Data Mesh

Em 2019, no artigo How to Move Beyond a Monolithic Data Lake to a Distributed Data Mesh, Zhamak Dehghani propõe uma nova abordagem para construção de plataformas de dados, chamado Data Mesh. De acordo com Dehghani (2019), os modelos atuais de plataforma de dados, sendo estes baseados em *Data Lakes* ou *Data Warehouses*, são centralizados, monolíticos e agnósticos a domínios (áreas de negócio). Embora inúmeras organizações tenham investido fortemente na construção de soluções que facilitem a tomada de decisão orientada por dados, como plataformas de dados e inteligência, estas organizações têm considerado os resultados medíocres (DEHGHANI, 2019).

Dehghani (2019) reconhece a necessidade de usuários de dados, como cientistas de dados e analistas, terem acesso simplificado a um conjunto diversificado de dados, bem como a necessidade de separar o uso de dados de sistemas operacionais dos dados consumidos para fins analíticos. Mas propõe que os modelos de solução centralizados existentes não são a resposta ideal para grandes empresas com múltiplos domínios (áreas de negócio), onde novas fontes de dados são continuamente adicionadas.

Dehghani (2019) complementa que não defende a fragmentação em silos de conjunto de dados orientados a domínios (áreas de negócio), ocultos sob algum sistema de informação e geralmente difíceis de descobrir, entender e consumir. Tão pouco a existência de vários *Data Warehouses* (ou *Data Lakes*) fragmentados em decorrência de anos de dívidas técnicas acumuladas. Dehghani (2019) argumenta que a resposta a esses silos acidentais de dados inacessíveis não é criar uma plataforma de dados centralizada, com uma equipe centralizada que possui e faz a curadoria dos dados de todos os domínios (áreas de negócio), pois este tipo de solução não é escalável.

Dehghani (2019) define as gerações de plataformas de dados da seguinte forma:

- **Primeira geração:** *Data Warehouses* proprietários e plataformas de *Business Intelligence* e alto custo, que deixaram as organizações com um elevado débito técnico (diversos processos de ETL insustentáveis, tabelas e relatórios que apenas um pequeno grupo de profissionais especializados conseguiam entender).
- **Segunda geração:** Um ecossistema de *big data* com *Data Lake* como solução para todos os problemas. Soluções complexas de *big data* e *jobs* em lote de longa duração, geridos por uma equipe central de engenheiros de dados super especializados que criam Data Lakes monstruosos, que, na melhor das hipóteses, permitem algumas poucas análises de P&D (Pesquisa e Desenvolvimento); com mais promessas que realizações.
- **Terceira geração:** Geração atual de plataforma de dados, similar à geração anterior, porém com um toque moderno de:
    - *streaming* para disponibilização de dados em tempo real com arquiteturas como o Kappa;
    - unificação de processos em lote com *streaming* de processamento de dados através de *frameworks* como o Apache Beam;

- utilização extensa de serviços de gerenciamento em nuvem para armazenamento de dados, motores de execução de *pipelines* de dados e plataformas de aprendizado de máquina.

Embora a última geração de plataforma de dados tenha abordado algumas lacunas das gerações anteriores, como a análise de dados em tempo real e redução do custo de gerenciamento da infraestrutura de *big data*, esta ainda sofre de muitas das características subjacentes que levaram aos fracassos das gerações anteriores (DEHGHANI, 2019).

Falhas das arquiteturas atuais

De acordo com Dehghani (2019), as arquiteturas atuais possuem os seguintes problemas:

