems based on the Data Mesh paradigm. The central focus of the analysis lies in the drastic reduction of latency in data ingestion and consumption, addressing the transition from seven -day processing windows (D -7) to real -time (Online). Through an analytical approach grounded in software engineering and distributed systems theory, it investigates how the implementation of Event -Driven Architectures (EDA) acts as the necessary connective tissue to enable the decentralization proposed by Data Mesh. The case study presents the reverse engineering of a real digital transformation, discussing the technical challenges of eventual consistency, computational governance in data meshes, and the strategic centralization of event consumption as a catalyst for the Customer 360° vision. The research concludes that the integration between autonomous data domains and continuous information flows is the only sustainable path for competitiveness in the modern financial sector.

Keywords: Data Mesh. Event -Driven Architecture. Financial Systems. Data Latency. Data Engineering. Real -Time.

1. INTRODUÇÃO

A indústria financeira contemporânea atravessa um momento de inflexão tecnológica sem precedentes, impulsionada pela digitalização acelerada e pela demanda por experiências de usuário hiperpersonalizadas. O modelo tradicional de gestão de dados, historicamente alicerçado em grandes repositórios monolíticos e processamentos em lote (batch processing), demonstra sinais claros de exaustão funcional. A latência inerente a esses sistemas, que frequentemente operam com dados atualizados em janelas de dias ou semanas (D -1, D-7), tornou -se um gargalo inaceitável em um mercado onde a fraude deve ser detectada em milissegundos e a oferta de crédito deve ser instantânea. A incapacidade de processar e reagir a eventos de negócio no momento em que ocorrem não representa apenas uma desvantagem operacional, mas um risco sistêmico à relevância da instituição financeira perante o consumidor moderno.

Neste cenário, a emergência do conceito de Data Mesh , introduzido por Zhamak Dehghani, surge como uma resposta estrutural às limitações dos Data Lakes centralizados, propondo uma mudança de paradigma sociotécnico onde os dados são tratados como produtos e a propriedade é descentralizada para os domínios de negócio. No entanto, a descentralização por si só não resolve o problema da latência; é necessário um mecanismo de transporte e integração que suporte a velocidade e o volume exigidos. É aqui que as Arquiteturas Orientadas a Eventos (EDA - Event -Driven Architecture) desempenham um papel crucial, permitindo o desacoplamento temporal e espacial entre produtores e consumidores de informação, facilitando o fluxo contínuo de dados através da organização.

O presente trabalho busca explorar a convergência dessas duas poderosas correntes arquiteturais — Data Mesh e EDA — aplicando -as especificamente ao contexto de ecossistemas financeiros escaláveis. A justificativa para tal abordagem reside na complexidade intrínseca dos dados bancários, que exigem rigorosa consistência transacional (ACID) em suas origens, mas que demandam flexibilidade e alta disponibilidade para fins analíticos e de engajamento do cliente. A transição de um modelo de latência D -7 para o processamento em tempo real não é apenas uma atualização de infraestrutura, mas uma reengenharia completa dos fluxos de valor da informação dentro da corporação.

Através da análise de um estudo de caso prático, detalharemos os desafios de engenharia enfrentados na redução do ciclo de vida da informação, saindo de um modelo reativo e tardio para um modelo proativo e imediato. Serão discutidos os aspectos de modelagem de dados, a escolha de tecnologias de streaming, e os trade -offs envolvidos no teorema CAP (Consistência, Disponibilidade e Tolerância a Partição) em sistemas distribuídos. O objetivo final é fornecer um referencial teórico e prático para arquitetos de soluções, engenheiros de dados e gestores de tecnologia que buscam navegar a complexidade da modernização de plataformas de dados em ambientes corporativos de alta criticidade.

2. A OBSOLESCÊNCIA DOS MONOLITOS ANALÍTICOS E A CRISE DO ETL

O modelo tradicional de Business Intelligence (BI) e Data Warehousing (DW), consolidado nas décadas passadas por autores como Ralph Kimball e Bill Inmon, baseava -se na premissa de que os dados deveriam ser extraídos de sistemas transacionais, transformados e carregados (ETL) em um repositório central para análise posterior. Em instituições financeiras, esse ciclo frequentemente obedecia a janelas operacionais noturnas ou semanais, resultando em uma latência

de informação que variava de D -1 a D -7 ou mais. Essa arquitetura, embora eficaz para relatórios regulatórios estáticos e fechamentos contábeis mensais, mostra -se inadequada para as dinâmicas de Open Finance e Instant Payments (como o PIX). O gargalo centralizado do time de engenharia de dados, responsável por manter complexos pipelines de ETL, cria uma desconexão entre quem entende o dado (o domínio de negócio) e quem o implementa tecnicamente, gerando dívida técnica e lentidão na entrega de valor.

