

The background features a complex geometric design. On the left, there are several 3D rectangular blocks of varying heights and colors, including shades of green, blue, and cyan. On the right, there is a repeating pattern of white circles on a blue background, which overlap to form a series of white, leaf-like shapes. A large white rectangle is positioned in the center-right, containing the title and author information.

EVENT STORMING

For Microservices

TANIA RAQUEL STORMOVSKI DE ANDRADE

A thin, vertical blue line is located in the lower right quadrant of the page.

ABOUT ME

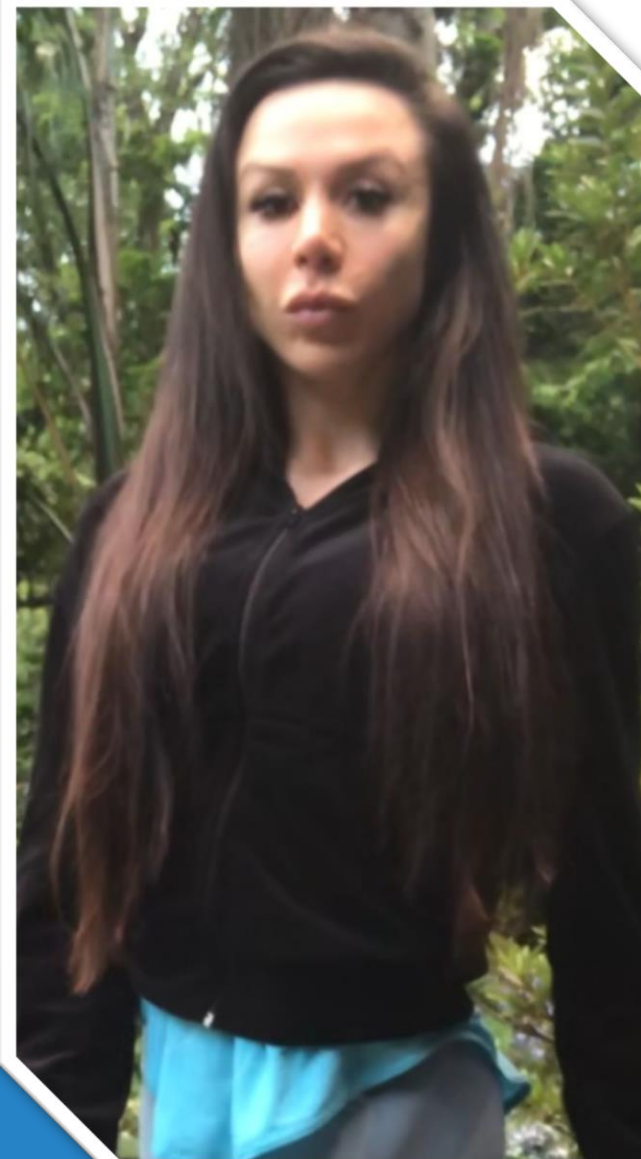
Possuo experiência de mais de 15 anos na área de desenvolvimento de software, trabalhando com .NET desde sempre.

Sempre tentei ver além do código, entender aspectos mais amplos da solução e foi assim que a arquitetura foi ficando cada vez mais presente e relevante para mim.

