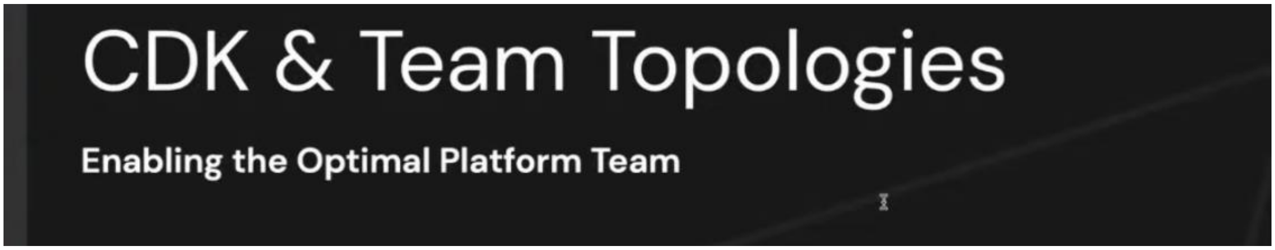
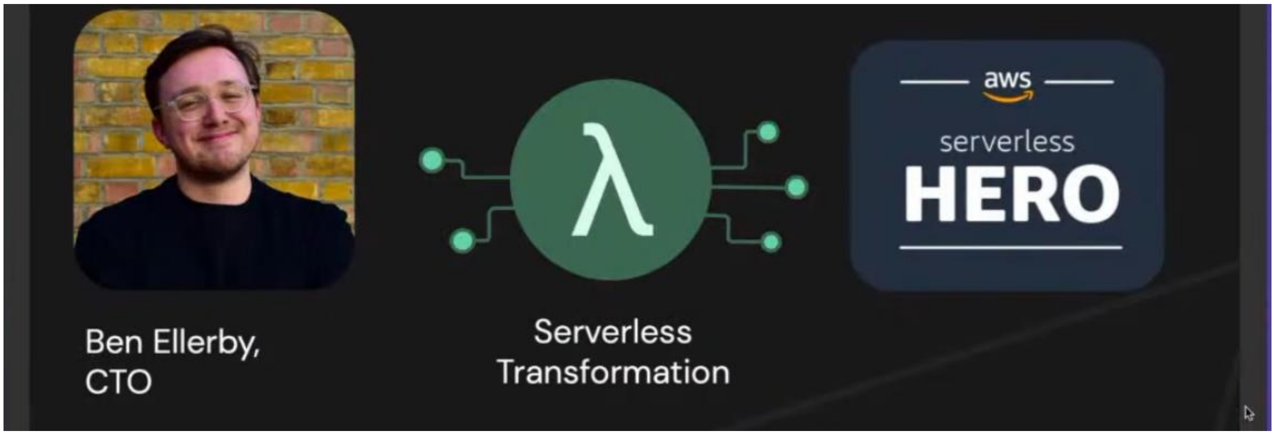


