Domain Driven Design (DDD) - A Technical Perspective On Software Platform Strategy

by Luke Gordon

https://by.dialexa.com/domain-driven-design-ddd-a-technical-perspective-on-software-platform-strategy

DIALEXA

DIALEXA

Dialexa

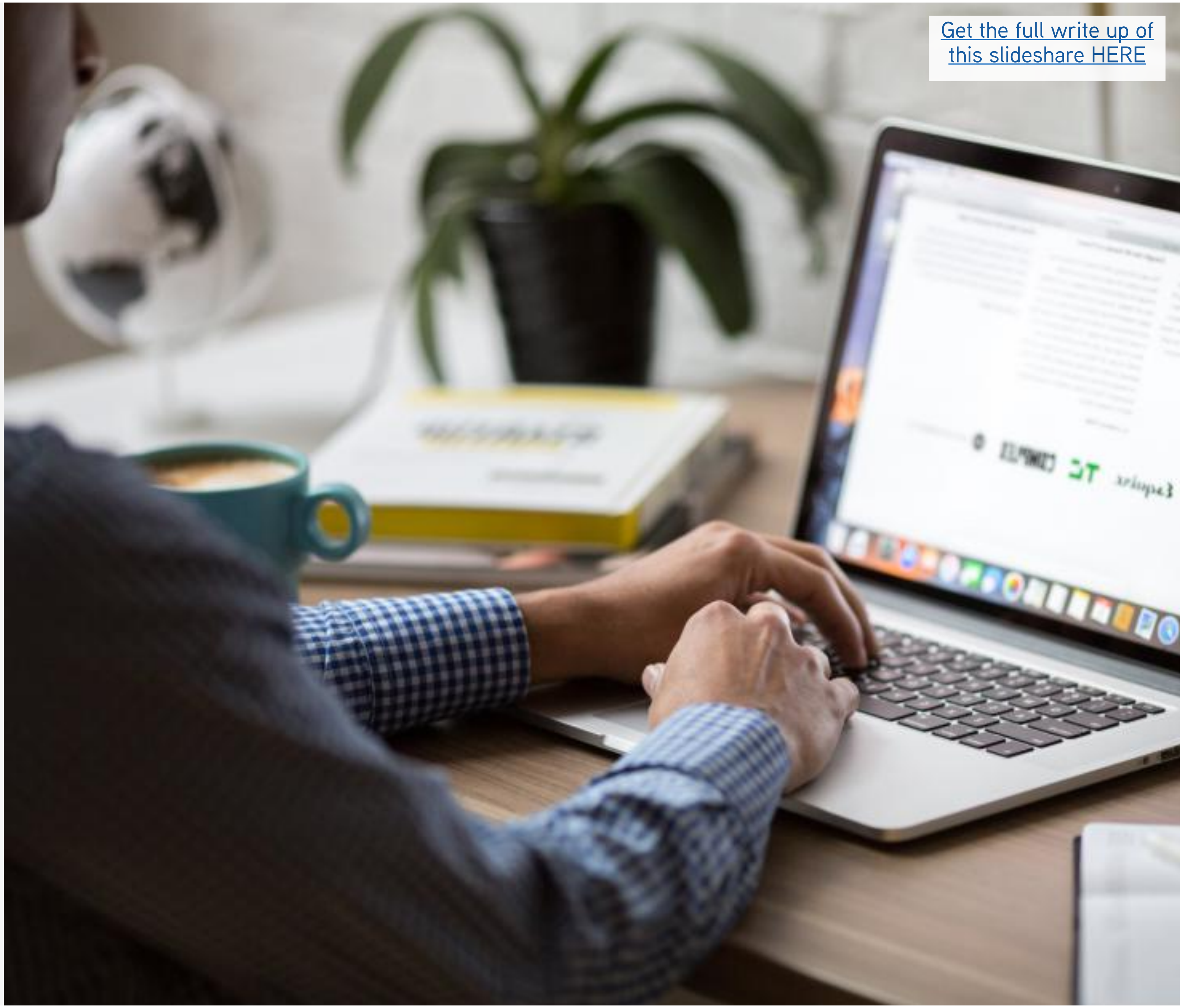
We are on a mission to make every company a great technology company.

We work with organizations to define and execute digital transformation strategies to improve business operations and customer experiences. Our services include:

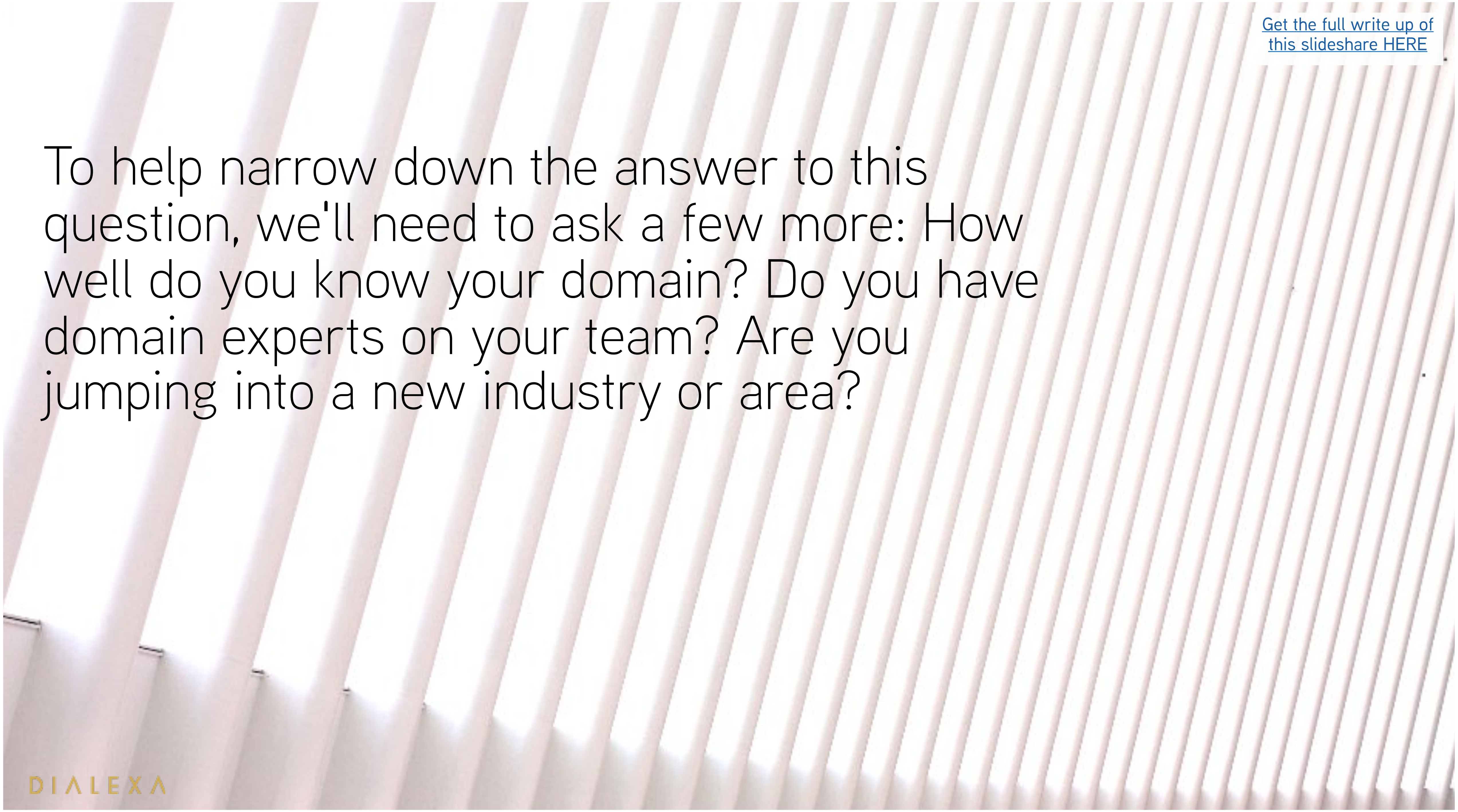
- Multi-Year Technology Roadmap
- Platform Engineering
- User Experience Design
- Custom Software Development
- Hardware Prototyping /IoT

In our last post of this series we discussed the application continuum and how it guides your software platform strategy. Understanding how your solution evolves over time opens your mind to various starting points, but we still haven't answered the main domain boundary question, "Where do I start?"

DIALEXA



Get the full write up of this slideshare HERE



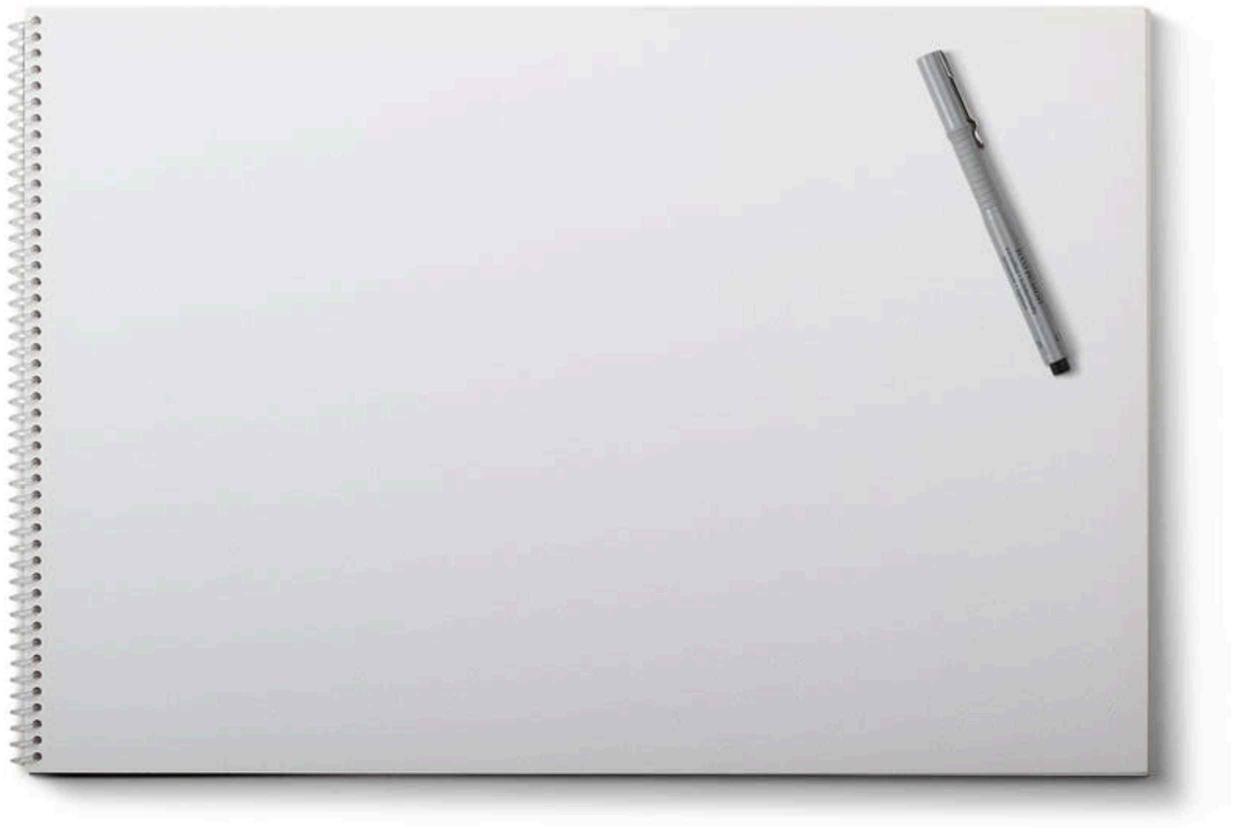
Get the full write up of this slideshare HERE