A manutenção de pipelines monolíticos em batch para alimentar uma visão de "Cliente 360°" em D-7 resulta em uma fotografia obsoleta da realidade do consumidor. Em um cenário financeiro, a situação de crédito, o comportamento de gastos e os riscos de segurança de um cliente mudam minuto a minuto. Tentar personalizar uma oferta ou prevenir uma fraude com base em dados da semana passada é ineficaz. Além disso, o custo computacional de processar grandes volumes de dados de forma recorrente (reprocessamento total ou janelas deslizantes grandes) cresce exponencialmente com o volume de dados, tornando a operação financeira insustentável.

A rigidez dos esquemas de dados (schema -on-write) típicos de DWs tradicionais também impede a evolução rápida dos modelos de dados necessários para suportar novos produtos digitais, criando um cenário de "ossificação" da arquitetura de dados.

A fragilidade dos processos em lote também se manifesta na tolerância a falhas. Em um pipeline sequencial longo que processa dados semanais, uma falha em uma etapa intermediária pode comprometer toda a cadeia de processamento, atrasando a disponibilidade da informação por dias e exigindo intervenções manuais complexas e arriscadas. Em contraste com a resiliência de microserviços modernos, os jobs de ETL monolíticos são pontos únicos de falha que afetam transversalmente a organização. A dependência de janelas de processamento noturno também entra em conflito com a natureza global e 24/7 das operações financeiras atuais, onde não existe mais "horário fora de pico" para a execução de rotinas pesadas sem impacto na performance dos sistemas transacionais de origem.

Do ponto de vista da infraestrutura, a escalabilidade vertical exigida pelos grandes servidores de banco de dados analíticos (MPP - Massively Parallel Processing) apresenta um custo proibitivo e uma elasticidade limitada. À medida que o volume de dados cresce na escala de Petabytes, a estratégia de "mover o dado até o processamento" torna -se inviável devido às limitações de largura de banda de rede e I/O de disco. A arquitetura precisa evoluir para modelos onde o

processamento ocorre próximo ao dado ou onde o dado flui continuamente através de topologias de processamento distribuído, eliminando a necessidade de grandes cargas em lote que saturam a rede corporativa e degradam a performance dos sistemas core banking durante os períodos de extração.

Adicionalmente, a qualidade do dado sofre no modelo de ingestão em batch centralizado. Com os engenheiros de dados responsáveis pelo ETL muitas vezes desconhecendo as regras de negócio profundas dos sistemas de origem, as transformações aplicadas podem introduzir erros semânticos ou perder nuances importantes do contexto original. A correção desses erros é lenta e burocrática, pois envolve múltiplos times e ciclos de desenvolvimento. Isso leva a um fenômeno de desconfiança nos dados do Data Lake ou Data Warehouse , onde os usuários de negócio acabam criando suas próprias planilhas paralelas ("Shadow IT") para obter respostas mais rápidas, minando a governança corporativa e a unicidade da verdade ("Single Source of Truth").

Por fim, a incapacidade de reagir a eventos em tempo real impede a implementação de lógicas de negócio complexas baseadas em gatilhos (triggers). No modelo D -7, o banco só descobre que o cliente teve um problema com o cartão de crédito dias depois, quando a oportunidade de retenção

ou suporte proativo já passou. A arquitetura baseada em batch é inerentemente passiva e retrospectiva, olhando sempre pelo espelho retrovisor, enquanto a gestão moderna de produtos financeiros exige uma postura preditiva e prescritiva, antecipando as necessidades do cliente antes mesmo que elas se manifestem explicitamente. A superação desse modelo não é apenas uma atualização tecnológica, mas uma necessidade de sobrevivência estratégica.