- **Centralizada e monolítica:**
  - De modo geral, uma arquitetura de dados possui os os objetivos de: (a) coletar dados de inúmeras fontes; (b) limpar, enriquecer e transformar os dados coletados; e (c) servir conjuntos de dados para inúmeros consumidores, com diversas necessidades.
  - As arquiteturas de dados atuais armazenam dados que logicamente pertencem a diversos domínios, como por exemplo: “alunos”, "servidores", “almoxarifado”, “compras”, etc.
  - Enquanto na última década tenhamos aplicado de forma bem sucedida o conceito de design orientado a domínios em nossas aplicações (com a utilização de microsserviços), desconsideramos amplamente os conceitos de domínio em uma plataforma de dados. Criamos o maior monolítico de todos, a plataforma de *big data*. Enquanto esta abordagem possa funcionar para organizações com poucos domínios e pouca diversidade de casos de uso para os dados, o mesmo não acontece para organizações com inúmeros domínios, inúmeras fontes de dados e grande diversidade de consumidores de dados.
  - Existem dois pontos críticos em uma arquitetura de dados centralizada:
    - Dados ubíquos e proliferação de fontes: à medida que mais dados tornam-se onipresentes, a capacidade de consumir tudo e harmonizá-los em um só lugar diminui. A velocidade de resposta à proliferação de novas fontes de dados tende a diminuir, uma vez que precisaremos incluí-las todas em uma mesma plataforma centralizada.
    - Agenda contínua de inovações e proliferação de consumidores de dados: cada vez mais, as organizações precisam disponibilizar uma grande quantidade de casos de uso de consumo de dados em suas plataformas, o que leva a uma crescente necessidade de novas transformações, agregações e projeções de dados para satisfazer o ciclo de teste e aprendizado envolvido no processo de inovação. O longo tempo de resposta para satisfazer as necessidades de consumidores de dados tem sido historicamente um ponto de atrito organizacional, e continua sendo nas arquiteturas modernas de

plataformas de dados.

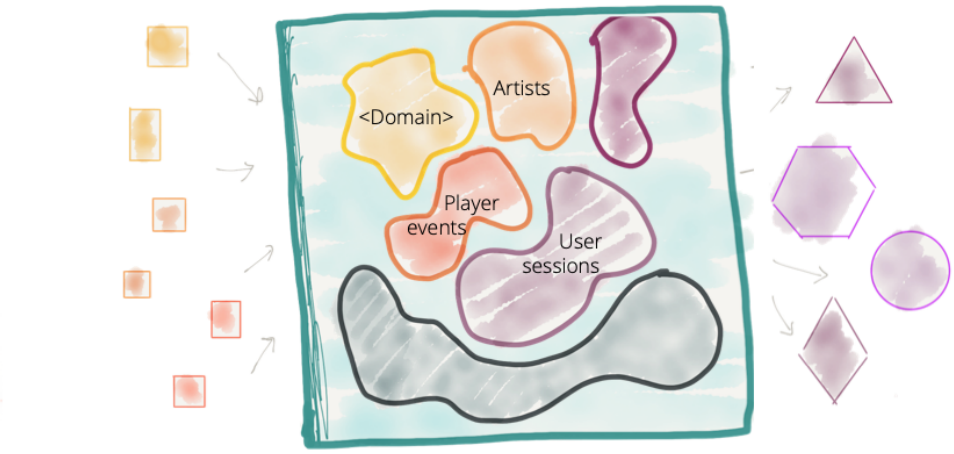

Figura 1: Plataforma de dados centralizada sem a clara definição de fronteiras entre dados de domínios distintos. Fonte: (DEHGHANI, 2019).

- **Pipelines acoplados**
  - Como já mencionado, a inclusão ou alteração de fontes de dados, bem como a mudança de requisitos dos clientes faz com que a plataforma de dados aumente de tamanho.
  - O modelo atual, que de forma resumida podemos sumarizar como: inclusão, processamento e disponibilização de dados, possui estágios fortemente acoplados. E embora seja possível um certo nível de escalabilidade, através da alocação de equipes à entregas específicas, este acoplamento excessivo entre os estágios limita a independência de entregas de novos valores e funcionalidades. Este acoplamento excessivo limita a velocidade de resposta a novos requisitos e/ou inclusão de novas fontes de dados.