Atualmente atuando como arquiteta na **Avanade**

Linkedin <https://www.linkedin.com/taniastormovski/>

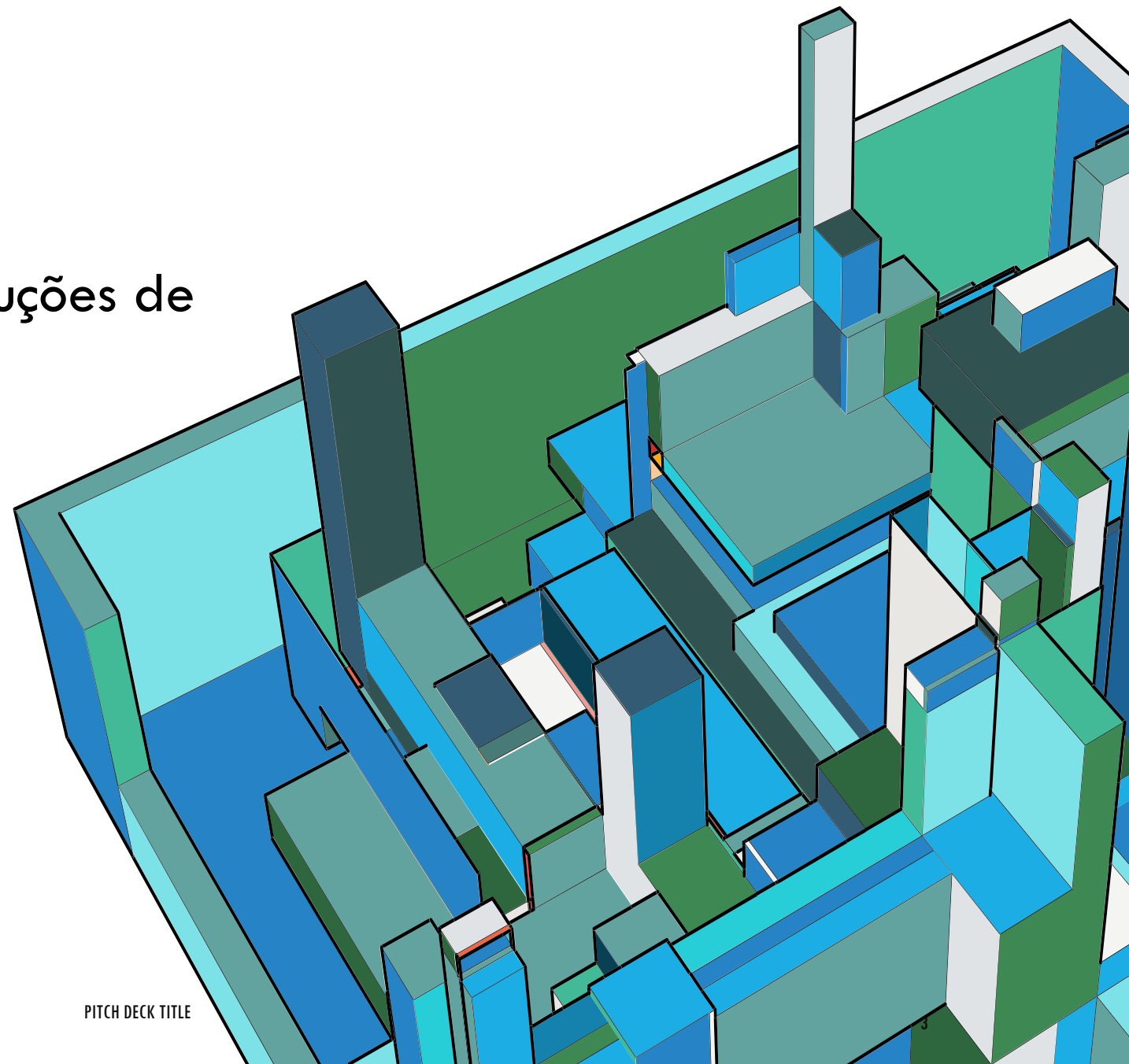
Instagram @tetestorm



CONTEXTO

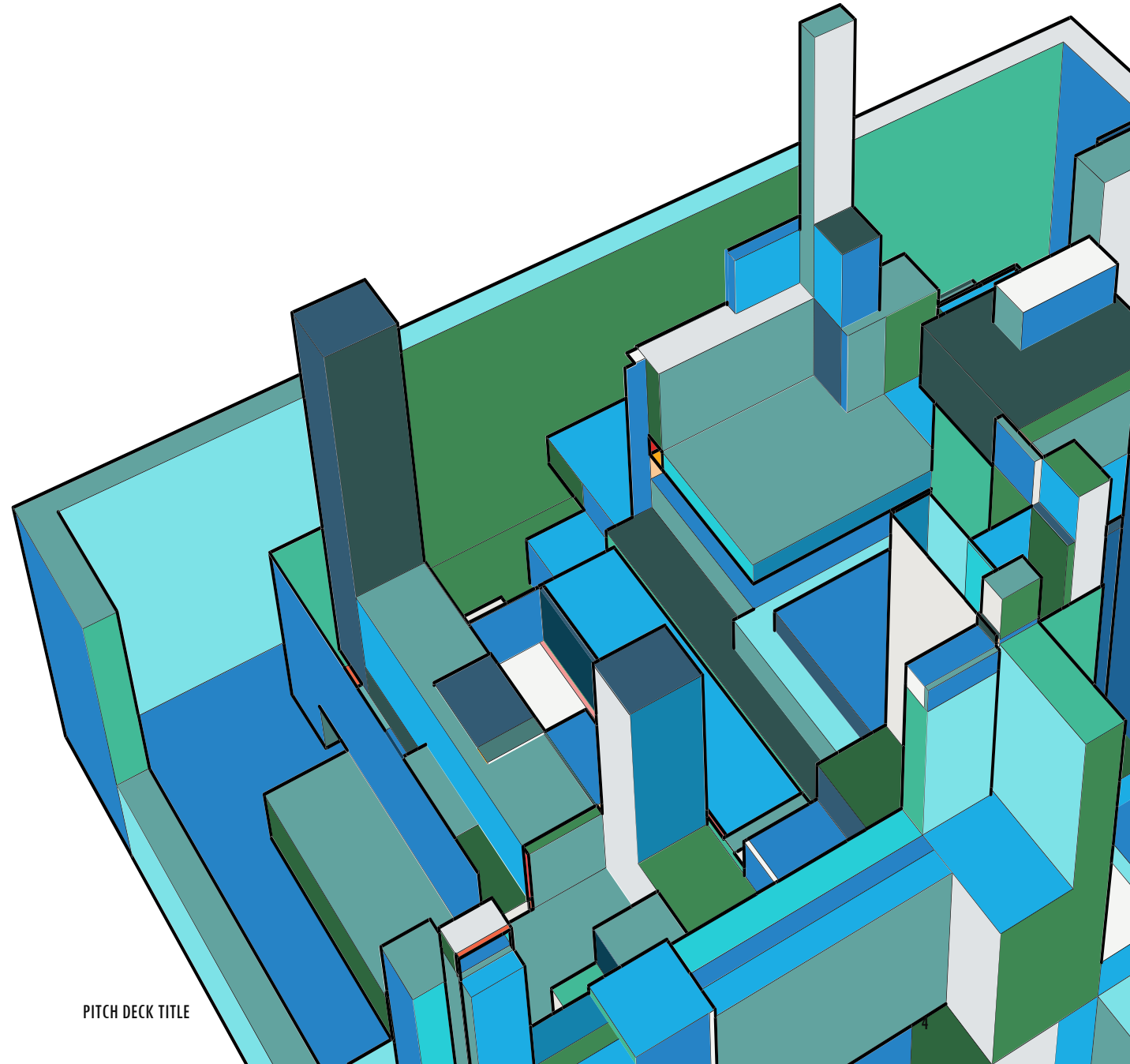
Qual a realidade atual das soluções de grande parte das empresas?

- Legados
- Monolíticos
- Alto Acoplamento
- Não escaláveis



STARTUPS

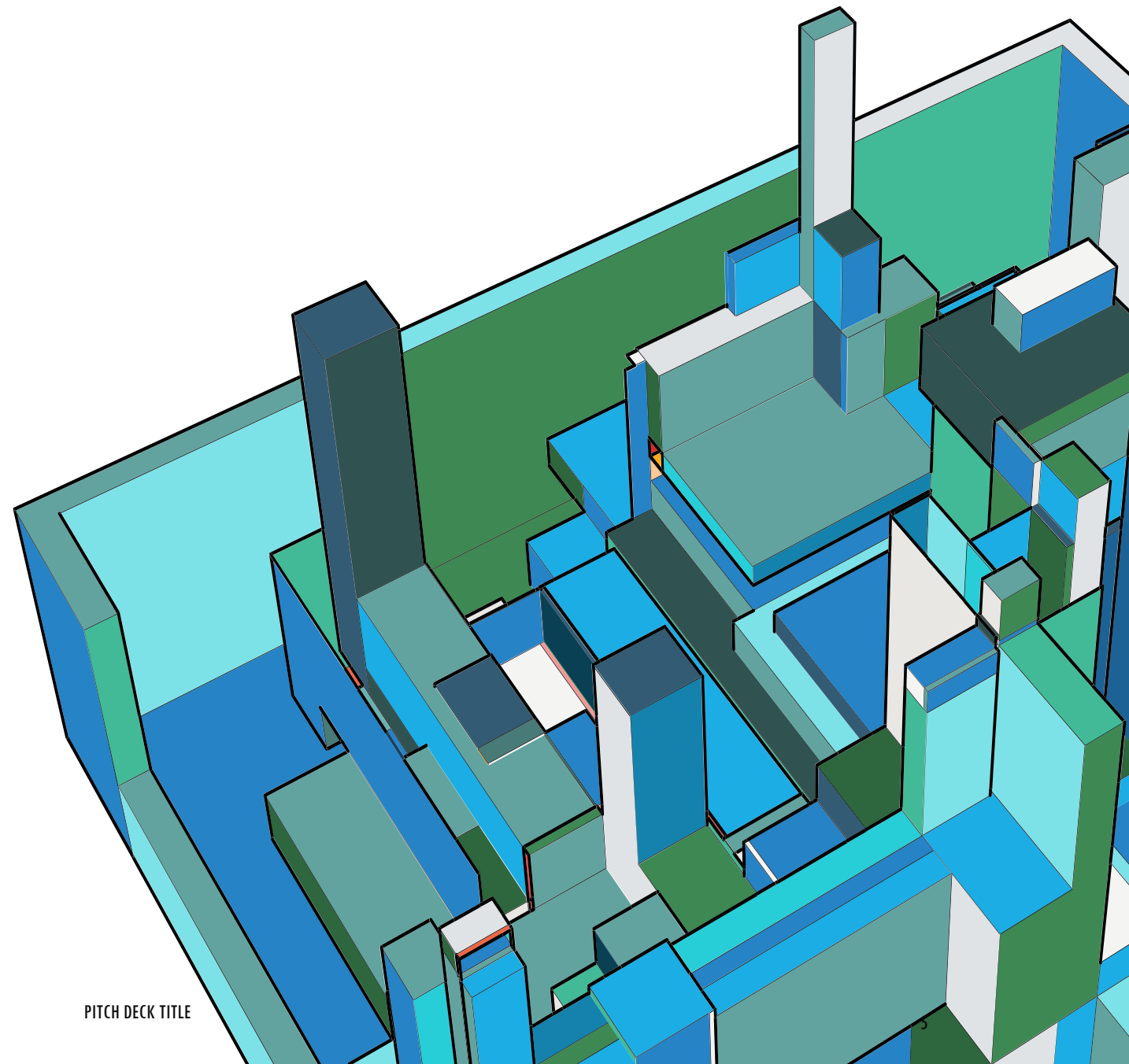
Borbulhando de idéias inovadoras que precisam ser colocadas em prática de maneira rápida para não perder oportunidades de investimento ou o time to market



ONDE ENTRAM OS MICROSSERVIÇOS?

Porque este tipo de arquitetura está ganhando tanta visibilidade e tem sido a bola da vez?

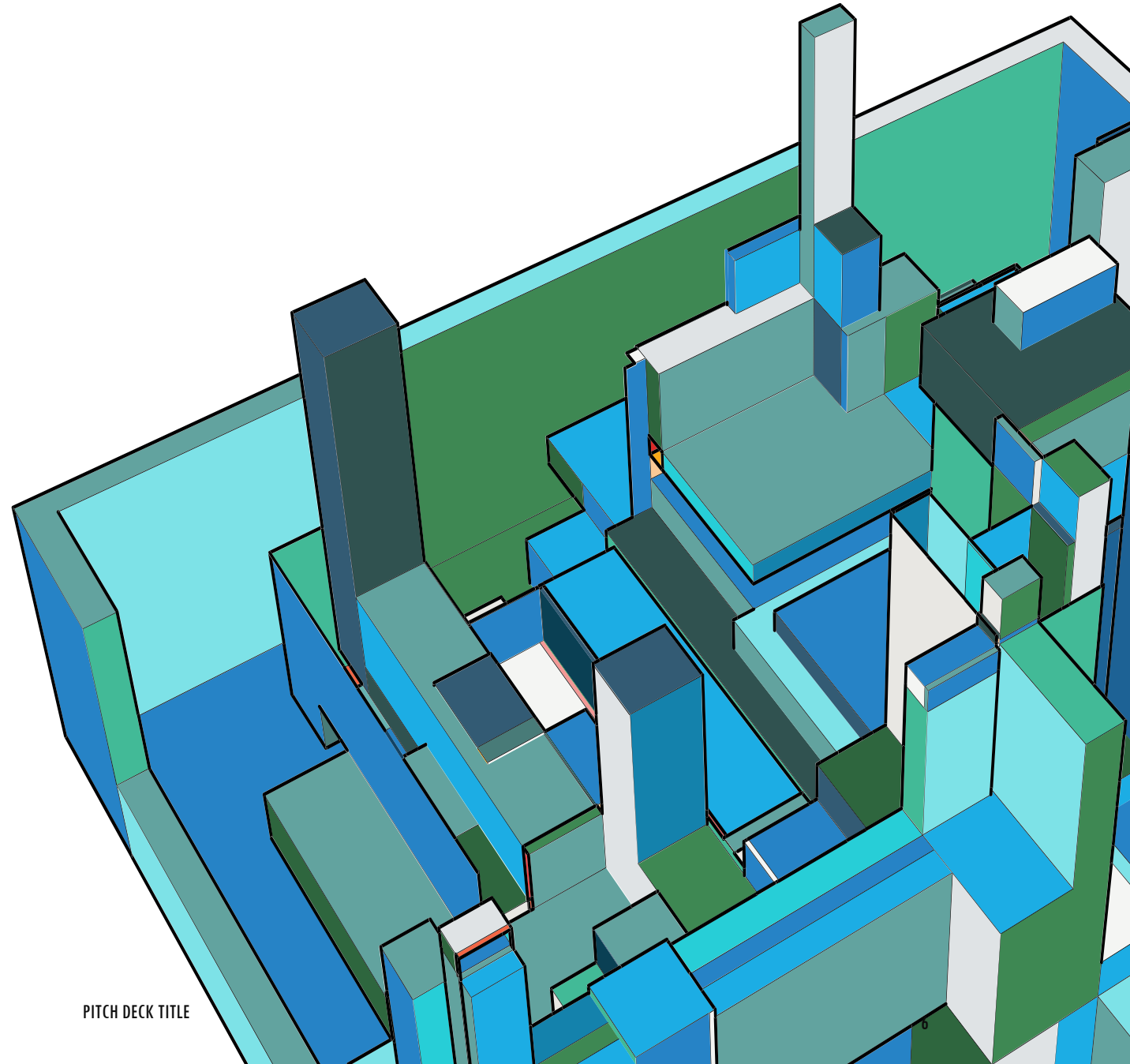
Desde C level até os desenvolvedores, se você perguntar qual seria a arquitetura desejada certamente a resposta vai ser microsserviços