To help narrow down the answer to this question, we'll need to ask a few more: How well do you know your domain? Do you have domain experts on your team? Are you jumping into a new industry or area?

DIALEXA

Get the full write up of this slideshare HERE

If you're lucky enough to have domain experts on your team with a vast knowledge of the industry or area your platform is targeting, there's a great chance you can leverage Domain Driven Design.



DIALEXA

Get the full write up of this slideshare HERE

Domain Driven Design (DDD), a term coined by Eric Evans, "provides a structure of practices and terminology for making design decisions that focus and accelerate software platform strategy projects dealing with complicated domains".

DIALEXA

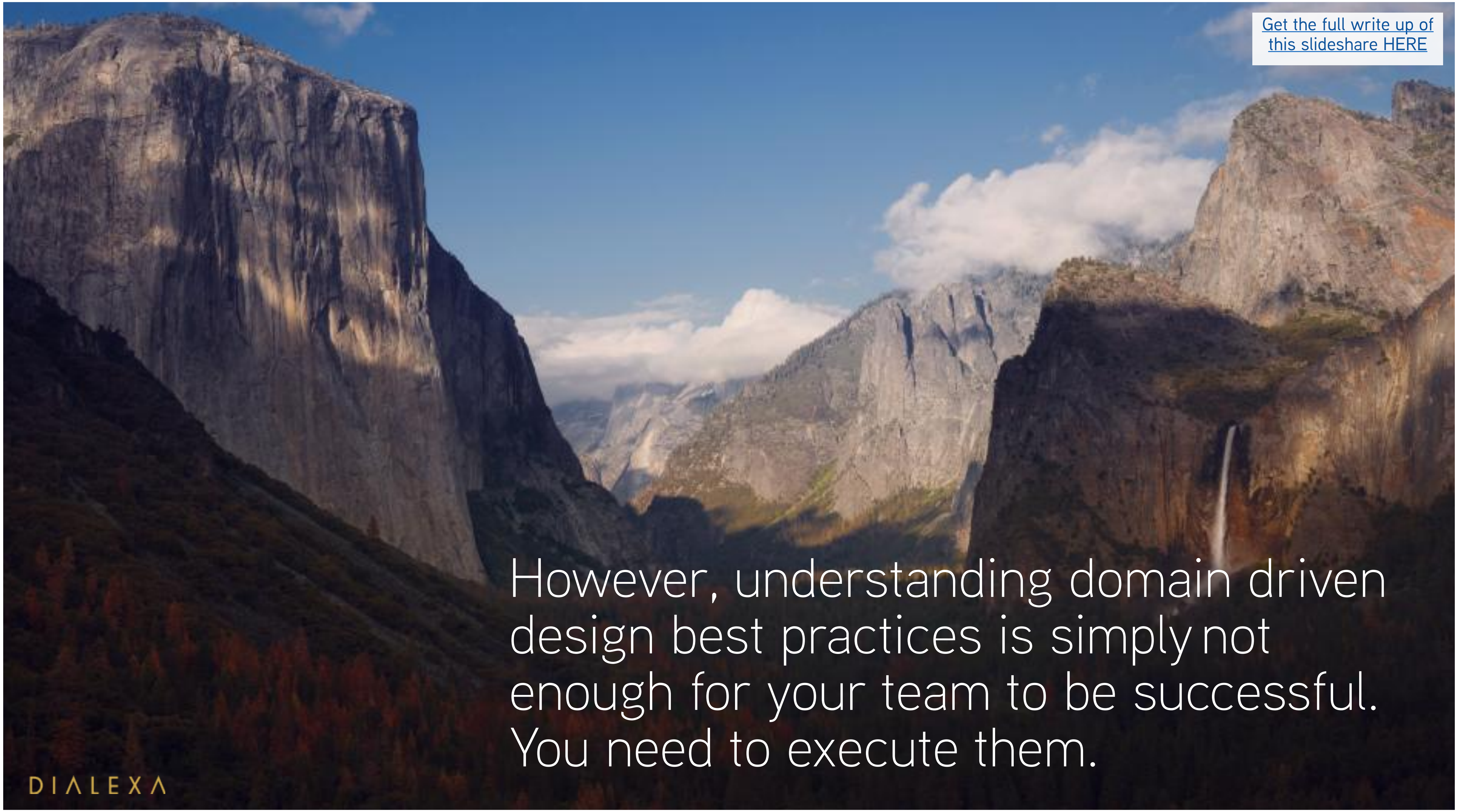
To read more you can find the full article at https://by.dialexa.com/domain-driven-design-ddd-a-technical-perspective-on-software-platform-strategy