CDK & Team Topologies

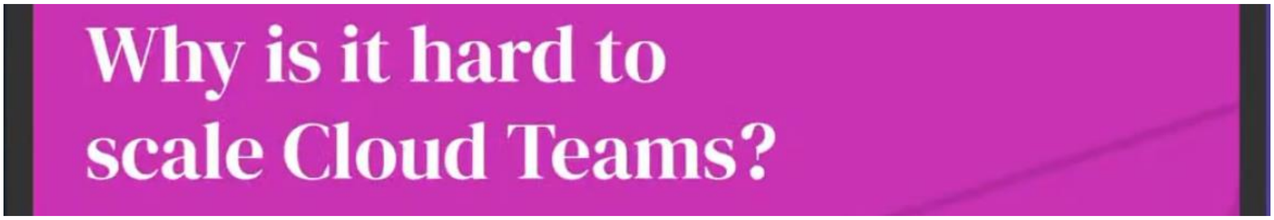
Enabling the Optimal Platform Team

How we enable the optimal platform teams

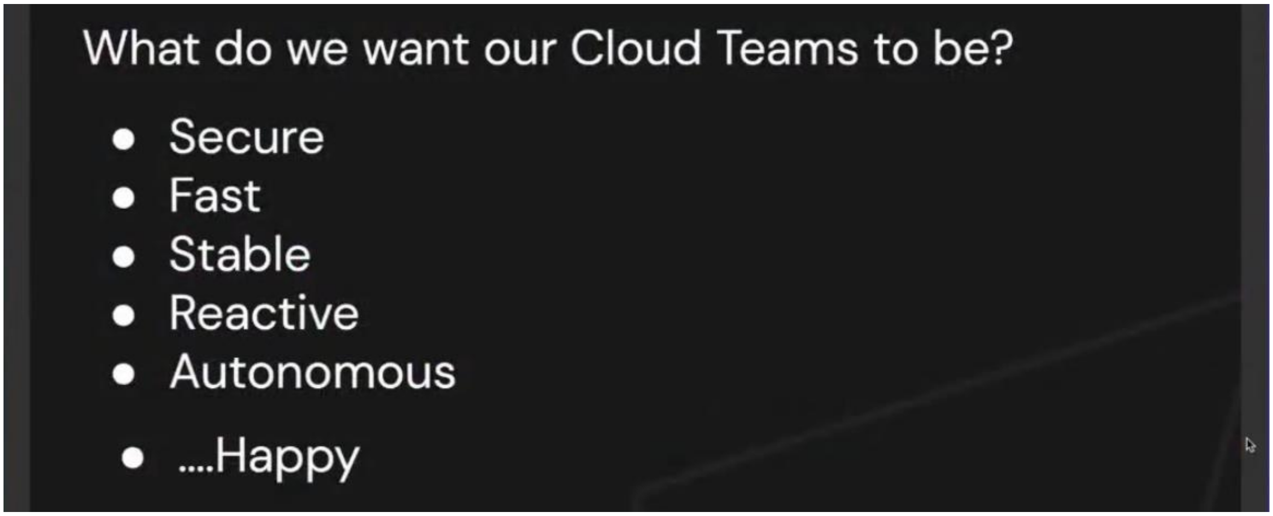


Ben Ellerby,
CTO

Serverless
Transformation

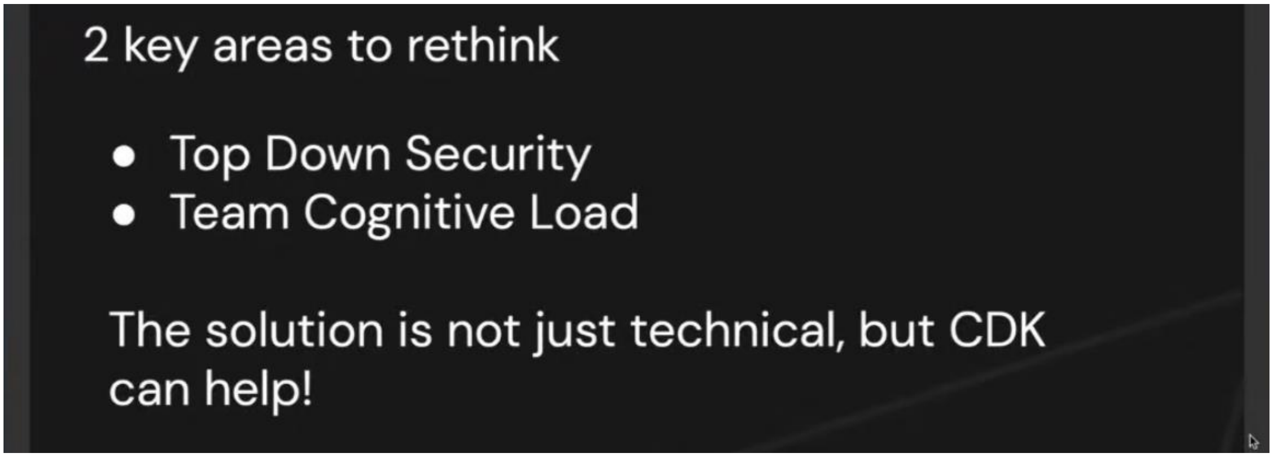


Why is it hard to scale Cloud Teams?



What do we want our Cloud Teams to be?

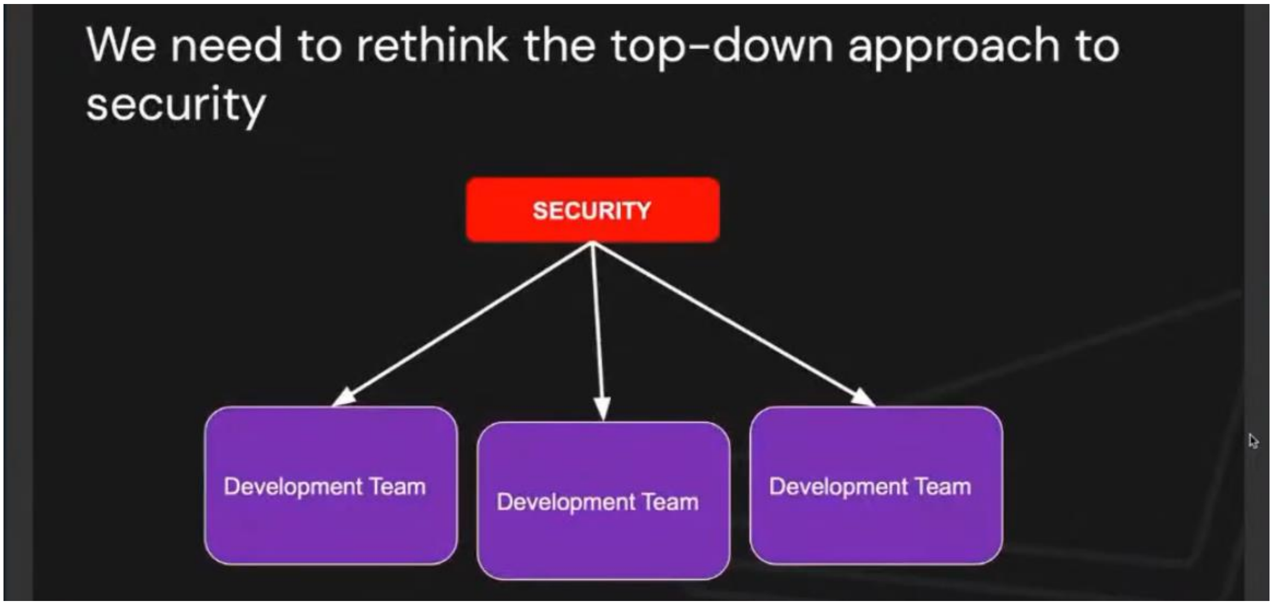
- Secure
- Fast
- Stable
- Reactive
- Autonomous
- ….Happy



2 key areas to rethink

- Top Down Security
- Team Cognitive Load

The solution is not just technical, but CDK can help!



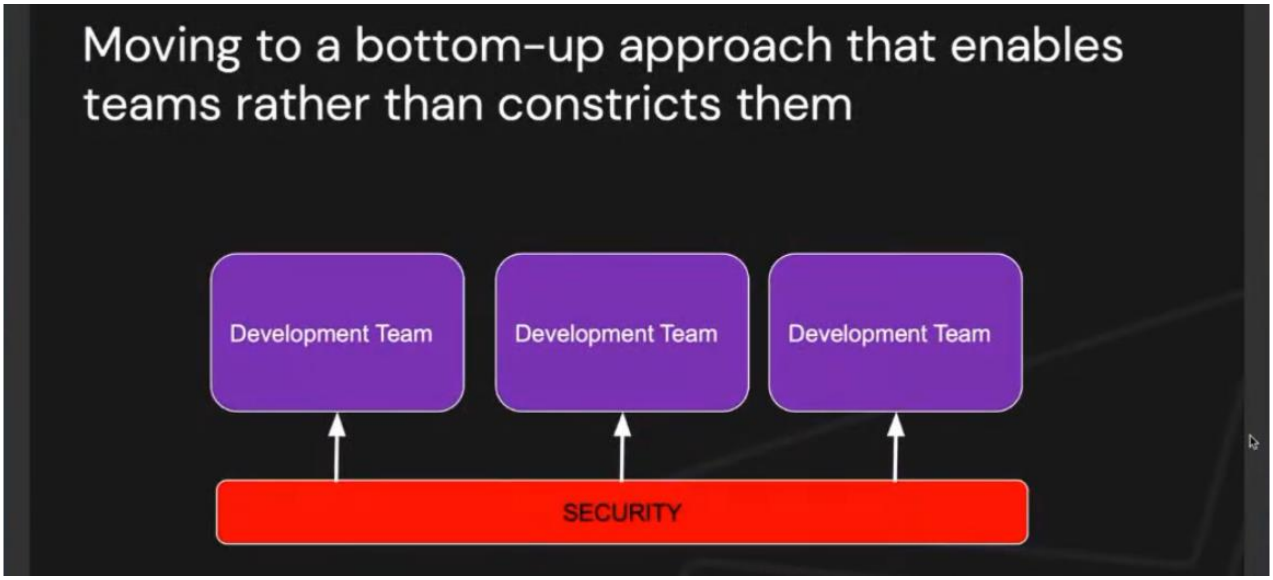
We need to rethink the top-down approach to security