3. DATA MESH: A FEDERAÇÃO DE DOMÍNIOS COMO RESPOSTA À COMPLEXIDADE

O paradigma de Data Mesh surge para resolver as limitações de escalabilidade organizacional e técnica dos modelos centralizados. Fundamentado nos princípios de Domain -Driven Design (DDD) de Eric Evans, o Data Mesh propõe a descentralização da propriedade dos dados, distribuindo -a para os times que estão mais próximos da origem da informação — os domínios.

Em uma instituição financeira, isso significa que os dados de "Cartões", "Investimentos", "Conta Corrente" e "Seguros" são geridos, mantidos e servidos pelos respectivos times de produto, e não por uma área central de dados. Essa mudança transfere a responsabilidade da qualidade e da disponibilidade do dado para quem detém o conhecimento do negócio, eliminando o gargalo do time central de engenharia e permitindo que a organização escale linearmente com o aumento da complexidade.

Um dos pilares fundamentais do Data Mesh é o conceito de "Dados como Produto" (Data as a Product). Isso implica que os domínios não apenas expõem dados brutos, mas oferecem produtos de dados curados, documentados, confiáveis e fáceis de consumir. Para viabilizar a transição de D-7 para Real -Time, esses produtos de dados não podem ser apenas tabelas e estáticas ou arquivos CSV; eles devem ser fluxos de eventos ou APIs de baixa latência. O Data Product deve encapsular a complexidade da transformação e garantir Acordos de Nível de Serviço (SLAs) de disponibilidade e frescor (freshness) da informação. Isso exige que os times de domínio incorporem competências de engenharia de dados, promovendo uma cultura de DevOps e DataOps integrada.

A infraestrutura de autoatendimento (Self-serve Data Infrastructure) é outro componente crítico para o sucesso dessa arquitetura. Para evitar que a descentralização resulte em duplicação de esforços e fragmentação tecnológica, a organização deve prover uma plataforma comum que abstraia a complexidade de provisionamento de recursos de armazenamento, processamento e orquestração. No contexto de redução de latência, essa plataforma deve oferecer capacidades de streaming "as a service" (como Kafka ou Pulsar gerenciados), ferramentas de Change Data Capture (CDC) e frameworks de processamento de fluxo (como Apache Flink ou Spark Streaming), permitindo que os domínios publiquem e consumam eventos em tempo real sem precisarem reinventar a roda da infraestrutura a cada novo projeto.

A governança federada computacional é o mecanismo que mantém a coesão nesse ecossistema distribuído. Diferente da governança centralizada tradicional, baseada em controle e burocracia,

a governança no Data Mesh foca na interoperabilidade e na automação de políticas. Para garantir que os dados de diferentes domínios possam ser correlacionados para formar a visão 360° do cliente, é necessário estabelecer padrões globais de identificação, esquemas de metadados e controle de acesso. A utilização de contratos de dados (Data Contracts) e esquemas evolutivos (como Avro ou Protobuf) garante que mudanças na estrutura dos dados de um domínio produtor não quebrem os processos dos consumidores a jusante, permitindo a evolução independente dos serviços.

A aplicação do Data Mesh em finanças enfrenta desafios específicos relacionados à segurança e conformidade regulatória. A descentralização não pode significar perda de controle sobre dados sensíveis (PII) ou sigilo bancário. A arquitetura deve incorporar políticas de segurança no código ("Policy as Code"), garantindo que o controle de acesso, a criptografia e a auditoria sejam aplicados automaticamente em todos os produtos de dados. Isso permite que a agilidade proporcionada pela malha de dados não comprometa a robustez institucional exigida pelos órgãos reguladores, transformando a conformidade de um obstáculo manual em um atributo de qualidade intrínseco da plataforma.

A mudança cultural exigida pelo Data Mesh é tão profunda quanto a técnica. Requer abandonar a mentalidade de "silos de dados" e adotar uma postura de colaboração e compartilhamento. Os times de produto passam a ser avaliados não apenas pelas funcionalidades de suas aplicações, mas pela qualidade e utilidade dos dados que disponibilizam para o resto da organização. Essa mudança de incentivos é fundamental para viabilizar a arquitetura de tempo real, pois motiva os domínios produtores a emitir eventos de qualidade na fonte, em vez de delegar a limpeza e organização dos dados para processos de ETL posteriores que adicionam latência e reduzem a fidelidade da informação.