PORQUE TAL POPULARIDADE?

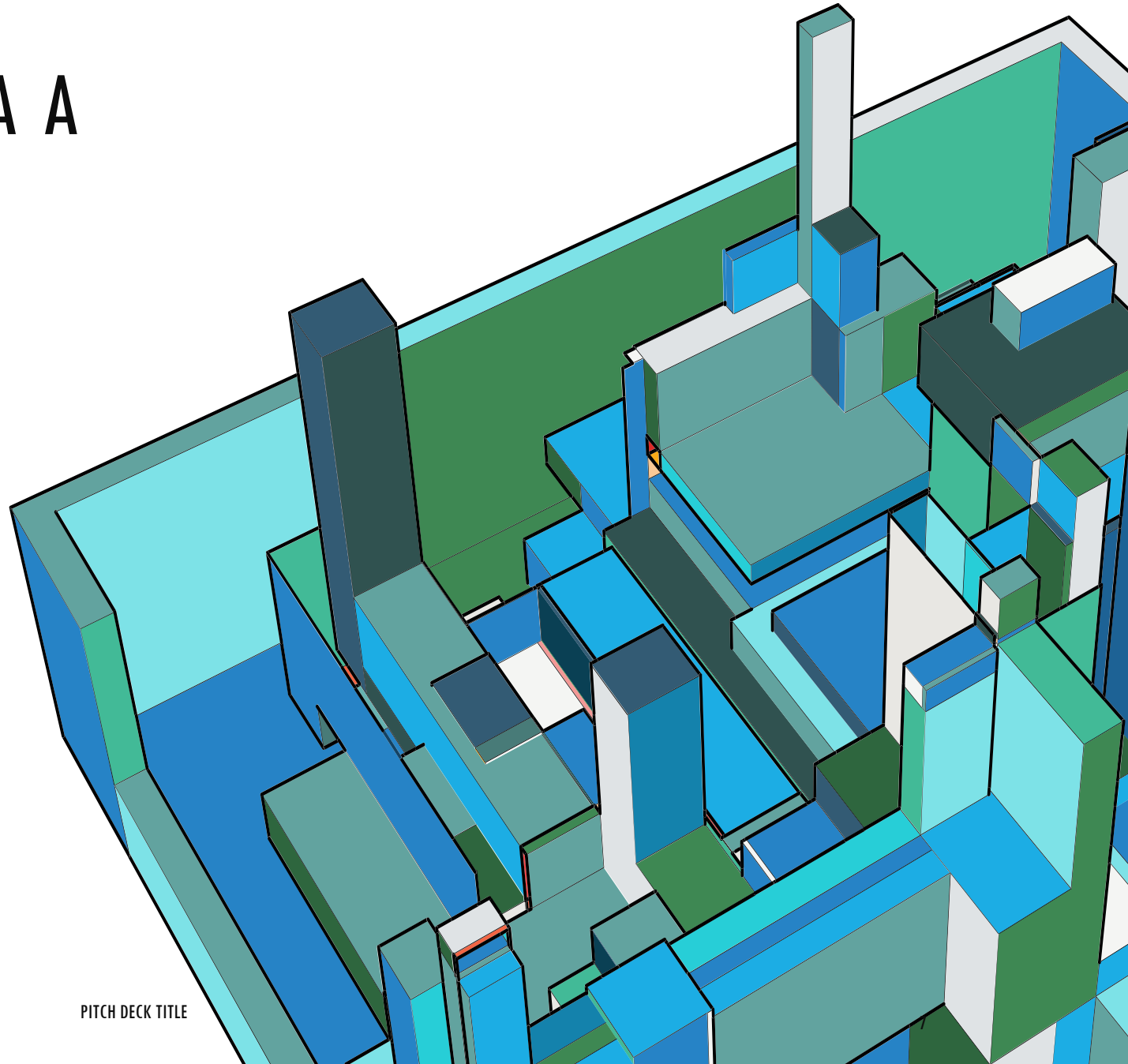
A lei de Conway é um ditado que afirma que as organizações projetam sistemas que refletem sua própria estrutura de comunicação.

"Qualquer organização que projeta um sistema (definido de forma mais ampla aqui do que apenas sistemas de informação) inevitavelmente produzirá um design cuja estrutura é uma cópia da estrutura de comunicação da organização"



LEI DE CONWAY APLICADA A MICROSSERVIÇOS

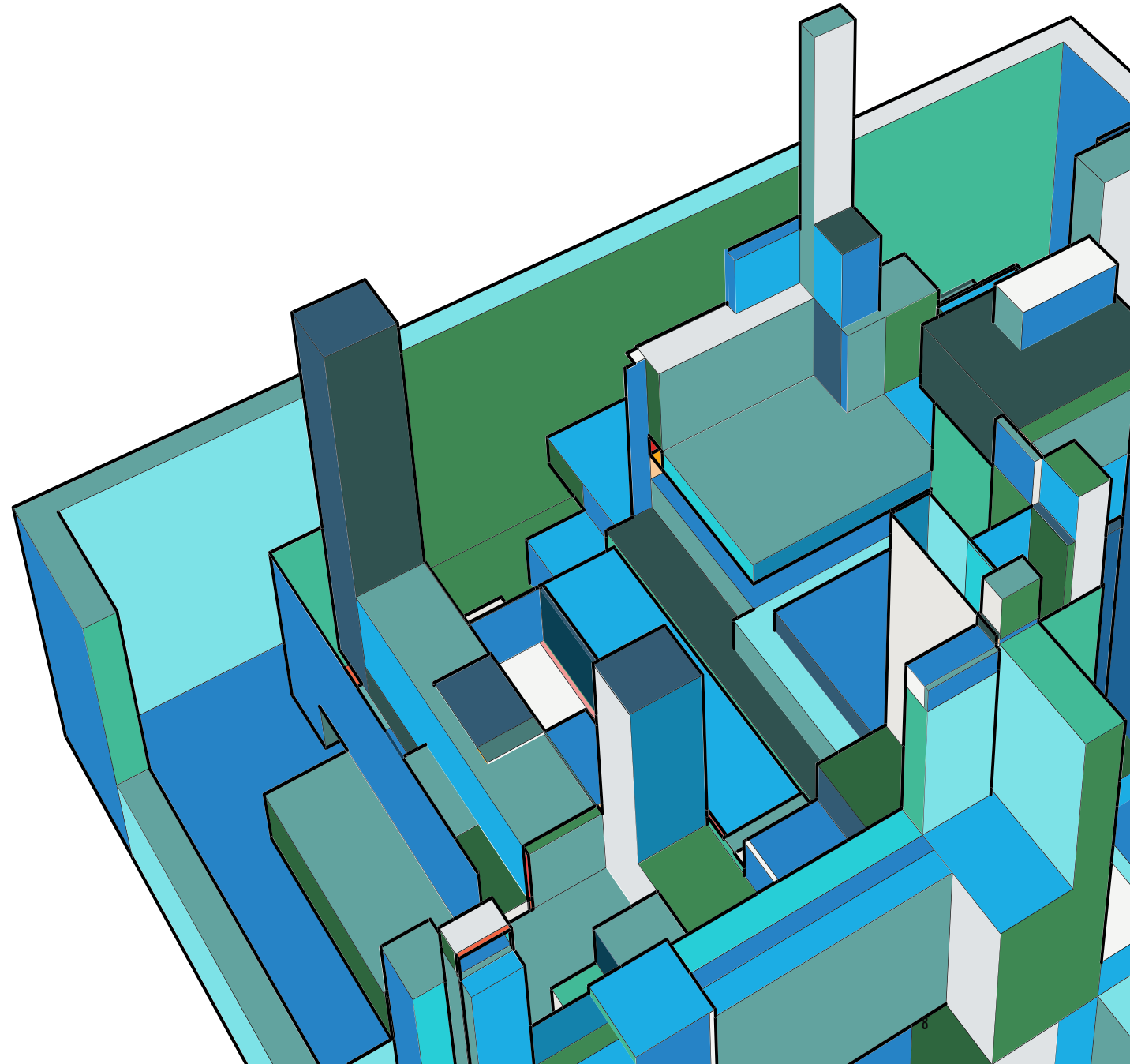
"Qualquer organização que projeta um sistema (definido de forma mais ampla aqui do que apenas sistemas de informação) inevitavelmente produzirá um design cuja estrutura é uma cópia da estrutura de comunicação da organização"