SECURITY

Development Team

Development Team

Development Team



Moving to a bottom-up approach that enables teams rather than constricts them

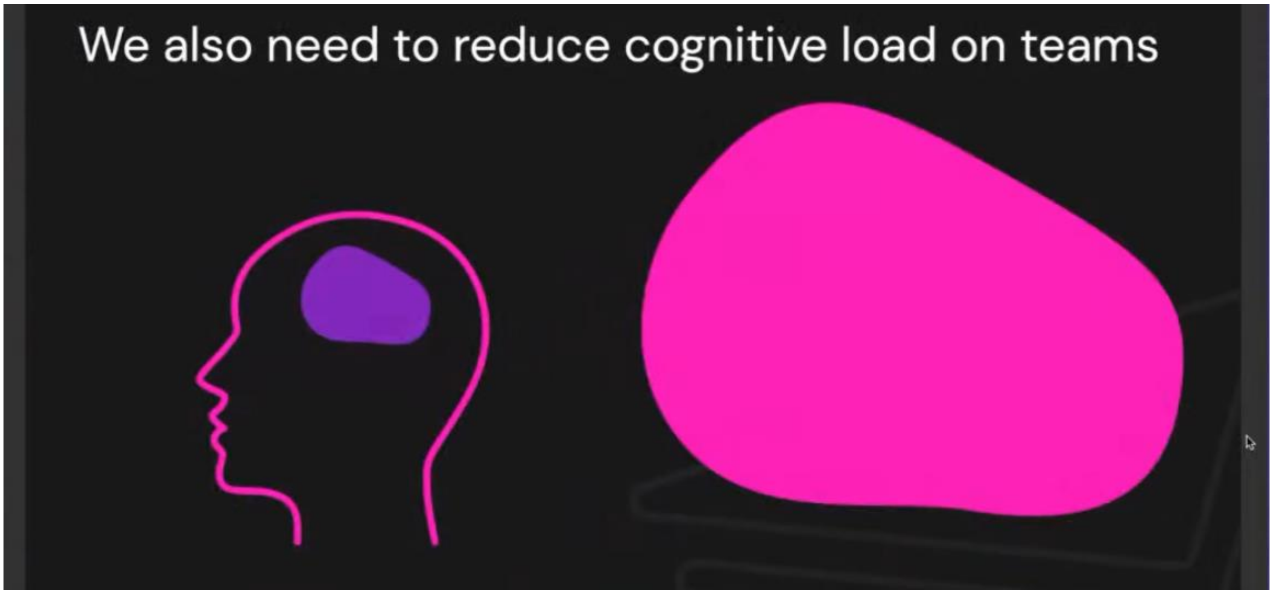
Development Team

Development Team

Development Team

SECURITY

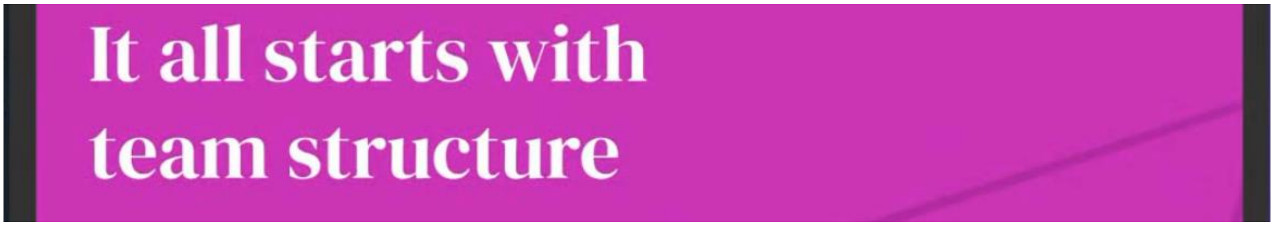
CDK allows building using abstractions and encapsulation, not console deployments and rulebooks.



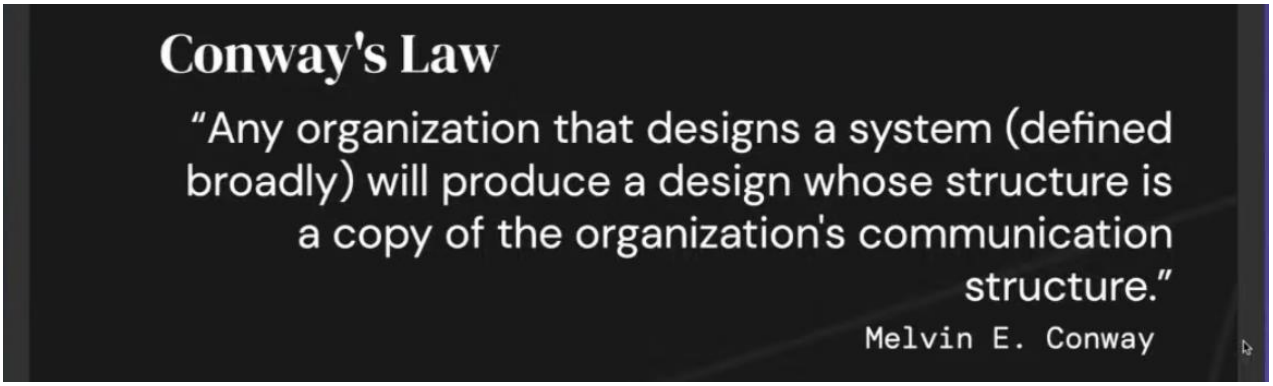
We also need to reduce cognitive load on teams

We can now have smaller teams working on separate smaller domains/business domains using CDK encapsulation for highly scalable event-driven architectures.