DIALEXA

Get the full write up of this slideshare HERE

Domain Driven Design Best Practices

DIALEXA



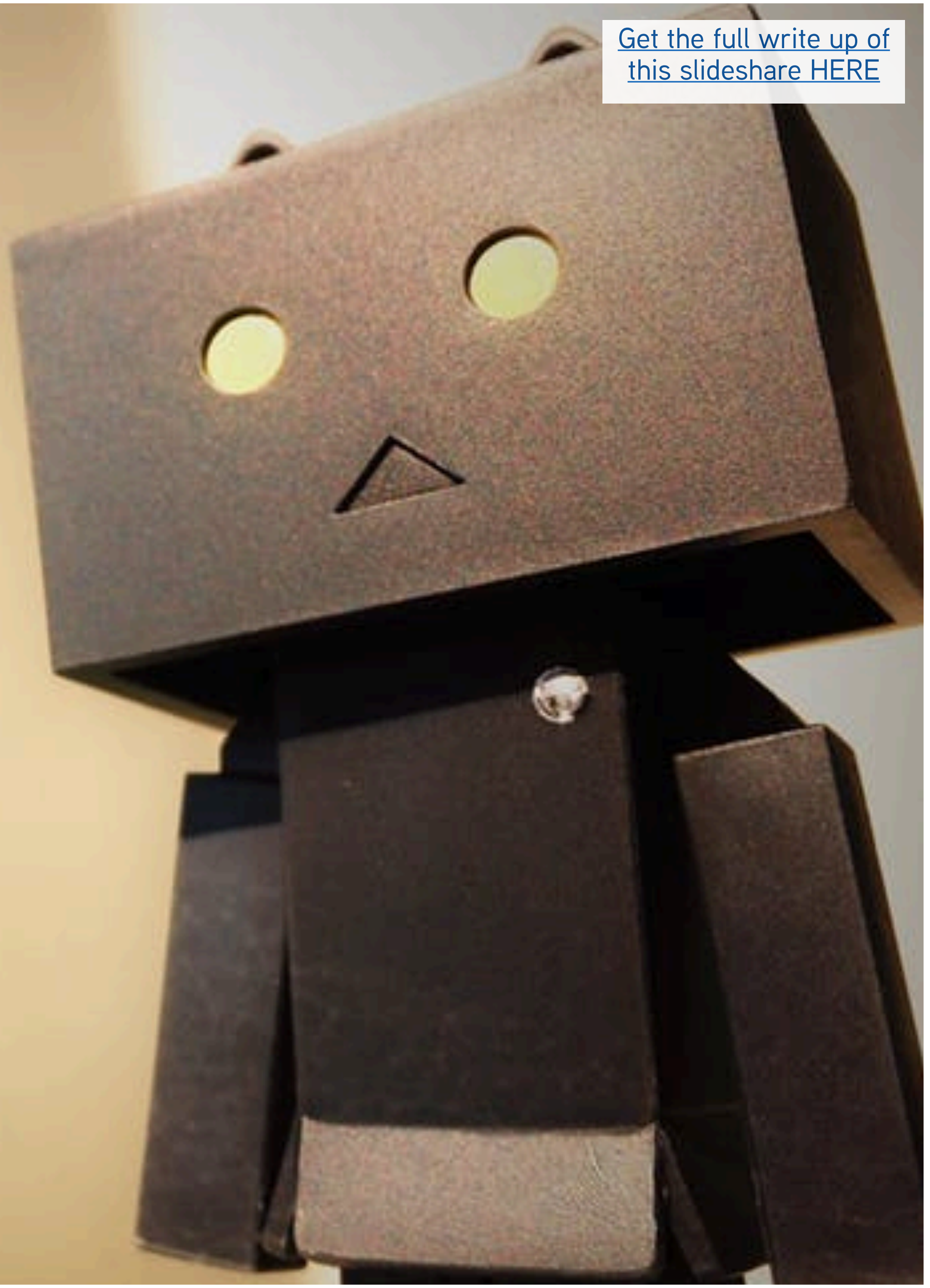
Get the full write up of this slideshare HERE

However, understanding domain driven design best practices is simply not enough for your team to be successful. You need to execute them.