4. ARQUITETURAS ORIENTADAS A EVENTOS (EDA) COMO TECIDO CONECTIVO

A Arquitetura Orientada a Eventos (EDA) representa a espinha dorsal tecnológica necessária para operacionalizar a visão de tempo real em um ambiente de Data Mesh. Em contraste com as arquiteturas baseadas em requisição-resposta (REST/HTTP), que são síncronas e criam acoplamento temporal forte entre serviços, a EDA permite que os sistemas se comuniquem de forma assíncrona através da emissão e consumo de fatos imutáveis (eventos). No contexto financeiro, um evento pode ser "Transação Autorizada", "Cliente Atualizou Endereço" ou "Proposta de Crédito Rejeitada". Esses eventos são propagados imediatamente para um log distribuído, tornando-se disponíveis para qualquer consumidor interessado com latência de milissegundos, rompendo definitivamente com o ciclo D-7.

A utilização de logs distribuídos de alta performance, como Apache Kafka, permite tratar os dados como fluxos contínuos e infinitos (unbounded streams), em vez de conjuntos finitos e estáticos. Isso habilita o padrão de arquitetura Kappa, onde a camada de processamento de dados serve tanto para análises em tempo real quanto para o reprocessamento histórico, simplificando a pilha tecnológica ao eliminar a necessidade de manter uma arquitetura Lambda complexa (com caminhos separados para batch e speed). A persistência dos eventos no log permite que novos serviços ou produtos de dados sejam "plugados" na arquitetura e replayem o histórico de eventos para construir seu estado local, garantindo a desacoplamento total entre produtores e novos consumidores.

Para reduzir a latência de dias para tempo real, é imperativo capturar as mudanças de estado diretamente na fonte. O padrão Change Data Capture (CDC) atua lendo os logs de transação dos bancos de dados operacionais (WAL - Write Ahead Log no PostgreSQL, por exemplo) e convertendo cada inserção, atualização ou exclusão em um evento de fluxo. Isso desacopla a geração de eventos da lógica da aplicação, garantindo que qualquer alteração no banco de dados seja capturada com latência mínima e sem impacto de performance no sistema transacional.

Essa técnica é fundamental para migrar sistemas legados (Mainframe/Cobol) para a arquitetura moderna sem a necessidade de refatoração profunda do código antigo.

O processamento de eventos complexos (CEP - Complex Event Processing) permite identificar padrões, tendências e anomalias em janelas de tempo deslizantes enquanto os dados estão em

movimento. Em vez de armazenar o dado para depois consultá-lo (abordagem passiva do DW), o sistema avalia o dado no momento em que ele chega. Isso é crucial para detecção de fraudes, onde o padrão de comportamento anômalo deve ser identificado e bloqueado antes que a transação seja concluída. A capacidade de realizar joins entre diferentes fluxos de dados (ex: fluxo de transações + fluxo de geolocalização) em tempo real enriquece o contexto da informação instantaneamente, algo impossível nos modelos D-7.

A consistência em sistemas distribuídos baseados em eventos segue o modelo de Consistência Eventual. De acordo com o Teorema CAP, em um sistema distribuído sujeito a partições de rede, deve-se escolher entre Disponibilidade e Consistência forte imediata. Em arquiteturas de Data Mesh em tempo real, prioriza-se geralmente a disponibilidade e a tolerância a partição, aceitando que haverá um pequeno delay (na ordem de milissegundos ou segundos) para que todos os nós do sistema converjam para o mesmo estado. O projeto de sistemas financeiros sob essa ótica exige lidar com idempotência (capacidade de processar o mesmo evento múltiplas vezes sem efeitos colaterais duplicados) e ordenação de eventos, garantindo a integridade financeira mesmo em cenários de falha.

A escalabilidade da EDA é horizontal e massiva. Tópicos de eventos podem ser particionados e distribuídos entre múltiplos brokers e consumidores, permitindo o processamento paralelo de milhões de eventos por segundo. Isso resolve o problema de escalabilidade dos DWs monolíticos. À medida que o volume de transações cresce, basta adicionar mais nós ao cluster de streaming e mais instâncias aos grupos de consumidores, mantendo a latência baixa e previsível.

Essa elasticidade é essencial para suportar picos de carga típicos do varejo financeiro (como Black Friday ou datas de pagamento de folha), onde os sistemas tradicionais costumam sofrer degradação de performance.