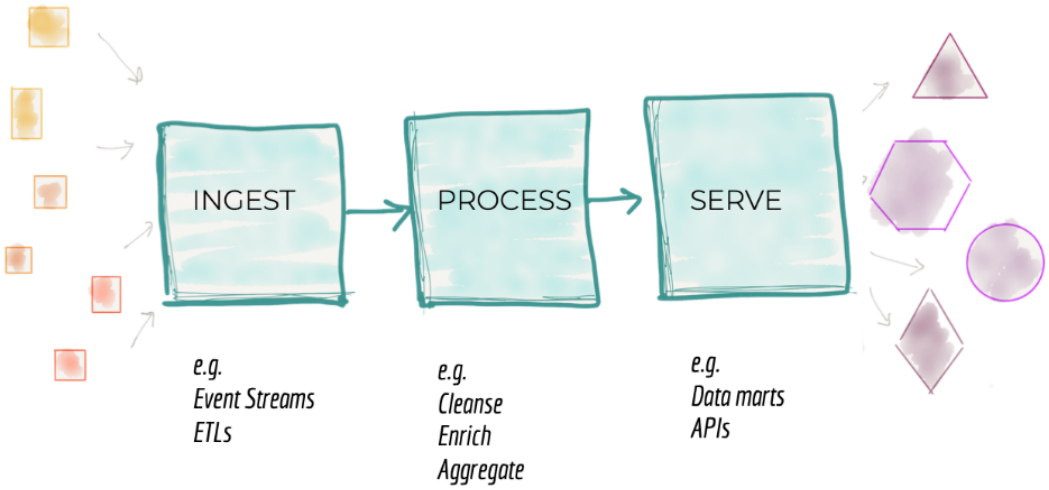

Figura 2: Decomposição arquitetônica da plataforma de dados. Fonte: (DEHGHANI, 2019).

- **Silos de propriedades super especializados**
    - O terceiro ponto de falha das arquiteturas de dados atuais está relacionado a como são estruturados os times que constroem e mantêm estas plataformas.
    - Quando olhamos para as pessoas que operam as plataformas de dados, encontramos engenheiros de dados hiper especializados, isolados das unidades operacionais da organização; onde os dados se originam ou onde são utilizados para tomadas de decisão. Os engenheiros das plataformas de dados não são apenas isolados organizacionalmente, mas também separados e agrupados em uma equipe com base em sua experiência técnica em ferramentas de *big data*, por exemplo, e muitas vezes sem nenhum conhecimento sobre o negócio.
    - Geralmente, o que encontramos em uma empresa são: equipes geradoras de dados desconectadas; consumidores de dados frustrados, lutando por um lugar no topo da lista de pendências (*backlog*) de uma equipe de dados sobrecarregada. Criamos uma arquitetura e estrutura organizacional que não escala e não entrega o valor prometido: criar uma organização orientada a dados (DEHGHANI, 2019).

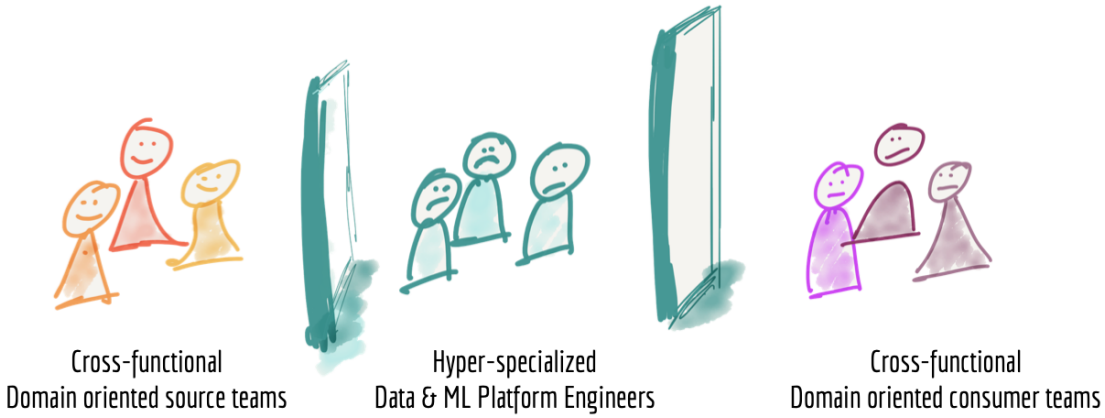

Figura 3: Silos hiper especializados de times de plataforma de dados. Fonte: (DEHGHANI, 2019).

A próxima arquitetura corporativa de plataforma de dados

A nova proposta de arquitetura de plataforma de dados trata a questão da onipresença de dados e fontes por meio do conceito de *Data Mesh* (malha de dados distribuída).

De acordo com Dehghani (2019), a solução dos problemas das arquiteturas de dados atuais requer uma mudança de paradigma. Dehghani (2019) sugere que a próxima arquitetura de plataforma de dados corporativos esteja na convergência dos conceitos de Arquitetura Distribuída Orientada a Domínio (*Distributed Domain Driven Architecture*), Design de Plataforma Self-Service (*Self-serve Platform Design*) e Pensamento em Dados como Produto (*Product Thinking with Data*).