Which means we need to restructure



It all starts with team structure



Conway's Law

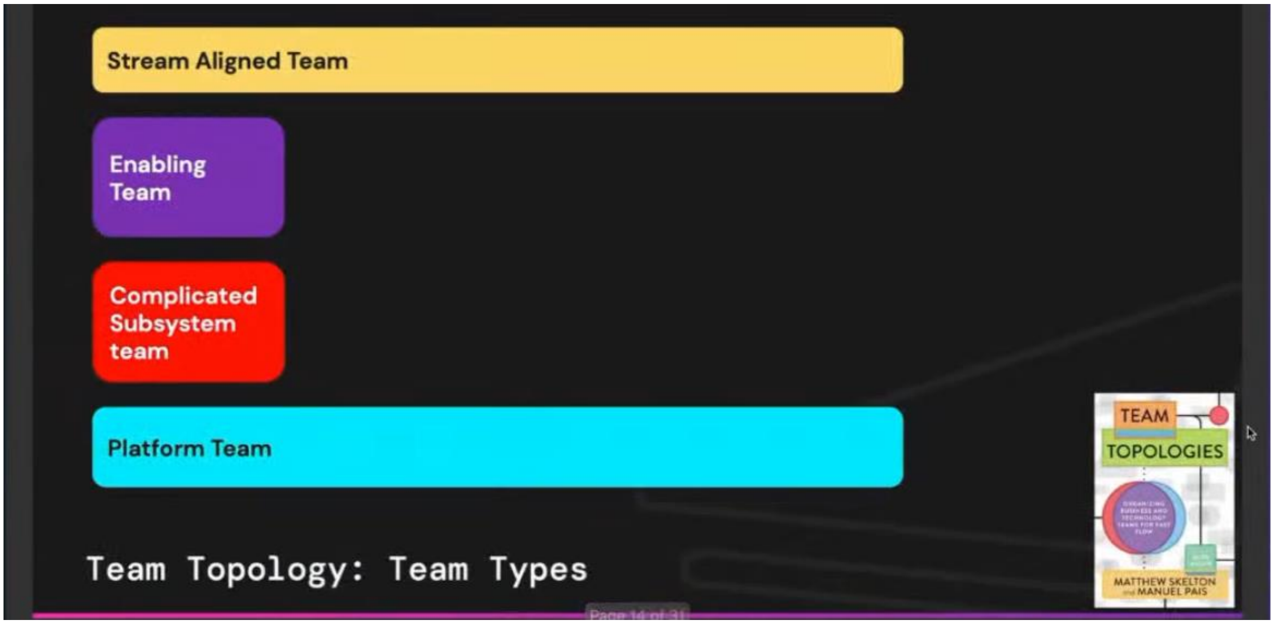
"Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure."

Melvin E. Conway

Read the book

A great framework for companies to organize for team flow

TEAM
TOPOLOGIES
ORGANIZING BUSINESS AND TECHNOLOGY TEAMS FOR FAST FLOW
Foreword by RUTH MALAN
MATTHEW SKELTON and MANUEL PAIS