DIALEXA

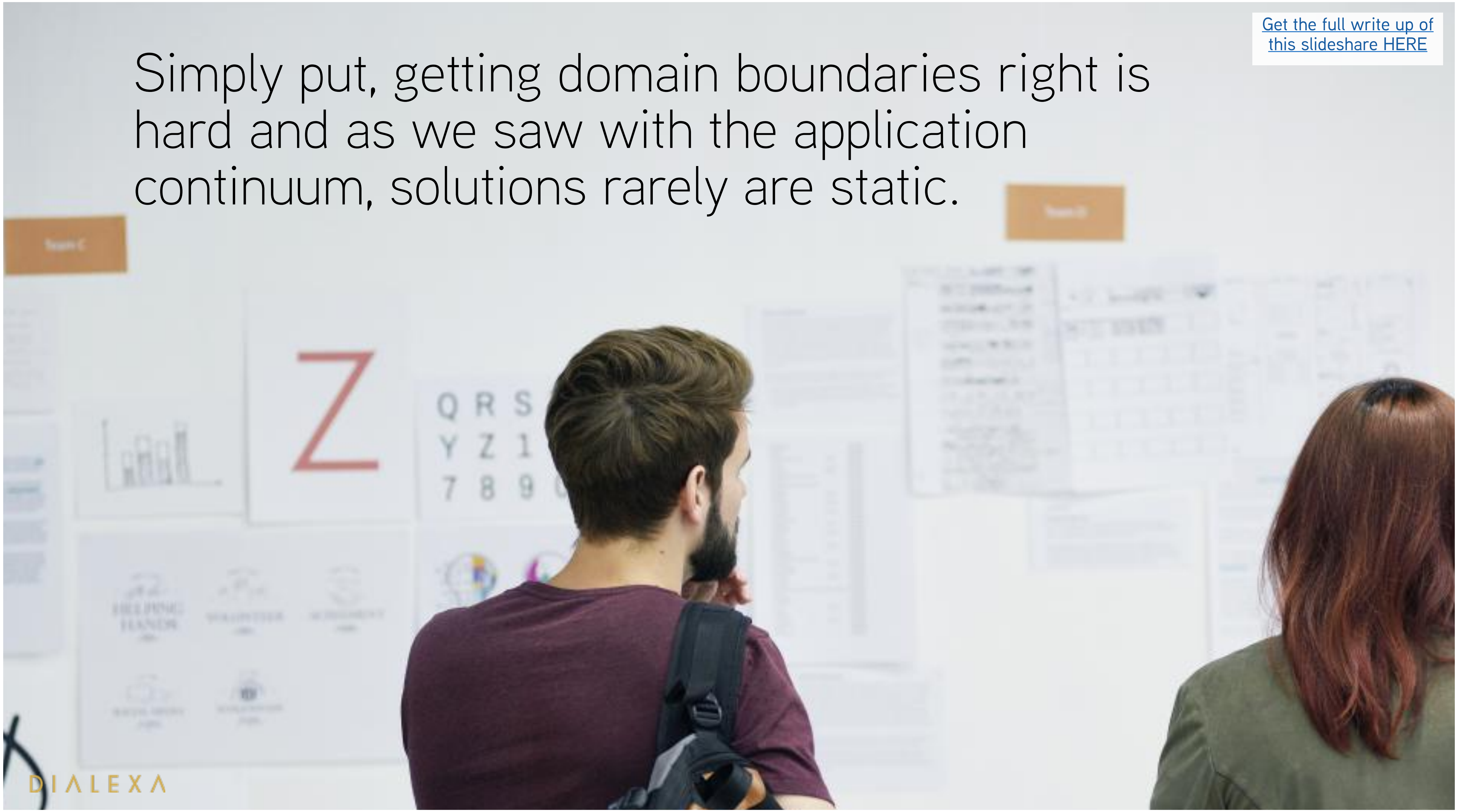
Get the full write up of this slideshare HERE

Eric Evans himself points out in a talk at the GOTO Conference, how time has shown even the best of teams struggle with discipline using Domain Driven Design on a monolithic scale. In his talk, he highlights how microservices helps drive these best practices by forcing teams to deal with domain boundaries.



Get the full write up of this slideshare HERE

All of that being said, if you feel so far microservices are the right fit for you, keep in mind a microservice approach might still not be your best option to start with. As Martin Fowler explains, "even experienced architects working in familiar domains have great difficulty getting boundaries right at the beginning."



Get the full write up of this slideshare HERE

Simply put, getting domain boundaries right is hard and as we saw with the application continuum, solutions rarely are static.

DIALEXA

Even if you do define the domain boundaries right at the beginning, they may very well change over time. A monolithic approach at the start enables you to pivot faster to changing business requirements.