5. ESTUDO DE CASO: A ENGENHARIA REVERSA DA LATÊNCIA (DE D-7 PARA ONLINE)

Analisando a trajetória de migração de um grande conglomerado financeiro, observou-se que o estado inicial (D-7) era caracterizado por uma malha de transferências de arquivos (MFT) e scripts SQL agendados via Control-M. A consolidação dos dados de clientes envolvia a extração de dados de Mainframes, sistemas de cartões e canais digitais, que eram depositados em uma Staging Area. O processamento de higienização e conformação ocorria semanalmente devido ao volume massivo de dados e à janela restrita de manutenção. Isso resultava em campanhas de marketing ineficientes e análises de risco defasadas, onde um cliente inadimplente na segunda-feira só seria identificado pelo sistema de CRM na segunda-feira seguinte.

O processo de engenharia reversa para atingir o tempo real iniciou-se pela identificação dos eventos de domínio críticos. Em vez de tentar replicar todas as tabelas do banco de dados legado, focou-se nas entidades de negócio: "Cliente", "Conta", "Transação". Foi implementada uma camada de CDC (Change Data Capture) conectada aos logs do DB2 do Mainframe, publicando alterações granulares em tópicos Kafka. Para lidar com a complexidade de consumo desses eventos brutos, criou-se a figura do "Centralizador de Consumo de Eventos" — um artefato arquitetural dentro do domínio de Clientes responsável por ingerir, deduplicar e ordenar os eventos provenientes de múltiplas fontes díspares, construindo uma visão unificada e atualizada do estado do cliente em memória ou em bancos de dados NoSQL de baixa latência (chave-valor).

A redução da latência expôs desafios de qualidade de dados que antes eram mascarados pelo processo de batch. Inconsistências de cadastro, duplicidade de registros e erros de formatação tornaram-se visíveis instantaneamente. Isso forçou a implementação de validação de schema em

tempo de produção (utilizando Schema Registry e Avro) e a adoção de padrões de Dead Letter Queues (DLQ) para segregar eventos inválidos sem interromper o fluxo de processamento. A governança deixou de ser um processo de auditoria post-mortem para se tornar um gatekeeper ativo no pipeline de ingestão, garantindo que apenas dados em conformidade entrassem na malha de dados em tempo real.

O desafio da consistência eventual foi mitigado através de padrões de design como Saga Pattern e Command Query Responsibility Segregation (CQRS). O modelo de leitura (utilizado pelos canais digitais e analistas) foi separado do modelo de escrita, sendo atualizado assincronamente pelos eventos. Para o usuário final, a percepção é de tempo real: ao realizar uma transferência, o saldo é atualizado na tela do aplicativo imediatamente, pois a interface consome da projeção de leitura que é alimentada pelo fluxo de eventos em milissegundos. Nos bastidores, a conciliação contábil robusta pode ocorrer de forma assíncrona, mas a experiência do usuário e a disponibilidade do dado para outros sistemas são imediatas.

A topologia de rede e processamento foi reestruturada para suportar o alto throughput. Substituiu-se o processamento centralizado em grandes clusters Hadoop por microserviços de processamento de fluxo (usando Kafka Streams ou KSQL) distribuídos nas bordas dos domínios. Isso reduziu o tráfego de rede "leste-oeste" (entre servidores internos) de grandes arquivos e passou a privilegiar o tráfego de pequenas mensagens binárias compactadas. O resultado foi uma redução drástica na utilização de banda e armazenamento, pois os dados não precisavam mais ser duplicados em múltiplas camadas de staging e tabelas temporárias, existindo como um log imutável acessível sob demanda.

O impacto no negócio foi mensurável e transformador. A capacidade de disparar ofertas contextualizadas (ex: oferecer seguro viagem no momento em que o cliente compra uma passagem aérea com o cartão) aumentou a conversão de produtos significativamente. A análise de risco de crédito passou a considerar o comportamento transacional dos últimos minutos, e não apenas o histórico do mês anterior, permitindo a aprovação de mais crédito com menor risco. A arquitetura validou a tese de que a centralização da lógica de consumo de eventos, combinada com a descentralização da produção de dados no modelo Mesh, é a chave para a agilidade em escala.