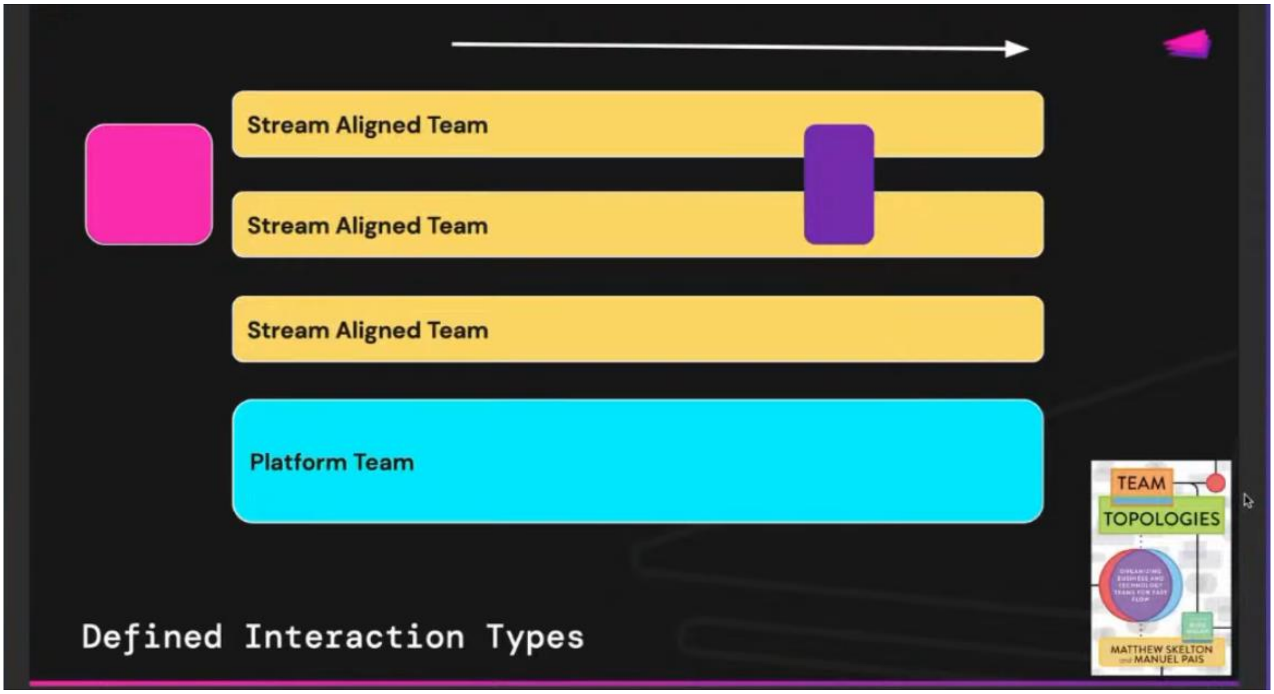
Stream Aligned Team

Enabling Team

Complicated Subsystem team

Platform Team

Team Topology: Team Types



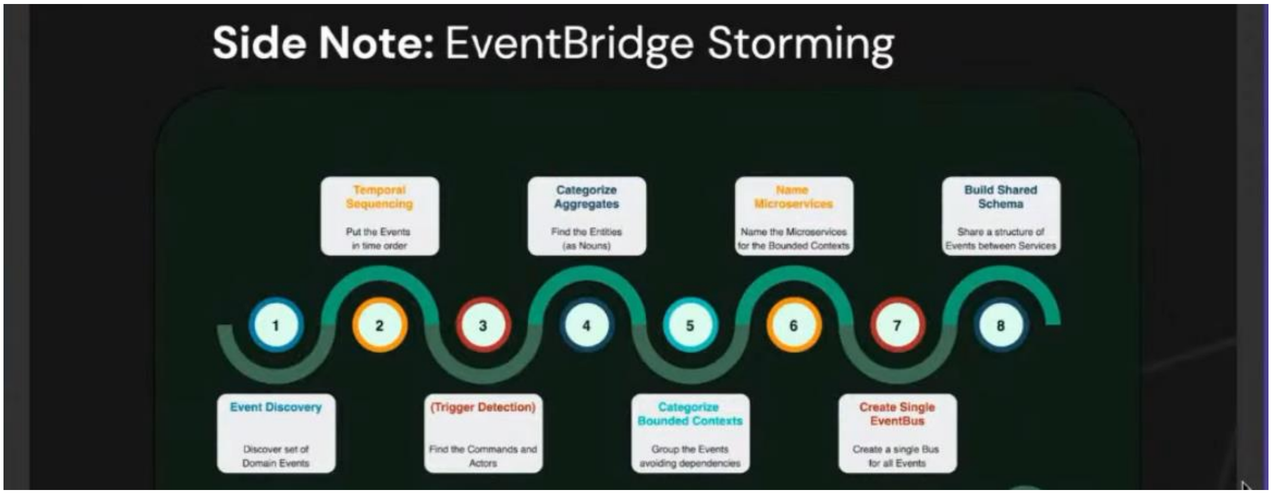
Stream Aligned Team

Stream Aligned Team

Stream Aligned Team

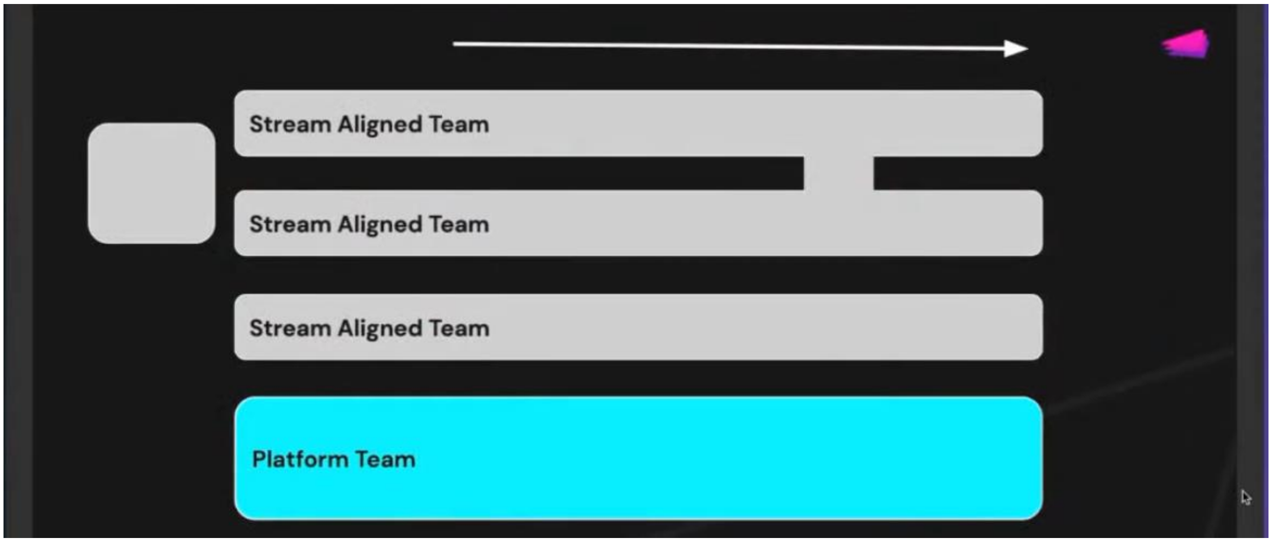
Platform Team

Defined Interaction Types



Side Note: EventBridge Storming

1. **Event Discovery** Discover set of Domain Events
2. **Temporal Sequencing** Put the Events in time order
3. **(Trigger Detection)** Find the Commands and Actors
4. **Categorize Aggregates** Find the Entities (as Nouns)
5. **Categorize Bounded Contexts** Group the Events avoiding dependencies
6. **Name Microservices** Name the Microservices for the Bounded Contexts
7. **Create Single EventBus** Create a single Bus for all Events
8. **Build Shared Schema** Share a structure of Events between Services