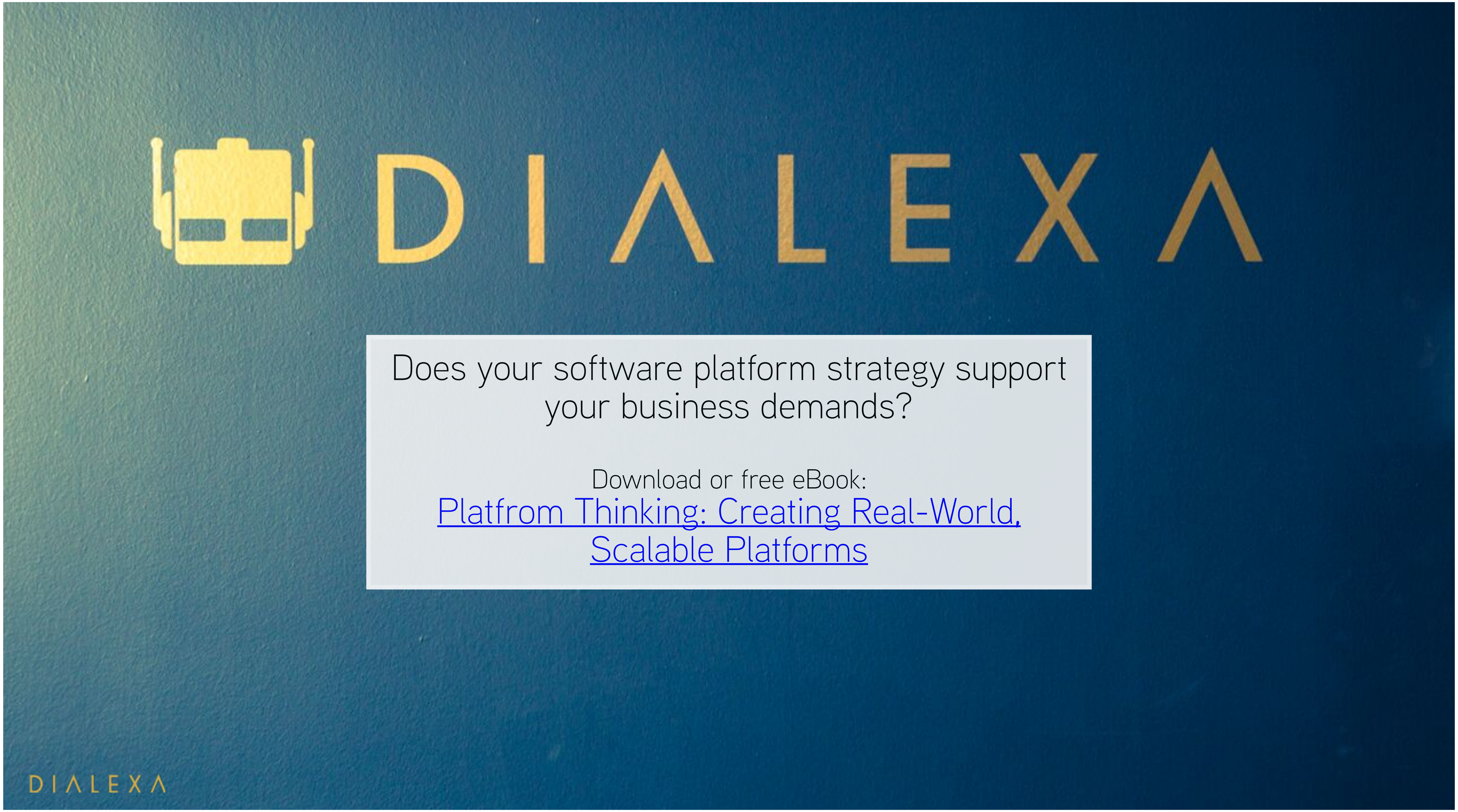
DIALEXA

Get the full write up of this slideshare HERE

Get the full write up of this slideshare HERE

Once you've solidified your architecture you can always break it apart into microservices. As Simon Brown has stated at various talks: "If you can't build a monolith, what makes you think microservices are the answer?"

DIALEXA



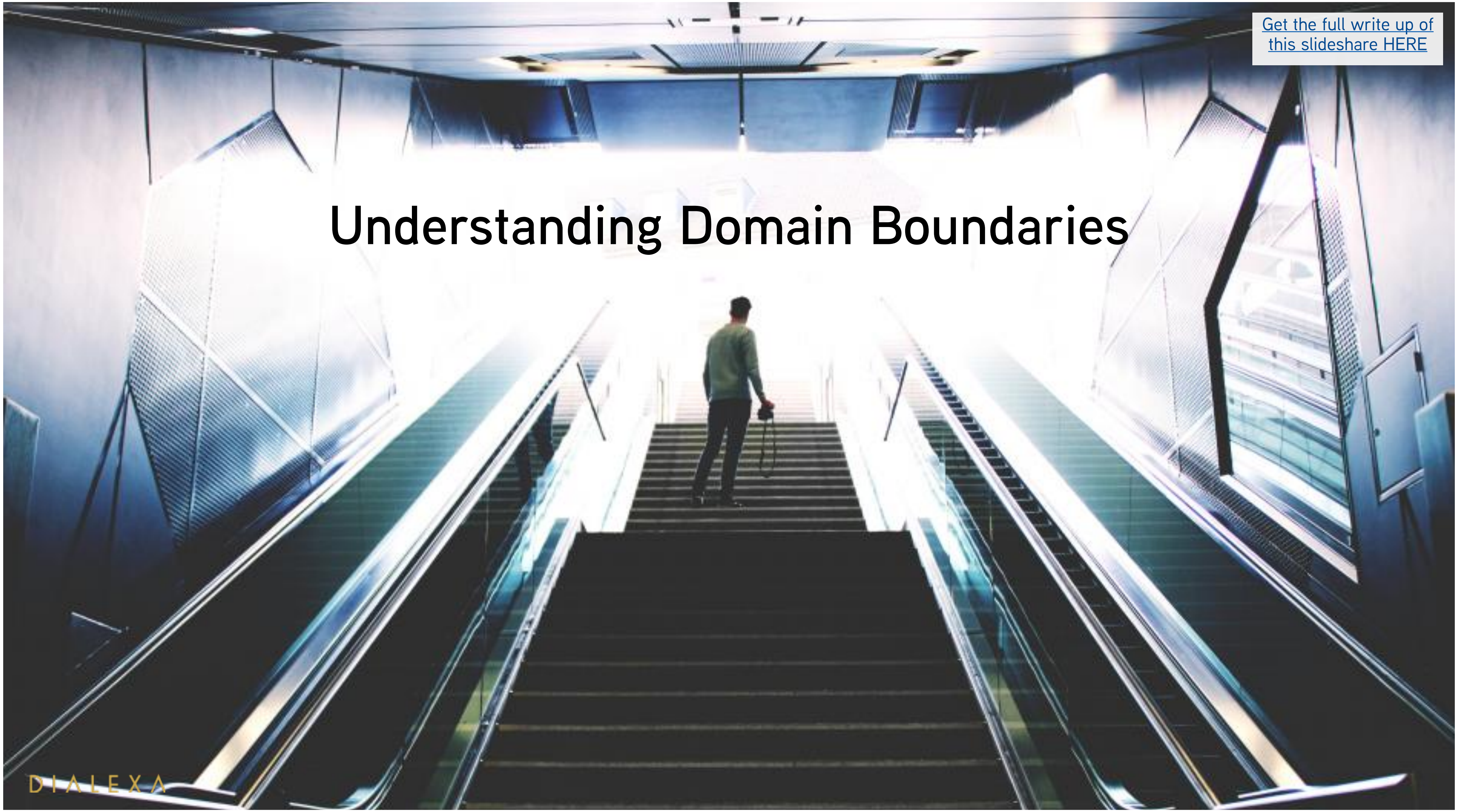
Does your software platform strategy support your business demands?