6. GOVERNANÇA, SEGURANÇA E O FUTURO DOS ECOSISTEMAS FEDERADOS

A migração para um ecossistema federado e em tempo real exige uma reinvenção dos modelos de segurança da informação. O controle perimetral tradicional (firewalls) é insuficiente em uma arquitetura de microserviços e malha de dados onde o tráfego é interno e intenso. Adota-se o modelo de Zero Trust, onde cada interação entre produtores e consumidores de eventos deve ser autenticada e autorizada. O uso de Listas de Controle de Acesso (ACLs) granulares nos tópicos do Kafka garante que apenas serviços autorizados possam ler ou escrever em determinados fluxos de dados, prevenindo o vazamento de informações sensíveis dentro da própria infraestrutura corporativa.

A Lei Geral de Proteção de Dados (LGPD) e regulamentações similares impõem desafios adicionais em arquiteturas de logs imutáveis. O "direito ao esquecimento" parece contradizer o princípio de imutabilidade dos eventos. Para resolver isso, utilizam-se técnicas de criptografia de chaves deletáveis (crypto-shredding). Os dados sensíveis dentro do evento são criptografados com uma chave específica do usuário; para "esquecer" o usuário, basta deletar a chave de criptografia, tornando os dados históricos matematicamente inacessíveis, mesmo que os bytes permaneçam no log. Isso demonstra como a arquitetura técnica deve evoluir em simbiose com os requisitos legais.

A catalogação e a linhagem de dados (Data Lineage) tornam -se complexas, porém essenciais, em ambientes distribuídos. É necessário rastrear o fluxo da informação desde a transação original no Mainframe, passando pelos tópicos de eventos, pelas transformações nos microsserviços, até o consumo no dashboard executivo ou no aplicativo móvel. Ferramentas de observabilidade de dados e catálogos automatizados são indispensáveis para mapear essas dependências dinâmicas. Sem isso, a arquitetura de Mesh pode degenerar em um caos inadministrável, onde ninguém sabe a origem ou a confiabilidade de um determinado dado.

A evolução dos padrões de metadados é o que permite a interoperabilidade sem acoplamento. A adoção de ontologias de negócios comuns e contratos de dados semânticos facilita o entendimento mútuo entre domínios distintos. Quando o domínio de "Cartões" publica a um evento de "Compra", o domínio de "Pontos de Fidelidade" deve ser capaz de interpretar esse

evento inequivocamente. A governança foca, portanto, na definição desses vocabulários controlados e na garantia de que os contratos de interface sejam respeitados, atuando como um "urbanista" da cidade de dados, definindo as zonas e as vias, mas deixando a construção dos prédios (produtos de dados) para os arquitetos locais (times de domínio).

O futuro dessas arquiteturas aponta para uma maior convergência entre os mundos operacional e analítico, um conceito por vezes chamado de Translytical ou HTAP (Hybrid Transactional/Analytical Processing). A latência zero não é apenas sobre velocidade, mas sobre a eliminação da barreira entre "fazer o negócio" e "analisar o negócio". A inteligência artificial e os modelos de Machine Learning serão implantados diretamente no fluxo de eventos (In-stream Analytics), tomando decisões autônomas em tempo real, sem intervenção humana, fechando o ciclo de feedback instantaneamente.

Por fim, a sustentabilidade dessas arquiteturas depende da contínua capacitação das equipes. A complexidade de sistemas distribuídos, consistência eventual e processamento de streams exige um perfil profissional com forte base em ciência da computação e sistemas de informação. A tecnologia não é mágica; ela demanda rigor de engenharia. A transição do D -7 para o Real-Time é, em última análise, uma jornada de amadurecimento técnico da organização, movendo -se de processos manuais e frágeis para automação robusta e resiliente, habilitada por uma arquitetura que reflete a natureza dinâmica e interconectada do mercado financeiro moderno.

7. CONCLUSÃO

A investigação realizada ao longo deste artigo demonstra que a convergência entre o paradigma de Data Mesh e as Arquiteturas Orientadas a Eventos (EDA) não constitui apenas uma tendência tecnológica passageira, mas sim uma evolução estrutural imperativa para instituições financeiras que almejam liderança na era digital. A análise da obsolescência dos modelos de processamento em batch e dos Data Warehouses monolíticos revelou que a latência de D -7 (ou mesmo D -1) é incompatível com as expectativas de imediato do consumidor moderno e com a necessidade de gestão de riscos em tempo real. A incapacidade de processar dados na velocidade dos negócios resulta em perda de competitividade, ineficiência operacional e cegueira estratégica.