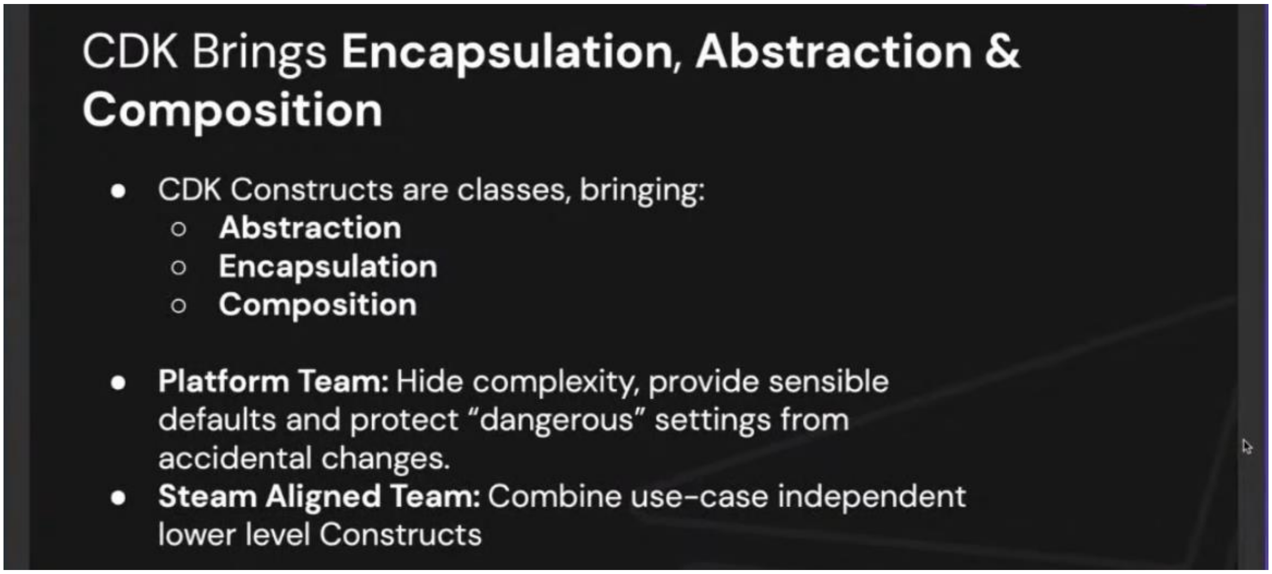
Stream Aligned Team

Stream Aligned Team

Stream Aligned Team

Platform Team

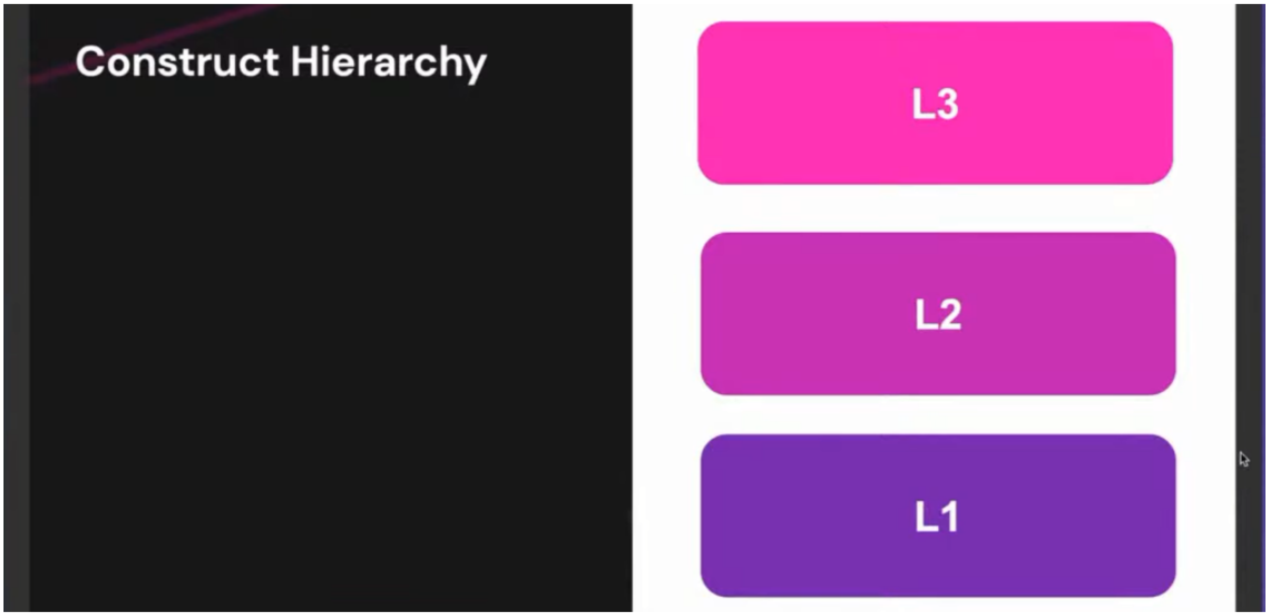
Enabled through technology



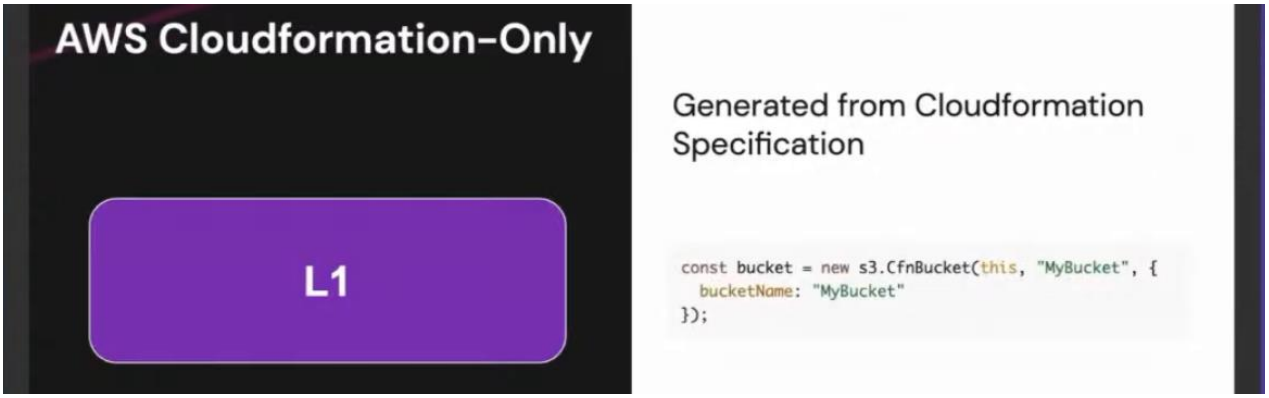
CDK Brings **Encapsulation, Abstraction & Composition**

- CDK Constructs are classes, bringing:
  - **Abstraction**
  - **Encapsulation**
  - **Composition**

- **Platform Team:** Hide complexity, provide sensible defaults and protect “dangerous” settings from accidental changes.
- **Steam Aligned Team:** Combine use-case independent lower level Constructs