Download or free eBook:

Platfrom Thinking: Creating Real-World, Scalable Platforms

Get the full write up of this slideshare HERE

Understanding Domain Boundaries



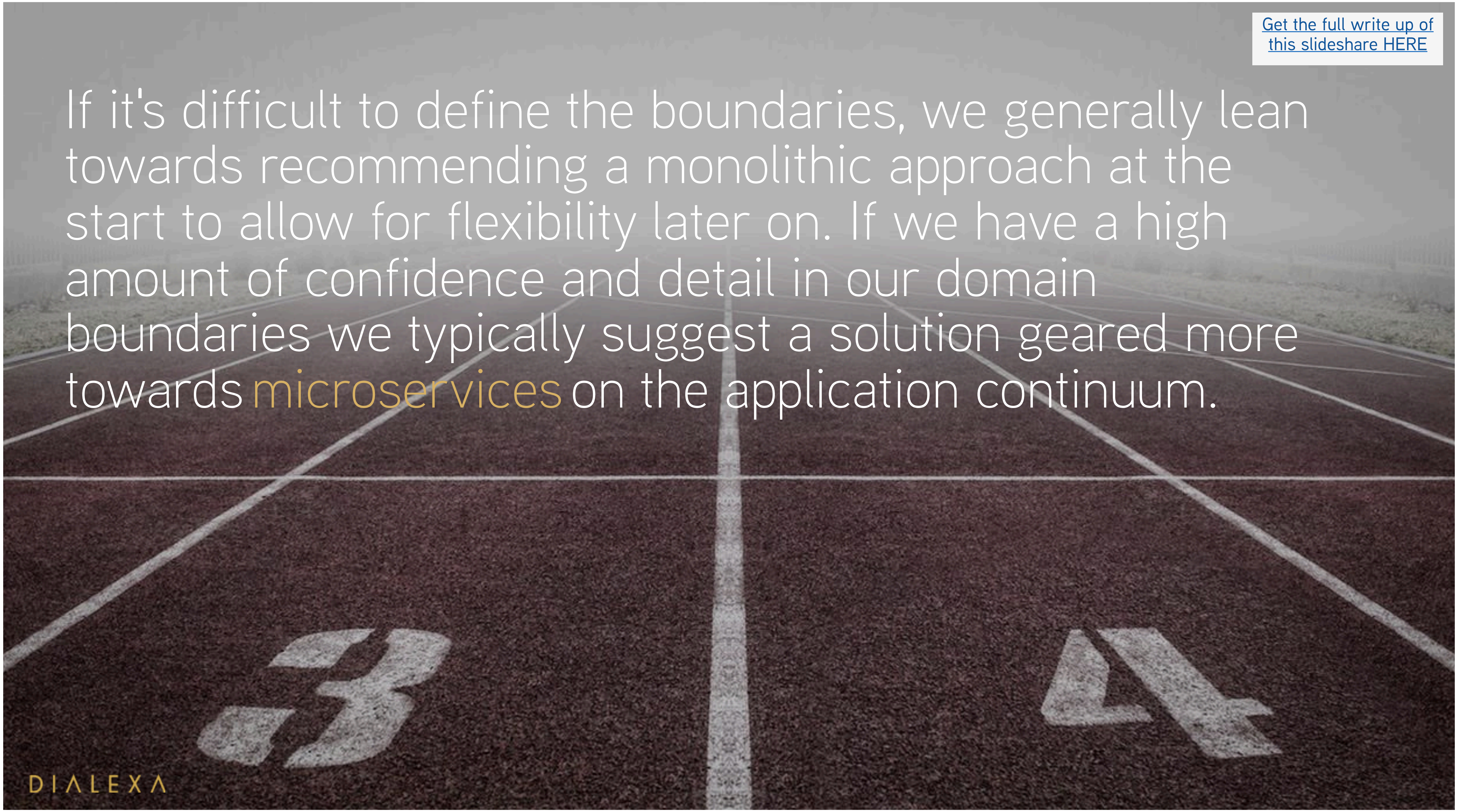
DIALEXA

Get the full write up of this slideshare HERE

Still not sure where to start when choosing between a microservices vs monolithic approach? At Dialexa we go through various activities with clients to help understand their domain boundaries and how well we can define them.

DIALEXA

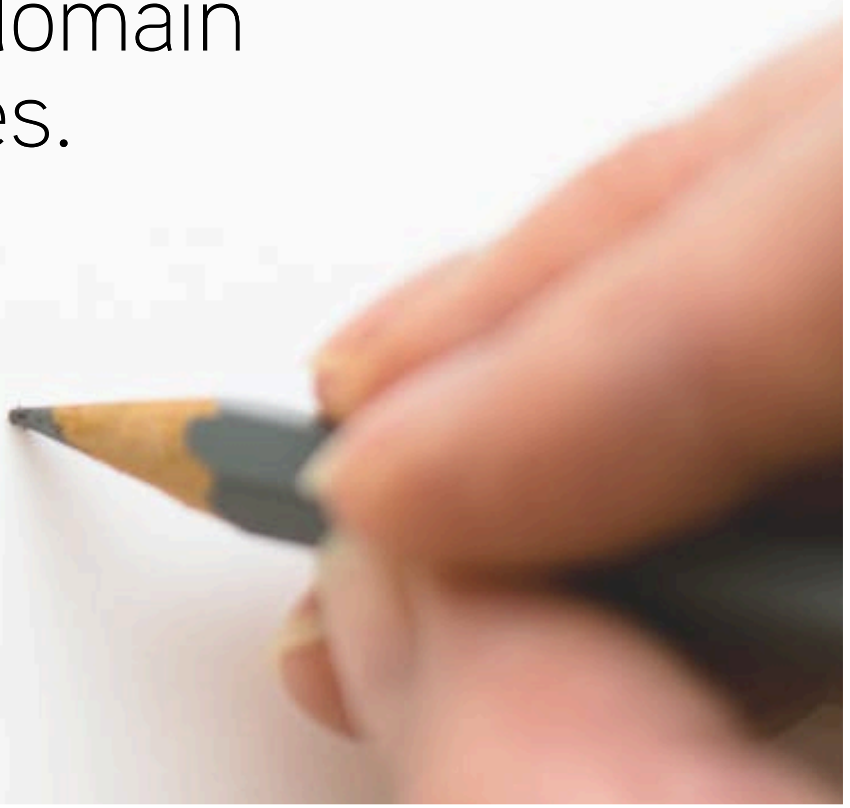
Get the full write up of this slideshare HERE