Decomposição e propriedade de dados orientada a domínio

Para que a plataforma de dados monolítica seja descentralizada, será necessário reverter a forma como pensamos sobre os dados, sua localidade e propriedade. Em vez de transferir os dados dos domínios para um *Data Lake* ou plataforma de propriedade central, os domínios precisam hospedar

e servir seus conjuntos de dados de domínio de maneira facilmente consumível. O armazenamento físico pode ser uma infraestrutura centralizada, como buckets do Amazon S3, mas o conteúdo e a propriedade dos conjuntos de dados devem permanecer com o domínio que os gera (DEHGHANI, 2019).

Dados de domínios orientado à origem

Em uma situação madura e ideal, um sistema de informação e sua equipe ou unidade organizacional não são apenas responsáveis por fornecer requisitos e recursos de negócios, mas também por fornecer as verdades de seu domínio de negócios como conjuntos de dados de domínio de origem (DEHGHANI, 2019).

Na escala corporativa, nunca há um mapeamento de um para um entre um conceito de domínio e um sistema de origem. Muitas vezes existirão muitos sistemas que podem servir partes dos dados que pertencem a um domínio, alguns legados e alguns mais susceptíveis a mudanças. Portanto, podem haver muitos conjuntos de dados alinhados à origem, que precisam ser agregados a um conjunto de dados alinhado ao domínio coeso.

Observe que os conjuntos de dados de domínio devem ser separados dos conjuntos de dados dos sistemas internos. A natureza dos conjuntos de dados de domínio é muito diferente dos dados internos que os sistemas de informação usam para fazer seu trabalho. Eles têm um volume muito maior, representam fatos temporais imutáveis e mudam com menos frequência do que seus sistemas de origem. Por esse motivo, o armazenamento subjacente real deve ser adequado para *big data* e separado dos bancos de dados operacionais existentes (DEHGHANI, 2019).

Pipelines distribuídos como implementação interna do domínio

A necessidade de limpeza, preparação, agregação e fornecimento de dados permanece, assim como o uso do *pipeline* de dados. Nesta arquitetura, um pipeline de dados é simplesmente uma complexidade interna e implementação do domínio de dados, sendo tratado internamente dentro do domínio. Como resultado, veremos uma distribuição dos estágios dos pipelines de dados em cada domínio.

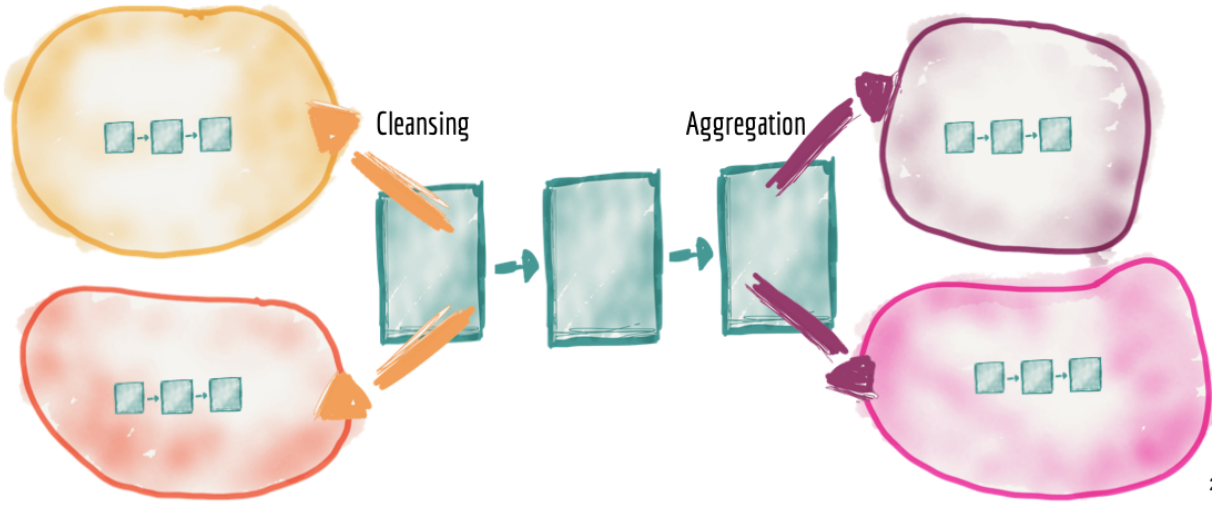

Figura 4: Pipelines distribuídos entre domínios como um detalhe interno de implementação. Fonte: (DEHGHANI, 2019).