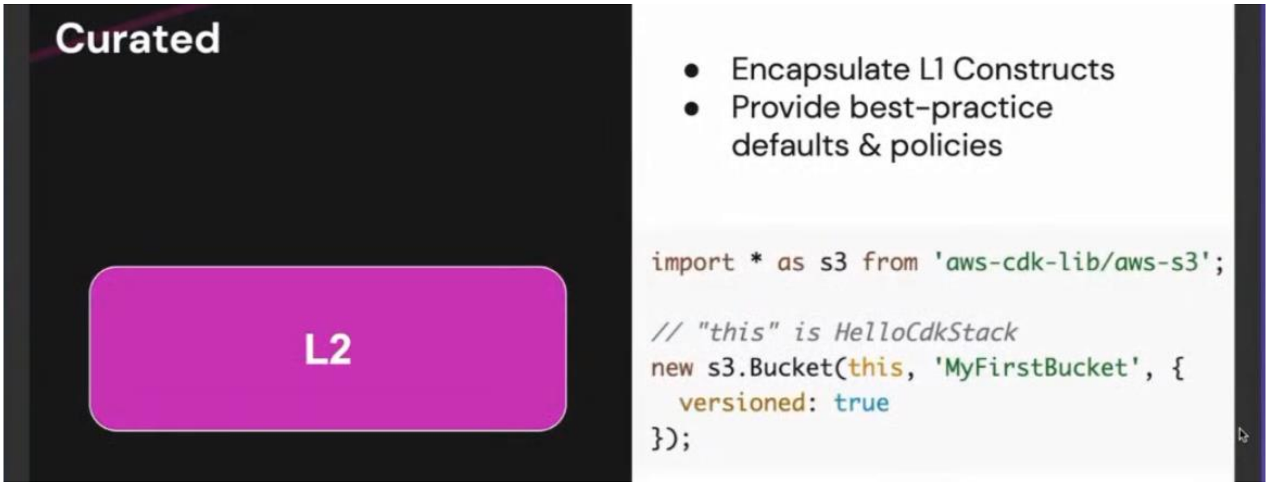
Construct Hierarchy



AWS Cloudformation-Only

Generated from Cloudformation Specification

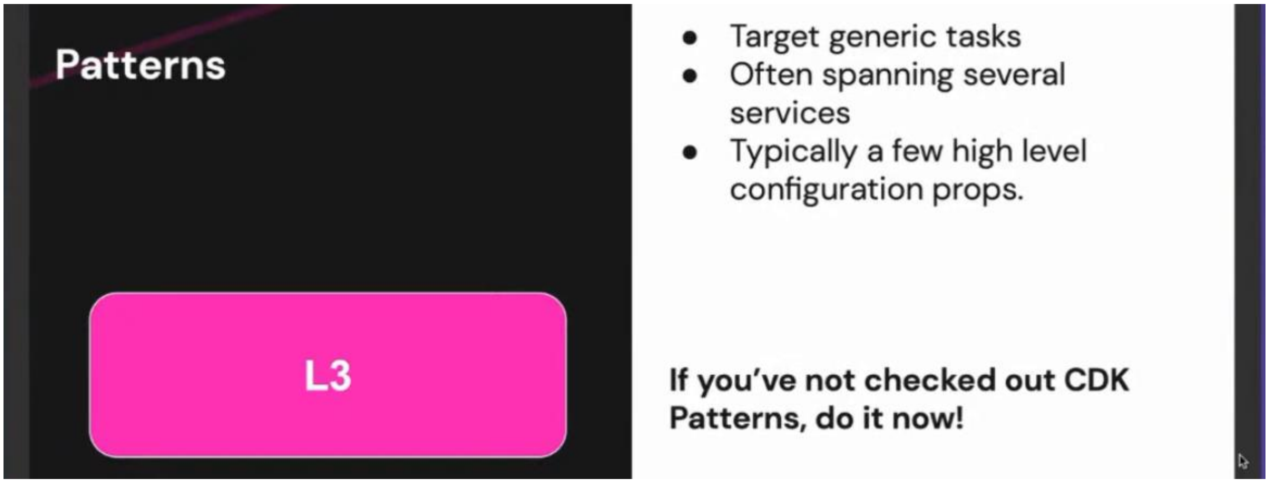
```
const bucket = new s3.CfnBucket(this, "MyBucket", {
  bucketName: "MyBucket"
});
```



Curated

- Encapsulate L1 Constructs
- Provide best-practice defaults & policies

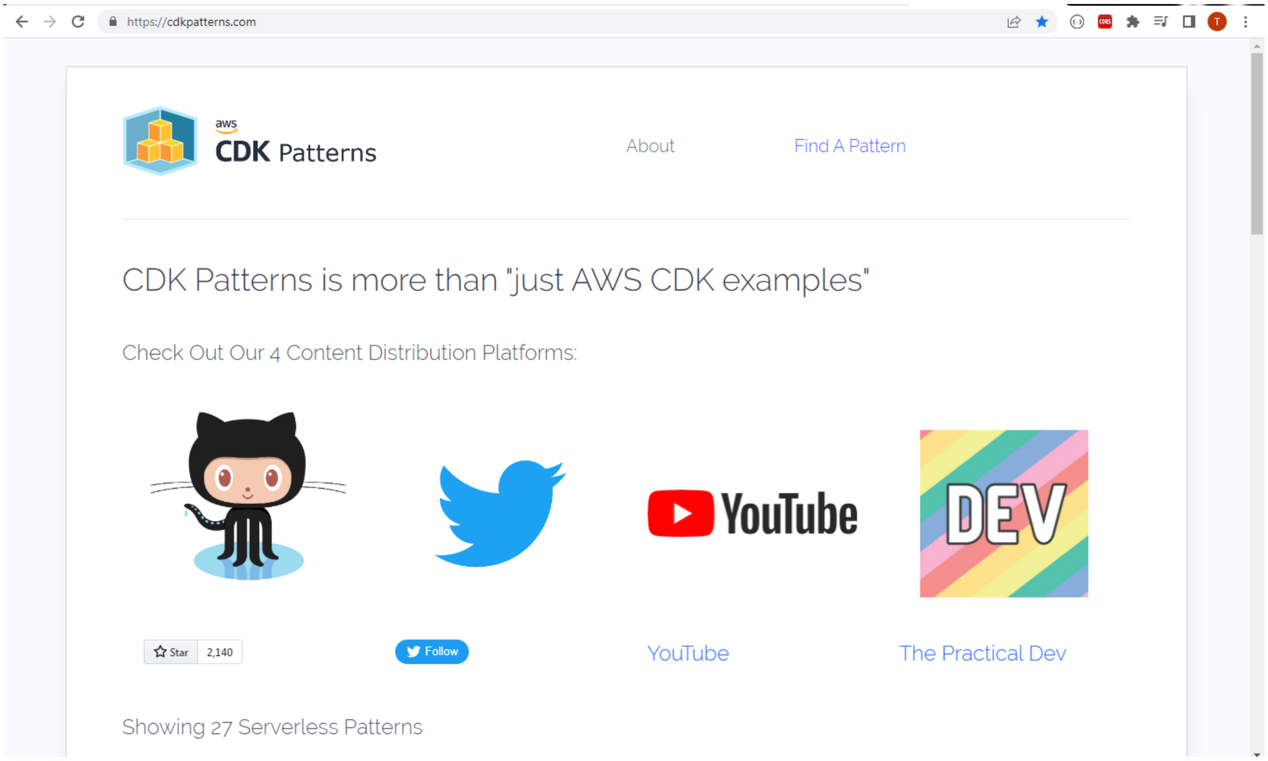
```
import * as s3 from 'aws-cdk-lib/aws-s3';

// "this" is HelloCdkStack
new s3.Bucket(this, 'MyFirstBucket', {
  versioned: true
});
```



Patterns

- Target generic tasks
- Often spanning several services
- Typically a few high level configuration props.