REFLEXÃO

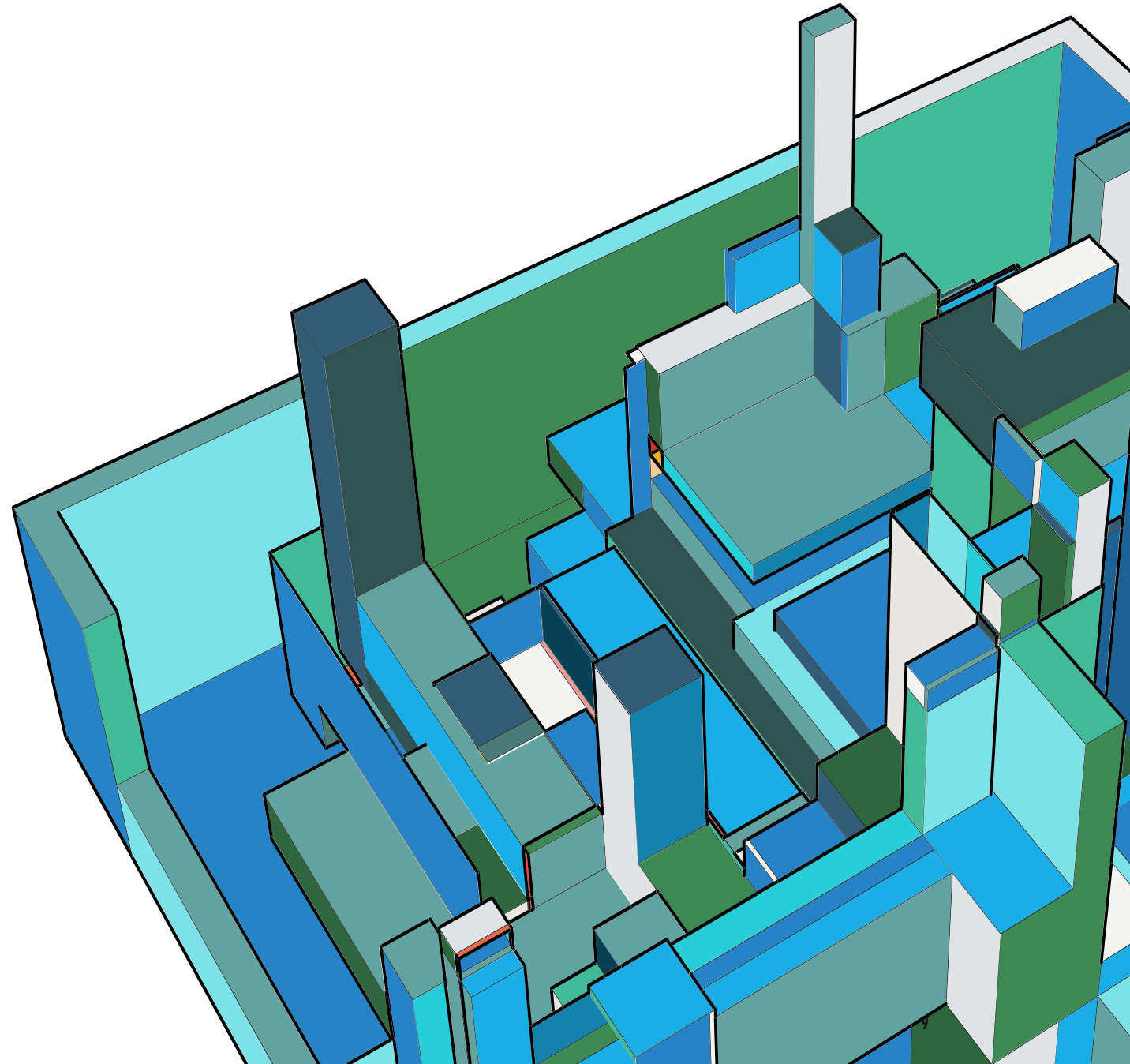
Será que a tendência à adoção de microsserviços é uma consequência dessa lei?

Por traduzir os modelos de organização, e atualmente, com a predominância do trabalho remoto, modelos de trabalho e equipes estamos sendo levados à adesão desta arquitetura?



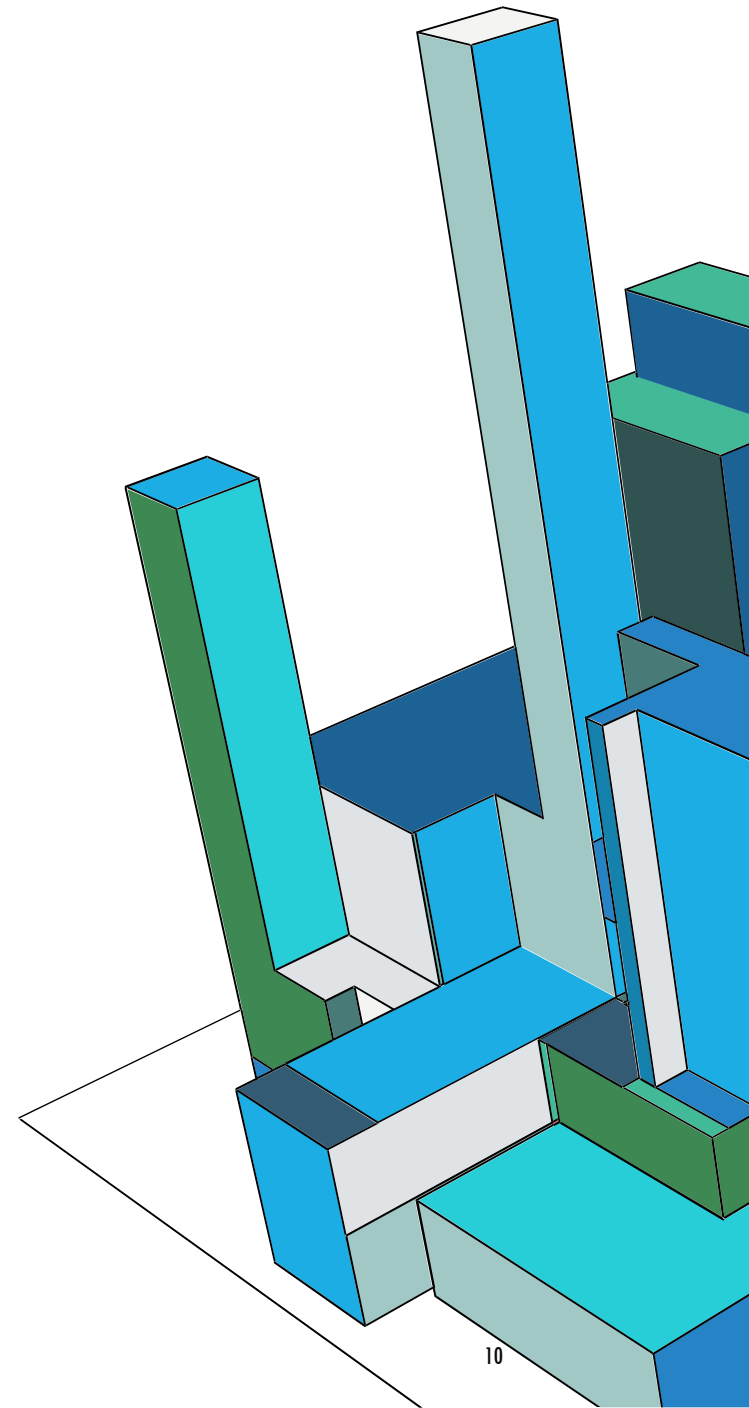
DESAFIOS

- Aumento da complexidade
- Monitoramento
- Consistência de dados
- Segurança
- Troubleshooting
- Rede
- Custos
- Time preparado
- Entre muitos outros..