Convergência entre dados e product thinking

A distribuição da propriedade dos dados e a implementação de *pipelines* nas mãos dos domínios de negócios levantam uma preocupação importante sobre acessibilidade, usabilidade e harmonização de conjuntos de dados distribuídos.

Para que uma plataforma de dados distribuídos seja bem-sucedida, as equipes de dados de domínio devem aplicar o pensamento de produto (*Product Thinking*) com rigor semelhante aos conjuntos de dados que fornecem; considerando seus ativos de dados como seus produtos e os outros cientistas/engenheiros de dados da organização como seus clientes (DEHGHANI, 2019).

Para prover a melhor experiência aos consumidores de dados, os produtos de dados de domínio precisam possuir as seguintes qualidades básicas (DEHGHANI, 2019):

- **Detectável:** um produto de dados precisa ser facilmente localizável. Uma implementação comum é ter um registro, um catálogo de dados, de todos os produtos de dados disponíveis com suas meta-informações, como seus proprietários, fonte de origem, linhagem, conjuntos de dados de amostra, etc.
- **Endereçável:** um produto de dados, uma vez descoberto, deve ter um endereço exclusivo seguindo uma convenção global que ajude seus usuários a acessá-lo programaticamente.
- **Confiável e verdadeiro:** Ninguém vai usar um produto que eles não podem confiar. A realização de limpeza de dados e testes automatizados de integridade no ponto de criação do produto de dados são algumas das técnicas a serem utilizadas para fornecer um nível aceitável de qualidade. Fornecer proveniência e linhagem de dados como metadados associados a cada produto ajuda os consumidores a ganhar mais confiança de sua adequação às suas necessidades específicas. Cada produto de dados deve definir e garantir o nível de sua integridade e veracidade por meio de um conjunto de SLOs (*Service Level Objectives*).
- **Alto descritivo:** Produtos de qualidade não requerem a manipulação do consumidor para serem usados: eles podem ser descobertos, compreendidos e consumidos de forma independente. Construir conjuntos de dados como produtos com o mínimo de atrito para os usuários (engenheiros e cientistas de dados) requer semântica e sintaxe bem descritas, idealmente acompanhadas de conjuntos de dados de amostra.
- **Interoperável:** A chave para possibilitar a correlação eficaz de dados entre domínios é seguir certos padrões e regras de harmonização. Tais padronizações devem pertencer a uma governança global, para permitir a interoperabilidade entre os conjuntos de dados de domínio. As preocupações comuns de tais esforços de padronização são a formatação do tipo de campo, a identificação de polissemia (multiplicidade de sentidos) em diferentes domínios, convenções de endereço, campos de metadados comuns, etc. Por exemplo, em uma área de negócio de *streaming* de mídia, um 'artista' pode aparecer em diferentes domínios e ter diferentes atributos e identificadores em cada domínio. Uma abordagem é considerar 'artista' com uma **entidade federada** e um identificador de entidade

federada global único para o 'artista', de forma semelhante à forma como identidades federadas são gerenciadas. Interoperabilidade e padronização das comunicações, governadas globalmente, é um dos pilares fundamentais para a construção de sistemas distribuídos.
- **Seguro e governado por um controle de acesso global:** Acessar conjuntos de dados de produtos com segurança é uma obrigação, quer a arquitetura seja centralizada ou não. No mundo dos produtos de dados orientados a domínios descentralizados, o controle de acesso deve ser aplicado em uma granularidade mais fina, para cada produto de dados de domínio.

Equipes multifuncionais

Equipes responsáveis por domínios (áreas de negócio) responsáveis pelo fornecimento de dados como produtos, precisam contar com novos conjuntos de habilidades: (a) o proprietário do produto de dados e (b) engenheiros de dados.