O modelo arquitetural proposto, validado pelo estudo de caso de engenharia reversa, evidenciou que a redução da latência para o patamar de Real-Time exige mais do que a simples adoção de ferramentas de streaming como Kafka. Exige uma reestruturação da topologia organizacional e dos fluxos de dados, descentralizando a responsabilidade para os domínios de negócio enquanto se centraliza a governança da infraestrutura e dos padrões de interoperabilidade. A figura do "centralizador de consumo de eventos" provou -se um padrão eficaz para reconciliar a natureza

distribuída da geração de dados com a necessidade de visões unificadas e coerentes do cliente (Customer 360°), superando os desafios teóricos do teorema CAP através de design de software inteligente.

Observou -se também que a consistência eventual, muitas vezes temida em ambientes financeiros conservadores, é um trade -off gerenciável e necessário para alcançar alta disponibilidade e escalabilidade global. Através de técnicas como Sourcing de Eventos, CQRS e validação de esquemas rigorosa, é possível manter a integridade dos dados financeiros enquanto se desfruta da agilidade dos sistemas desacoplados. A governança computacional federada surge, então, como o fiel da balança, garantindo que a autonomia dos times não resulte em anarquia de dados, mas sim em um ecossistema harmonioso de produtos de dados interoperáveis.

A transição de uma cultura de processos em lote para uma cultura orientada a eventos representa uma mudança profunda na forma como a organização "pensa". Deixa -se de olhar para o passado (relatórios estáticos) para reagir ao presente e prever o futuro (análises preditivas em fluxo). Essa mudança habilita novos modelos de negócio baseados em hiperpersonalização, detecção proativa de fraudes e serviços financeiros contextualizados, gerando valor tangível tanto para a instituição quanto para o cliente final. O dado deixa de ser um subproduto do processo para se tornar o ativo central da estratégia.

Conclui -se, portanto, que a arquitetura aqui discutida fornece a base técnica para a próxima geração de serviços bancários. A escalabilidade horizontal, a resiliência a falhas e a flexibilidade evolutiva proporcionadas pela união de Mesh e EDA preparam as corporações para um futuro onde o volume de dados continuará a crescer exponencialmente e a latência aceitável tenderá a zero. A conquista técnica de reduzir a disponibilidade da informação de sete dias para tempo real é, acima de tudo, uma conquista de negócio, libertando a organização das amarras dos ciclos de processamento legados e permitindo que ela opere na velocidade da vida digital.

Com tudo isso, recomenda -se que trabalhos futuros explorem a integração de malhas de dados com tecnologias emergentes de Inteligência Artificial Generativa e Data Fabric , investigando como a automação semântica pode reduzir ainda mais o atrito na integração entre domínios. A jornada da modernização de dados é contínua, e o domínio dos fundamentos de sistemas distribuídos e topologias de rede, conforme discutido sob a ótica dos Sistemas de Informação, permanecerá como o diferencial crítico para os profissionais que desenharão o futuro do setor financeiro Global.

REFERÊNCIAS

BREWER, Eric. CAP twelve years later: How the "rules" have changed. *Computer* , v. 45, n. 2, p. 23 -29, 2012.

DEHGHANI, Zhamak. *Data Mesh: Delivering Data -Driven Value at Scale* . Sebastopol: O'Reilly Media, 2020.

EVANS, Eric. *Domain -Driven Design: Tackling Complexity in the Heart of Software* . Boston: Addison -Wesley Professional, 2003.

HOHPE, Gregor; WOOLF, Bobby. Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions . Boston: Addison -Wesley Professional, 2003.

INMON, William H. Building the Data Warehouse . 4. ed. Indianapolis: Wiley, 2005.

KIMBALL, Ralph; ROSS, Margy. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling . 3. ed. Indianapolis: Wiley, 2013.

KLEPPMANN, Martin. Designing Data -Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems . Sebastopol: O'Reilly Media, 2017.

KREPS, Jay. I Heart Logs: Event Data, Stream Processing, and Data Integration . Sebastopol: O'Reilly Media, 2014.

NEWMAN, Sam. Building Microservices: Designing Fine -Grained Systems . 2. ed. Sebastopol: O'Reilly Media, 2021.

STOPFORD, Ben. Designing Event -Driven Systems: Concepts and Patterns for Streaming Services with Apache Kafka . Sebastopol: O'Reilly Media, 2018.