ONDE ENTRA O EVENT STORMING?

Entra como um apoio para traduzirmos nosso problema, nossos domínios até chegarmos aos possíveis microsserviços candidatos



GRANULARIDADE

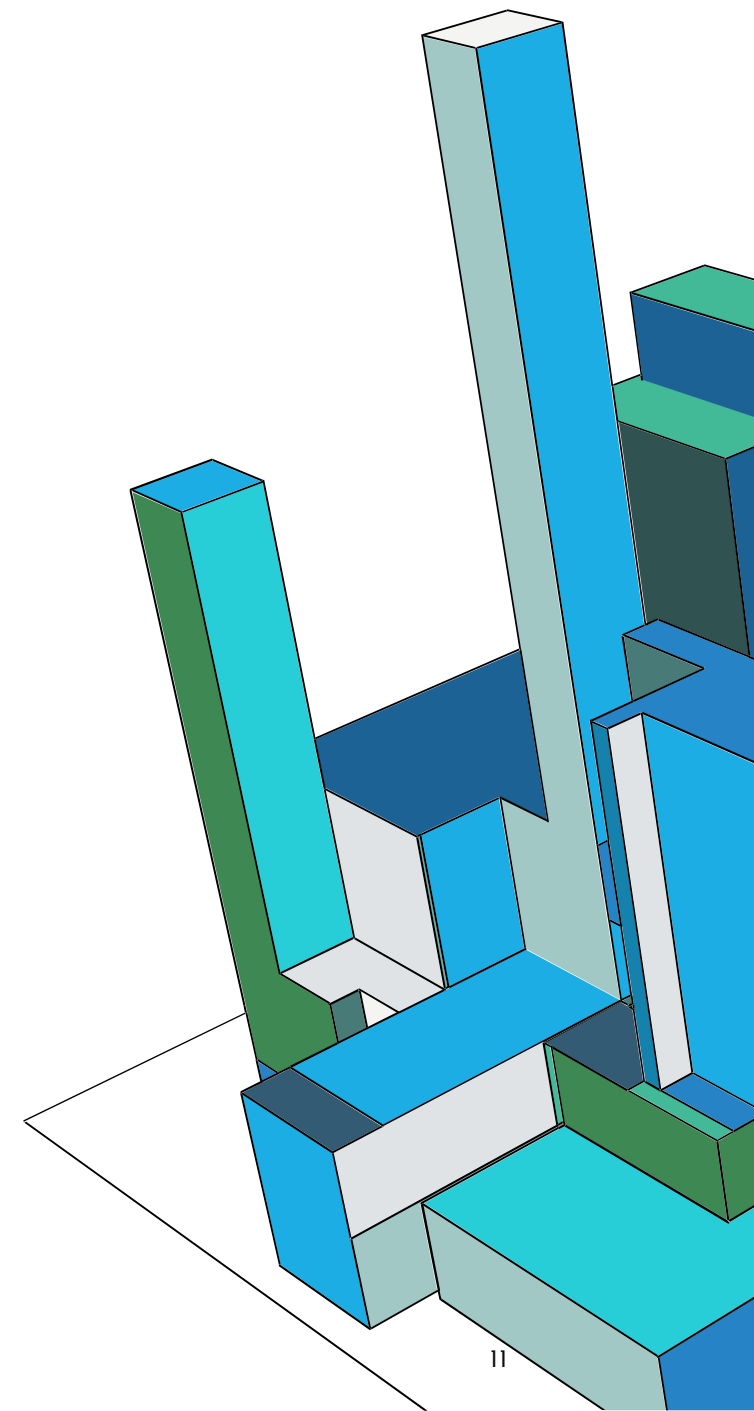
"A meta de identificar os limites do modelo e o tamanho de cada microsserviço não é obter a separação mais granular possível, embora você deva tender a usar microsserviços pequenos se possível.

Em vez disso, sua meta deve ser obter a separação mais significativa orientada pelo seu conhecimento do domínio.

A ênfase não está no tamanho, mas em funcionalidades empresariais. Além disso, se for necessária uma clara coesão para uma determinada área do aplicativo com base em um grande número de dependências, isso será a indicação da necessidade de um único microsserviço também.

A coesão é uma maneira de identificar como separar ou agrupar microsserviços. Por fim, enquanto você obtém mais conhecimento sobre o domínio, deve adaptar o tamanho do seu microsserviço iterativamente. Localizar o tamanho correto não é um processo único."

<https://docs.microsoft.com/pt-br/dotnet/architecture/microservices/architect-microservice-containerapplications/identify-microservice-domain-model-boundaries>

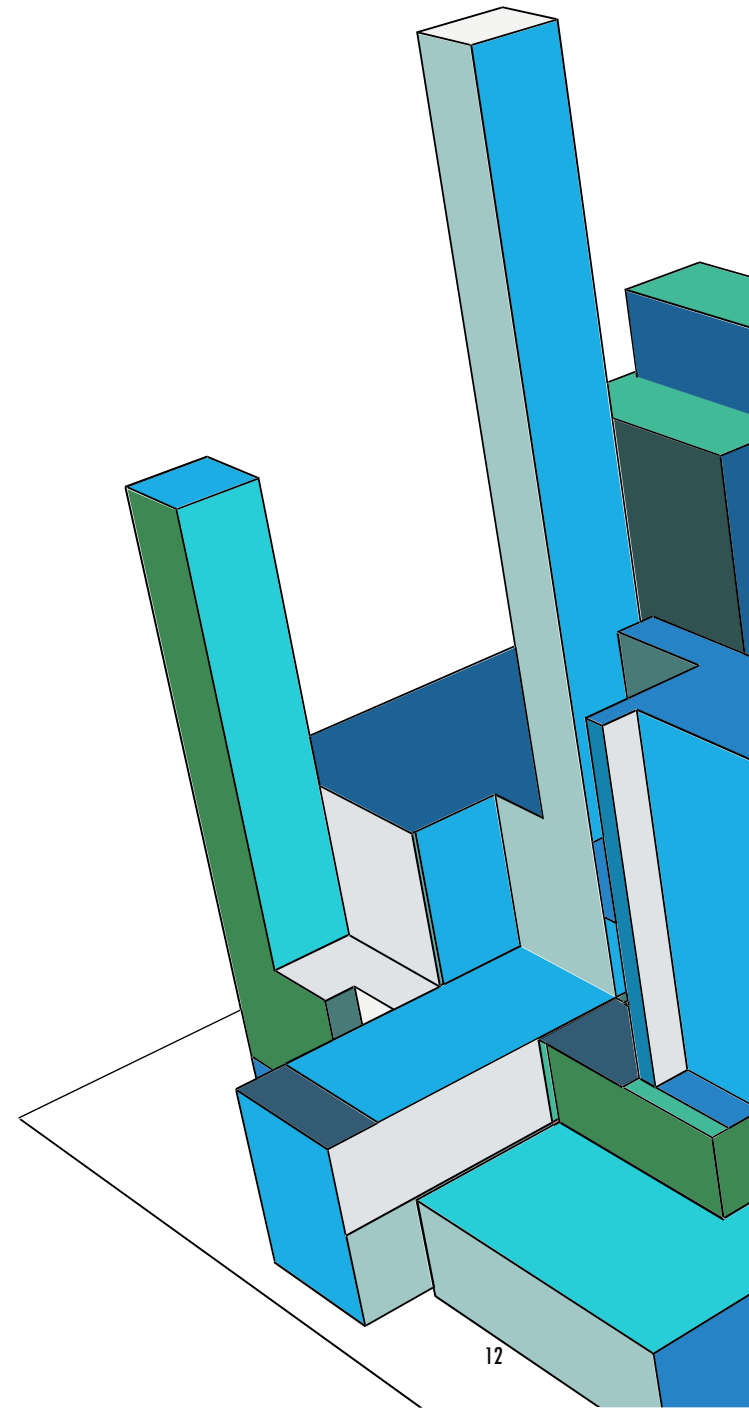


E O QUE É O EVENT STORMING

"Event Storming é um formato de workshop para explorar domínios complexos"

— **Alberto Brandolini**

A intenção é tentar capturar um sistema em termos das coisas que acontecem, os Eventos.