If it's difficult to define the boundaries, we generally lean towards recommending a monolithic approach at the start to allow for flexibility later on. If we have a high amount of confidence and detail in our domain boundaries we typically suggest a solution geared more towards microservices on the application continuum.

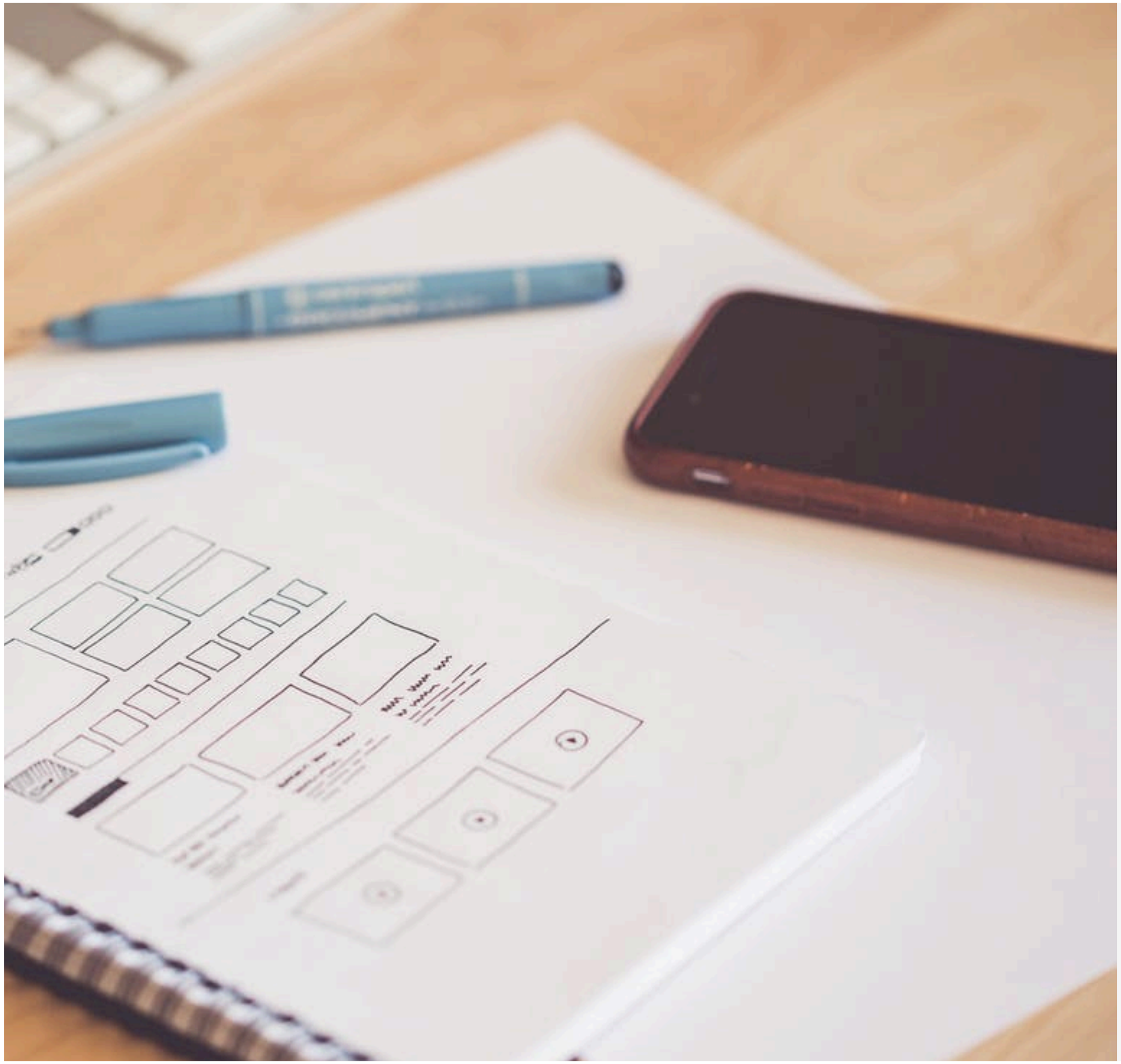
DIALEXA

Get the full write up of this slideshare HERE



One simple activity we use only requires pencil and paper. Ask various team members to define the domain boundaries for your platform by themselves.

DIALEXA



Get the full write up of this slideshare HERE

After they're done compare the results. How similar are they? Are there significant differences? Are some areas harder to define? While this activity seems obvious and simple, it highlights your team's current understanding of the problems your platform solves.

Get the full write up of this slideshare HERE

Armed with this knowledge you can make a better decision on where your team should start.

DIALEXA

Thank You

Doug Platts

VP of Marketing

marketing@dialexa.com

If you are looking to undertake a digital transformation initiative, contact Dialexa today and see how we can help make your company a great technology company.

Get the full write up of this slideshare at https://by.dialexa.com/domain-driven-design-ddd-a-technical-perspective-on-software-platform-strategy

DIALEXA