Um proprietário de produto de dados toma decisões em torno da visão e do *roadmap* para os produtos de dados, preocupa-se com a satisfação de seus consumidores e mede e aprimora continuamente a qualidade e a riqueza dos dados que seu domínio possui e produz. É responsável pelo ciclo de vida dos conjuntos de dados do domínio, buscando um equilíbrio entre as necessidades concorrentes dos consumidores de dados do domínio (DEHGHANI, 2019).

Para construir e operar os *pipelines* de dados internos dos domínios, as equipes devem incluir engenheiros de dados. Um efeito colateral vantajoso dessa equipe multifuncional é o compartilhamento de habilidades entre os membros da equipe. Observa-se em algumas situações do mundo real que alguns engenheiros de dados, embora competentes no uso das ferramentas da sua área de atuação, carecem de práticas concernentes à engenharia de *software*, como entrega contínua e testes automatizados, por exemplo, quando se trata de construir ativos de dados. Da mesma forma, os engenheiros de *software* que estão construindo sistemas de informação geralmente não têm experiência na utilização de soluções de engenharia de dados (DEHGHANI, 2019).

Os dados devem ser tratados como uma peça fundamental de qualquer ecossistema de *software*, portanto, engenheiros e generalistas de *software* devem adicionar a experiência e o conhecimento do desenvolvimento de produtos de dados ao seu cinto de ferramentas.

Convergência de design de plataforma de dados *self-service*

A construção de infraestruturas comuns como plataforma é um problema bem compreendido e resolvido, embora reconhecidamente as ferramentas e técnicas não sejam tão maduras para ecossistemas de dados.

A utilização de recursos de infraestrutura agnóstica a domínio a partir de uma plataforma de infraestrutura de dados resolve a necessidade de duplicar o esforço de configurar mecanismos de *pipeline* de dados, armazenamento e infraestrutura de *streaming*. Neste sentido, uma equipe de infraestrutura de dados deve fornecer a tecnologia necessária que os domínios precisam para

capturar, processar, armazenar e disponibilizar seus produtos de dados (DEHGHANI, 2019).

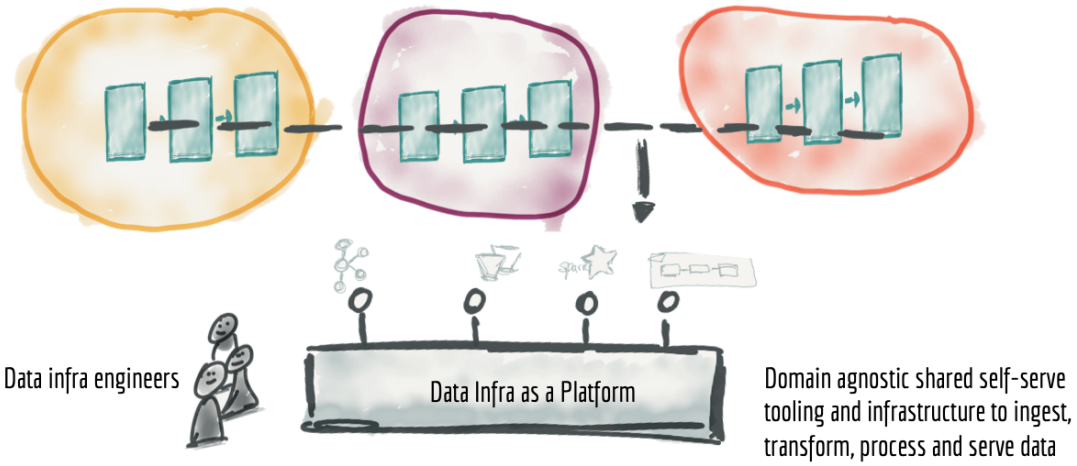

Figura 5: Plataforma de infraestrutura de dados agnóstica a domínio. Fonte: (DEHGHANI, 2019).