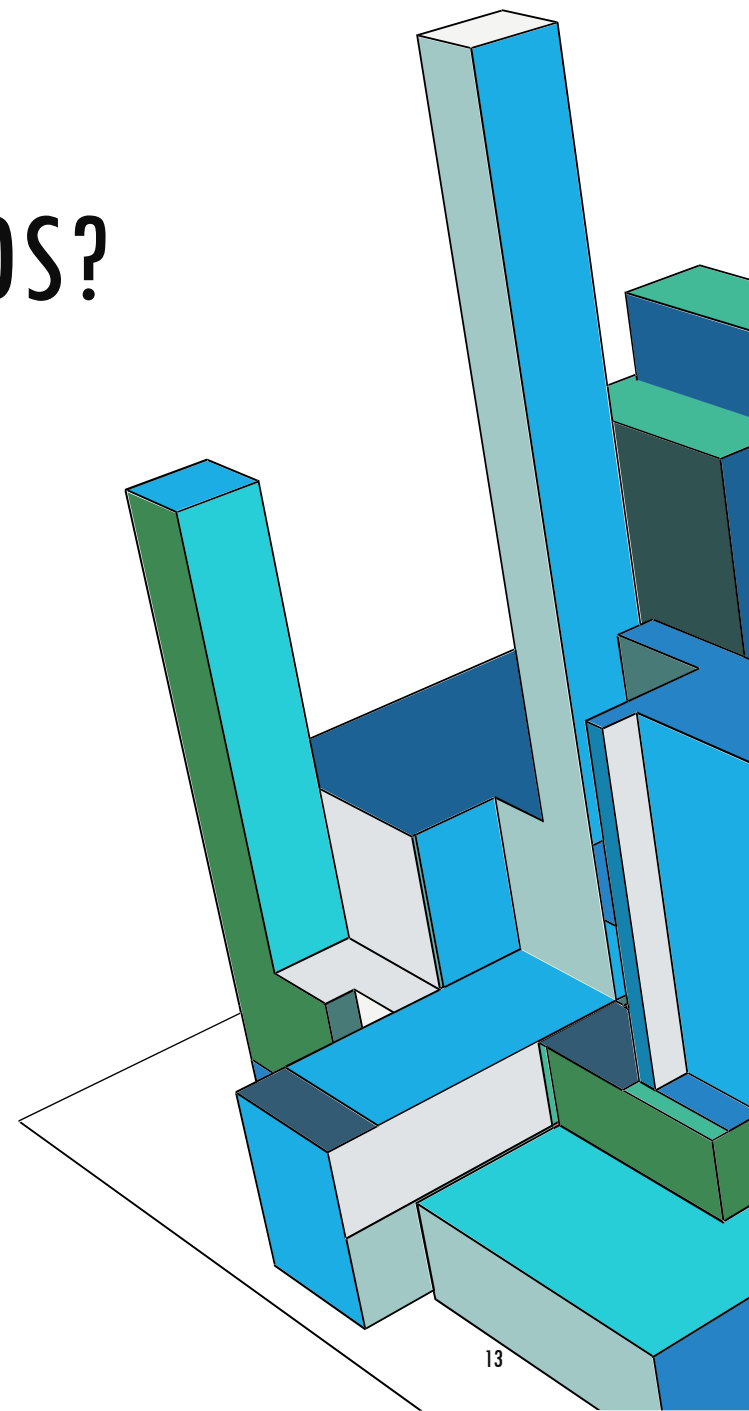
QUAL A RELAÇÃO COM MICROSERVIÇOS?

MUDANÇA DE PARADIGMA

Diminui a curva de aprendizado trazendo ao universo de Eventos aqueles que somente trabalharam em arquiteturas monolíticas e orientados a dados ajudando na mudança do mindset do time

MODELO DINÂMICO E COMPORTAMENTAL

O modelo resultante é totalmente comportamental e dinâmico tornando o entendimento facilitado e o conhecimento explícito



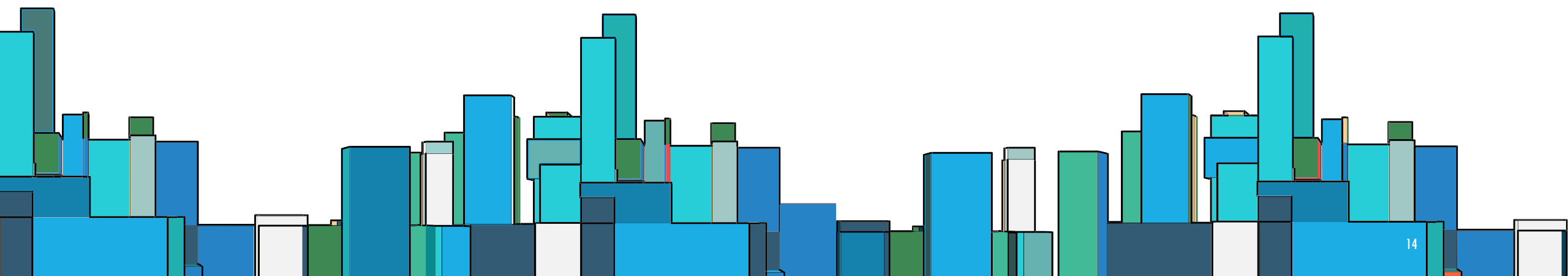
BENEFÍCIOS ADICIONAIS

MODELAGEM RÁPIDA

Diferente de outras técnicas onde a construção é a passos lentos com Event Storming em poucas e às vezes, até mesmo em uma sessão chegamos à um modelo representativo

PROMOVE APRENDIZADO

Como a dinâmica é colaborativa promove o engajamento e faz com que todos tenham a big picture completa do domínio, trazendo inclusive possíveis pontos de inconsistência sendo um ótimo ponto inicial de on boarding para novas pessoas no time



An abstract graphic on the left side of the page consists of several 3D rectangular blocks of varying heights and widths. The blocks are colored in shades of blue and green, with some having black outlines. They are arranged in a way that suggests a staircase or a series of steps, with some blocks overlapping others. The perspective is from a low angle, looking up at the blocks.

QUEM CRIOU

ALBERTO BRANDOLINI

Ele é consultor 360° (arquiteto, mentor, gerente e desenvolvedor) na área de Tecnologia da Informação.

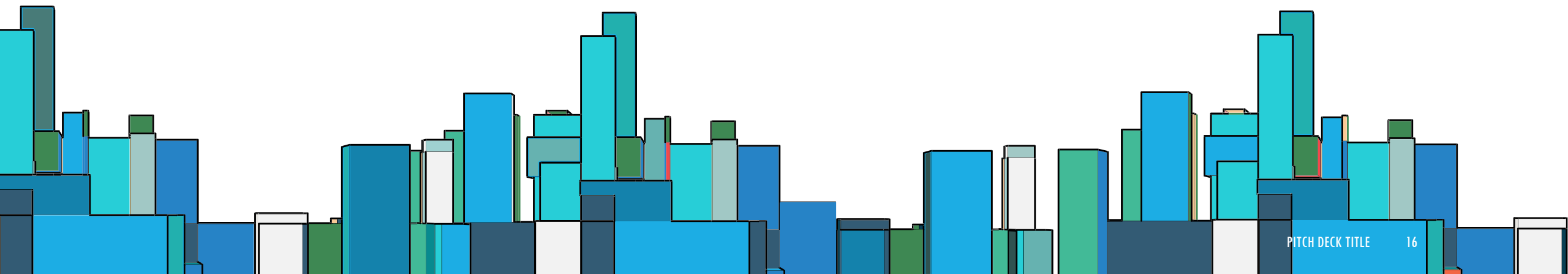
COMO SURTIU

"Eu precisava definir algo rapidamente, não tenho tempo para desenhar um diagrama UML preciso, então peguei o rolo de papel - havia quadros brancos, mas havia outras equipes também - e comecei a estabelecer meu entendimento com os blocos de construção DDD e CQRS, e cheguei a algo em alguns minutos, enquanto as outras equipes nem tinham começado."

COMO SE POPULARIZOU?

A verdadeira explosão aconteceu quando **Vaughn Vernon** (autor do famoso DDD Destiled que é uma visão revisitado do clássico Domain Driven Desing do Eric Evans) convidou Brandolini como 'convidado especial' em sua turnê IDDD em 2013, na Bélgica e na Polônia

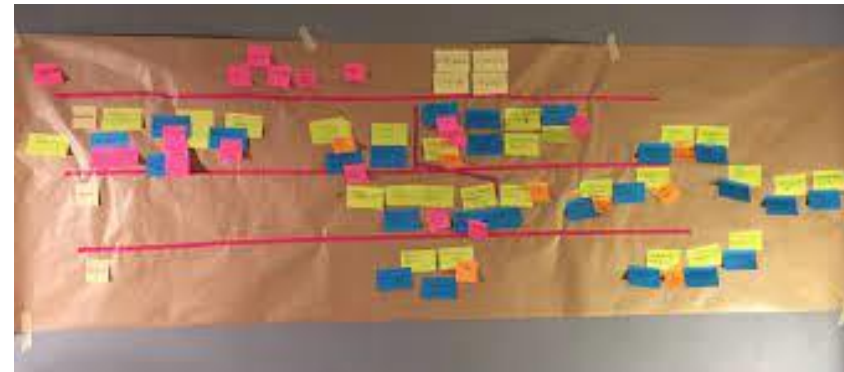
Isso não era “apenas uma aula”. Havia 50-70 pessoas incríveis na sala e eles **adoraram o EventStorming e começaram a experimentar em seus locais de trabalho**, em suas comunidades locais e fornecendo feedback, ideias e variações incrivelmente valiosos.



PREPARAÇÃO

UMA SUPERFÍCIE DE MODELAGEM ILIMITADA

Uma parede suficientemente extensa para desenharmos uma linha do tempo
(no modelo remoto ferramentas de board como o Miro)





PREPARAÇÃO

MENTES AGUÇADAS

O workshop reúne três tipos de pessoas:

- **Pessoas com perguntas** : em geral são os arquitetos, desenvolvedores, testers, ux.
- **Pessoas com respostas**: especialistas no domínio, os que conhecem muito sobre o assunto em questão, pos, clientes
- **O facilitador**: scrum master, agile coach, arquiteto, etc é a pessoa que vai moderar o workshop, sendo responsável por explicar a metodologia e conduzir o encontro

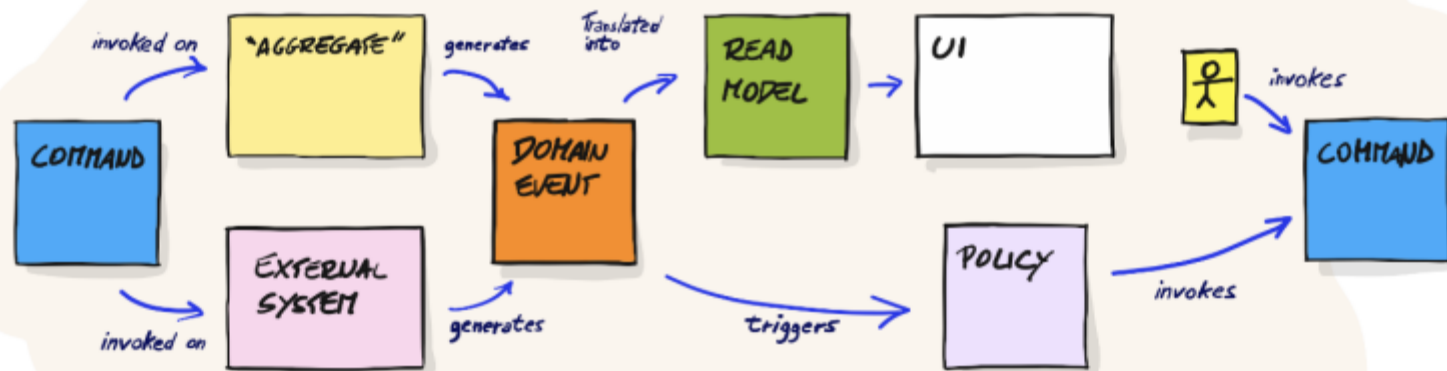
PREPARAÇÃO

O PROBLEMA A SER EXPLORADO

Qual domínio precisamos entender, modelar ou qual problema desejamos resolver?

É importante nos certificarmos do escopo que vamos abranger.

Pode ser necessário e até recomendado, segmentar em processos menores, em sessões incrementais divididas por áreas de conhecimento





TIPOS

BIG PICTURE

Obter a visão do fluxo de eventos

PROCESS LEVEL

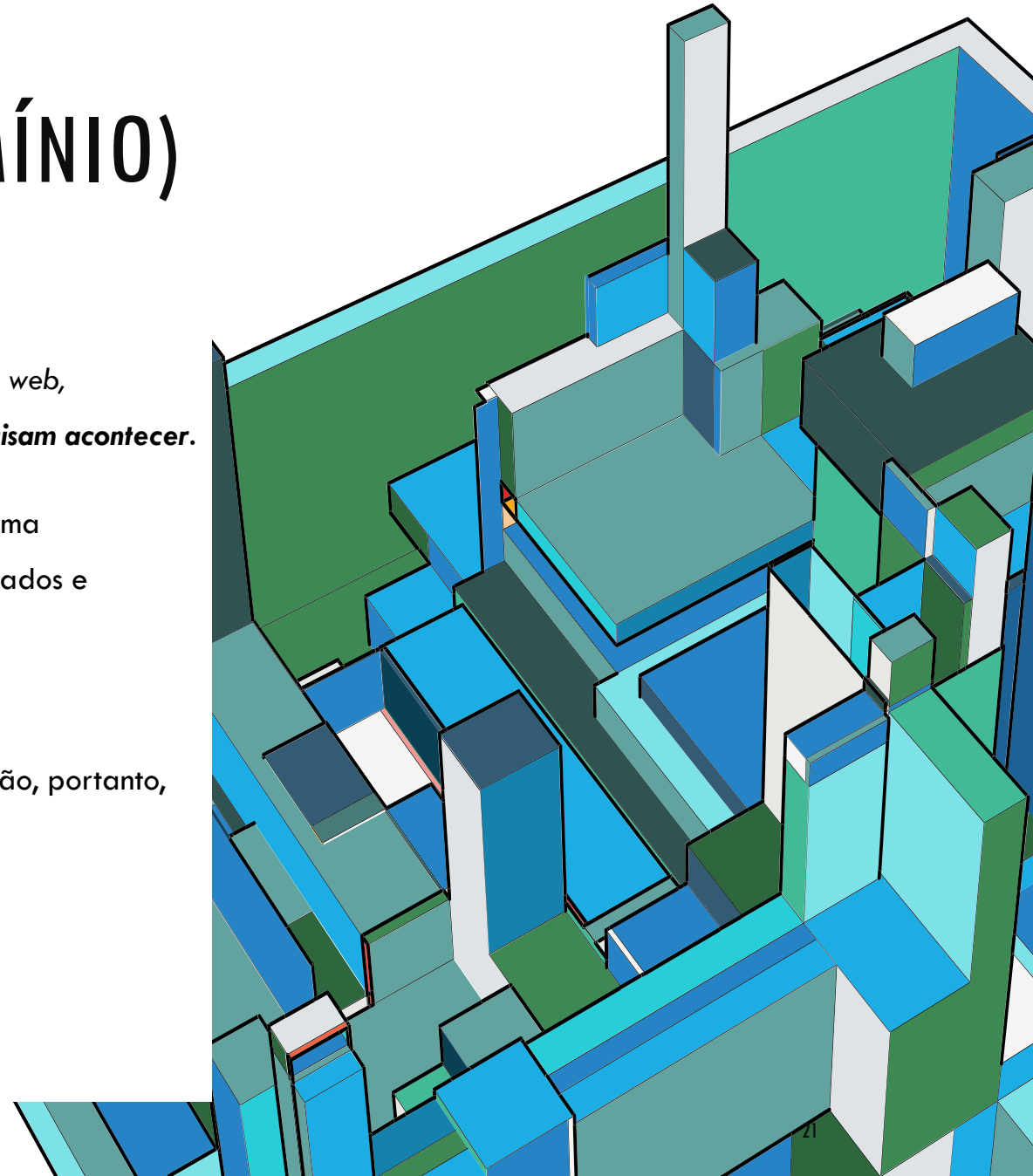
Identificação das causas

SOFTWARE DESIGN LEVEL

Modelagem dos agregados, contextos e possíveis microserviços

DOMAIN EVENT (EVENTO DE DOMÍNIO)

- É tudo que acontece e é do interesse de um especialista em domínio.
- O especialista em domínio não está interessado em bancos de dados, comunicação web, arquitetura ou padrões de design, mas **no domínio de negócios das coisas que precisam acontecer.**
- Os eventos de domínio capturam os fatos de uma maneira que não especifica uma implementação particular. Embora existam inúmeras opções para representar dados e implementar comportamentos, **não há alternativas para eventos de domínio.**
- Como os eventos de domínio representam fatos sobre o domínio, eles só mudam significativamente quando o negócio subjacente muda. Os eventos de domínio são, portanto, **uma estrutura mais estável e resiliente para seu modelo.**