A chave para construir uma infraestrutura de dados como plataforma é: (a) não incluir nenhum conceito específico de domínio ou lógica de negócios, mantendo-a agnóstica de domínio, e (b) certificar-se de que a plataforma oculte toda a complexidade subjacente e forneça os componentes de infraestrutura de dados em uma forma *self-service (DEHGHANI, 2019).*

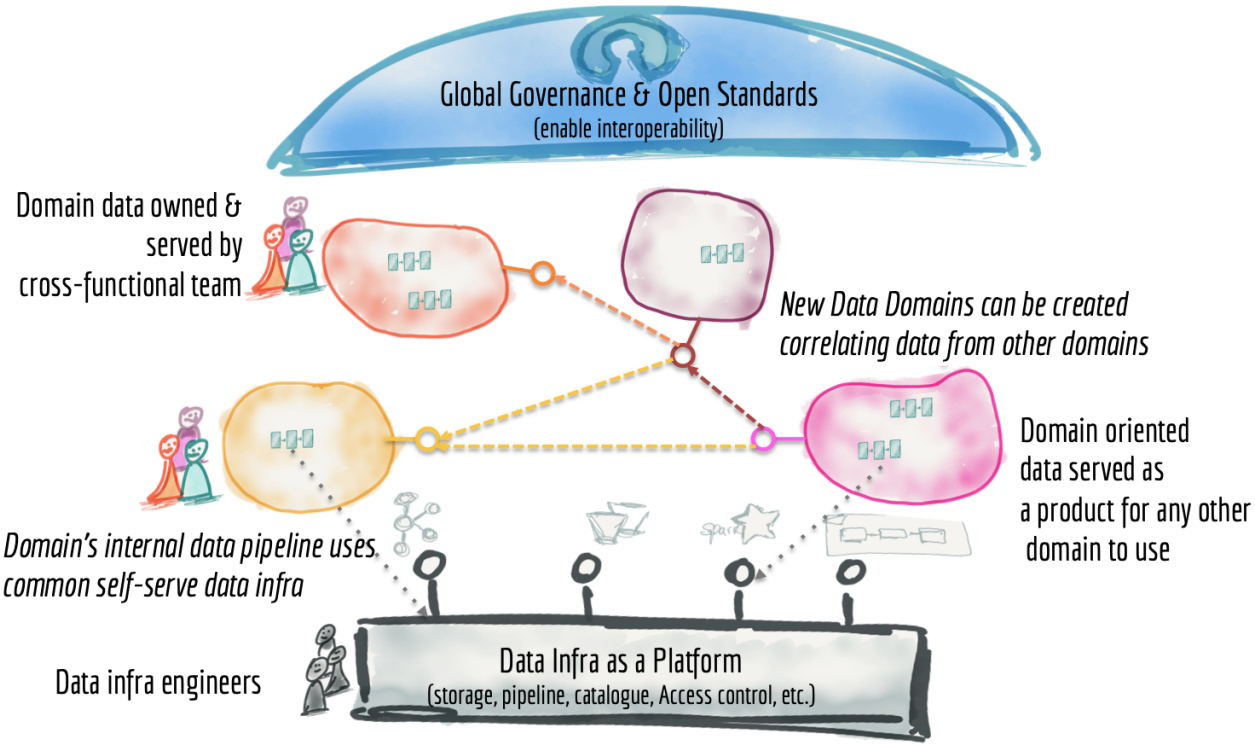

Figura 6: Visão de altíssimo nível de uma arquitetura de Data Mesh. Fonte: (DEHGHANI, 2019).

Você deve estar se perguntando onde o *Data Lake* ou o *Data Warehouse* se encaixam nesta arquitetura. Eles são simplesmente nós na malha. É muito provável que não precisemos de um Data Lake, porque os *logs* distribuídos e os dados originais de sistemas estão disponíveis para exploração de diferentes conjuntos de dados imutáveis endereçáveis como produtos. No entanto,

nos casos em que precisamos fazer alterações no formato original dos dados para exploração adicional, como rotulagem, o domínio com essa necessidade pode criar seu próprio *Data Lake* ou *Data Warehouse (DEHGHANI, 2019)*.

A principal mudança é tratar o produto de dados de domínio como uma preocupação de primeira classe e as ferramentas e o pipeline de *Data Lake* / *Data Warehouse* como uma preocupação de segunda classe - um detalhe de implementação. Isso inverte o modelo mental atual de um Data Lake / DW centralizado para um ecossistema de produtos de dados que funcionam juntos, uma malha de dados (DEHGHANI, 2019).

Revisão #6
Criado 2022-03-11 12:41:36 UTC por FLAVIO LOPES DE MORAIS
Atualizado: 2022-12-14 09:02:06 UTC por FLAVIO LOPES DE MORAIS