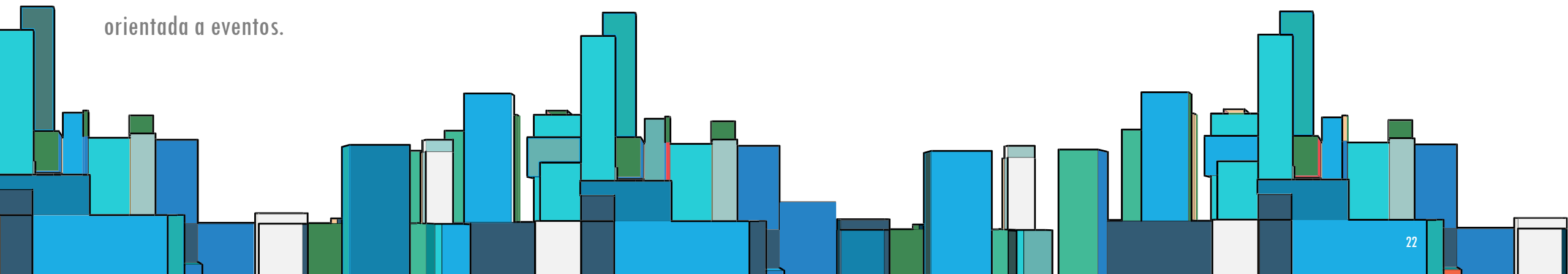
EVENT STORMING & EVENT DRIVEN ARCHITECTURES

EVENT BASED

Ao enriquecermos nosso modelo com comandos e sistemas externos vamos tendo uma visão mais clara de quão distribuída precisa ser nossa arquitetura, com qual complexidade precisamos lidar ou seja, conseguimos transitar do domínio do problema adentrando no domínio da solução em uma mesma visualização.

A segunda razão é que para projetar software Event-Driven e/ou Event-Sourced com altíssima eficiência precisamos minimizar a incompatibilidade de impedância entre a narrativa de negócios e a implementação de software.

As partes interessadas podem melhorar o modelo juntas e chegar muito mais rapidamente em uma solução realmente aderente a uma arquitetura orientada a eventos.



EVENTOS INTERNOS X EXTERNOS

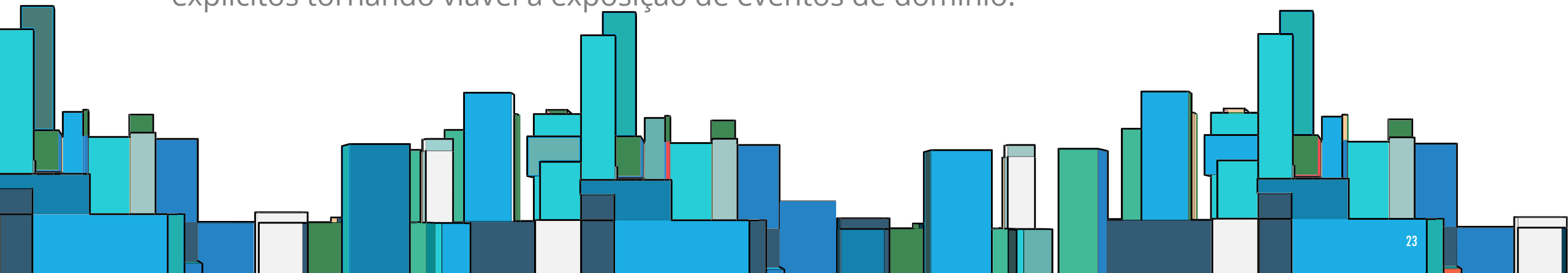
Geralmente, os eventos de domínio serão chamados de "eventos internos" porque não saem de seus limites.

Enquanto os Eventos de Integração são chamados de "Eventos Externos" porque sua intenção é sair de seus limites.

Existe uma recomendação de não expor eventos de domínio fora do seu limite porém se torna aceitável para processo de negócios de longa duração (long running process) e coreografia entre serviços caso atenda a esses 3 requisitos:

Estabilidade, Compreensão e Requisitos do Consumidor

É aí que a modelagem de Event Storming consegue ajudar, tornando os conceitos de negócio estáveis, melhorando a compreensão e deixando os requisitos de quem vai consumir mais explícitos tornando viável a exposição de eventos de domínio.





LINKS ÚTEIS

- ✓ <https://www.eventstorming.com/resources/>
- ✓ <https://github.com/mariuszgil/awesome-eventstorming>
- ✓ <https://docs.microsoft.com/en-us/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/domain-events-design-implementation>
- ✓ <https://docs.microsoft.com/en-us/azure/architecture/guide/architecture-styles/event-driven>
- ✓ <https://devblogs.microsoft.com/cesardelatorre/domain-events-vs-integration-events-in-domain-driven-design-and-microservices-architectures/>
<https://github.com/TeteStorm/OrderMS>

VAMOS PRATICAR?

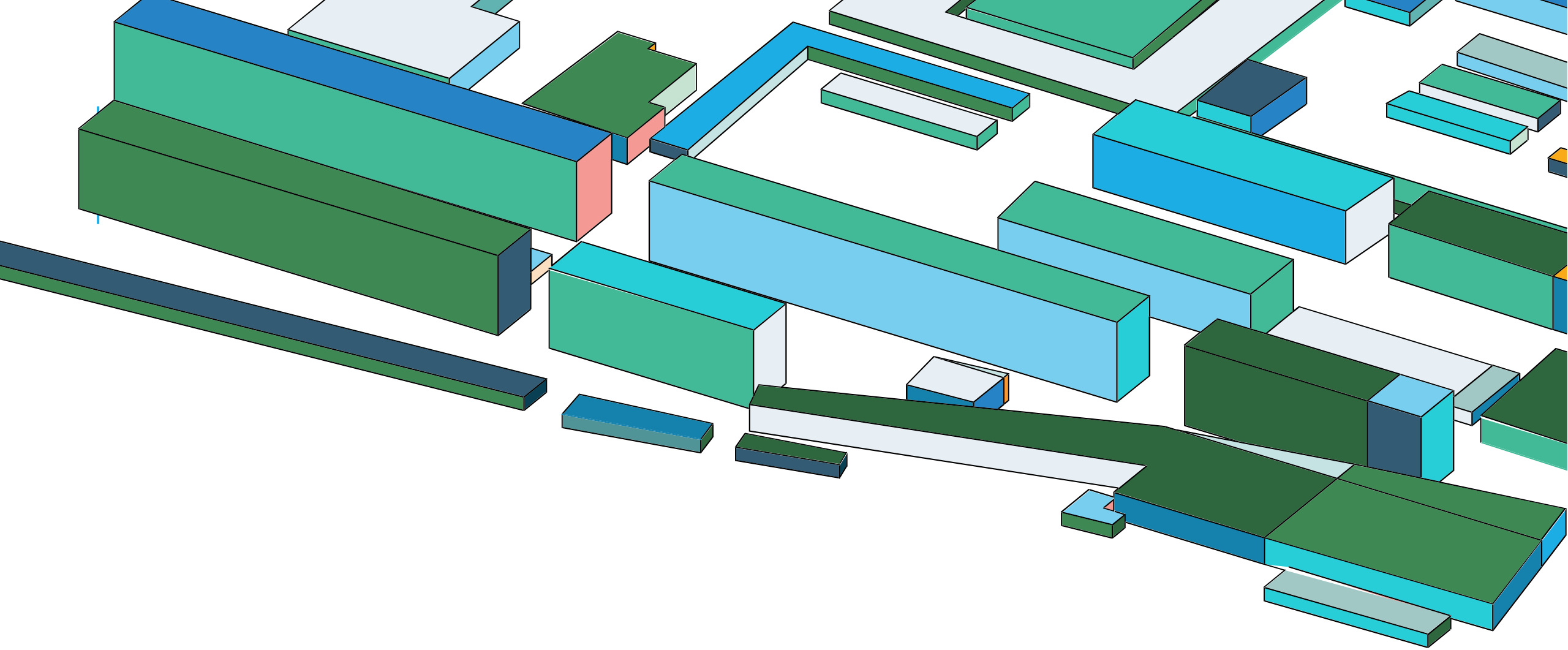


Apresentação no Miro

[https://miro.com/app/board/o9J_lIBEmFk
=?share_link_id=843288462493](https://miro.com/app/board/o9J_lIBEmFk=/?share_link_id=843288462493)

Exemplos de Códigos

[https://github.com/TeteStorm/Event-Storming-
and-Domain-Events-Sample-Concepts](https://github.com/TeteStorm/Event-Storming-and-Domain-Events-Sample-Concepts)



THANKS FOR ALL!