If you've not checked out CDK Patterns, do it now!

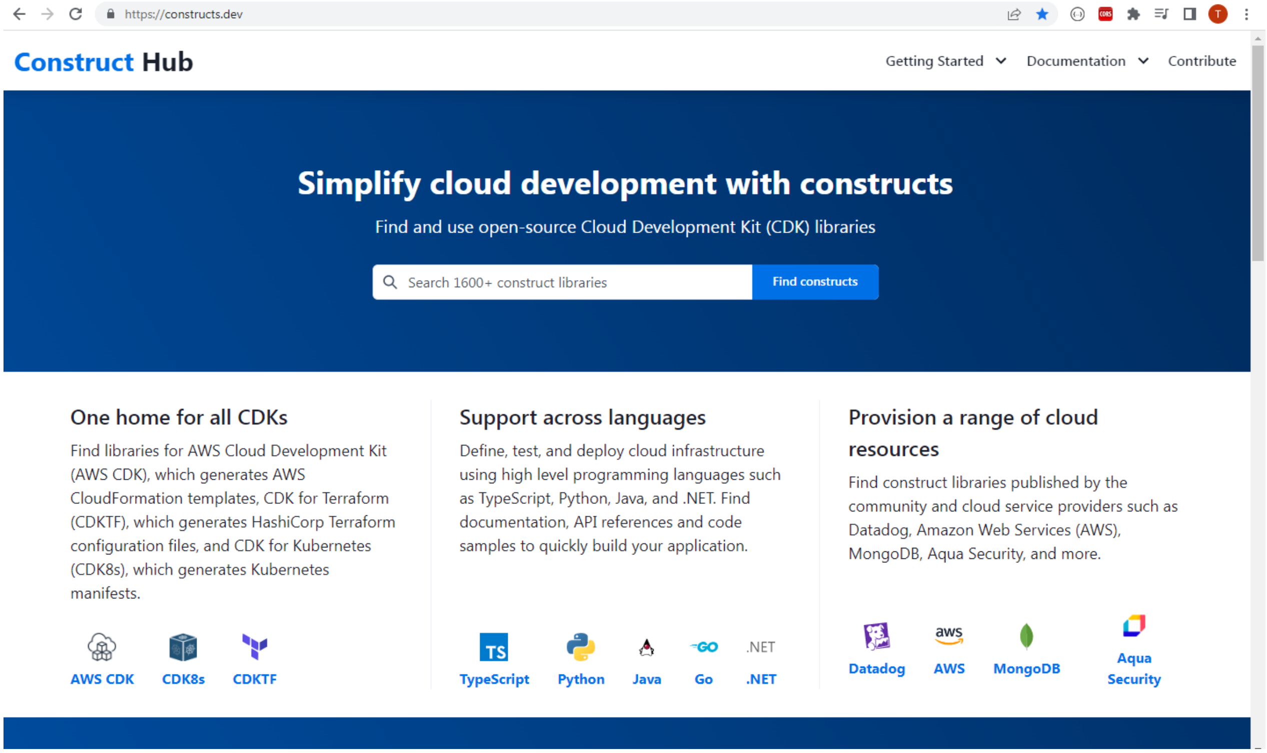


CDK Patterns is more than "just AWS CDK examples"

Check Out Our 4 Content Distribution Platforms:

Showing 27 Serverless Patterns

https://cdkpatterns.com/ and https://constructs.dev/ below are two great CDK resource links



Simplify cloud development with constructs

Find and use open-source Cloud Development Kit (CDK) libraries

One home for all CDKs

Find libraries for AWS Cloud Development Kit (AWS CDK), which generates AWS CloudFormation templates, CDK for Terraform (CDKTF), which generates HashiCorp Terraform configuration files, and CDK for Kubernetes (CDK8s), which generates Kubernetes manifests.

AWS CDK CDK8s CDKTF

Support across languages

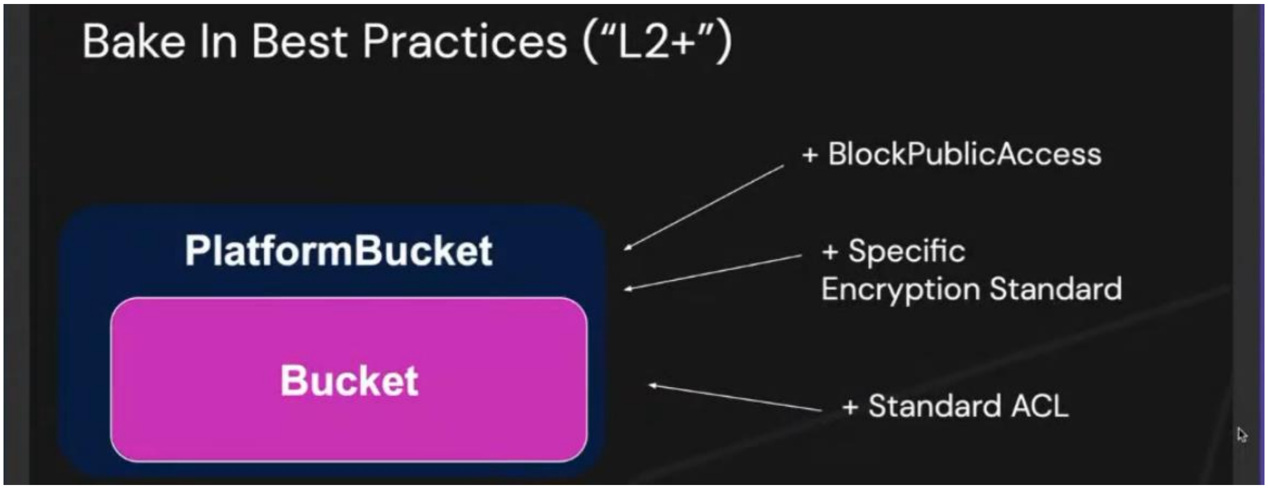
Define, test, and deploy cloud infrastructure using high level programming languages such as TypeScript, Python, Java, and .NET. Find documentation, API references and code samples to quickly build your application.

TypeScript Python Java Go .NET

Provision a range of cloud resources

Find construct libraries published by the community and cloud service providers such as Datadog, Amazon Web Services (AWS), MongoDB, Aqua Security, and more.

Datadog AWS MongoDB Aqua Security



Bake In Best Practices ("L2+")

PlatformBucket

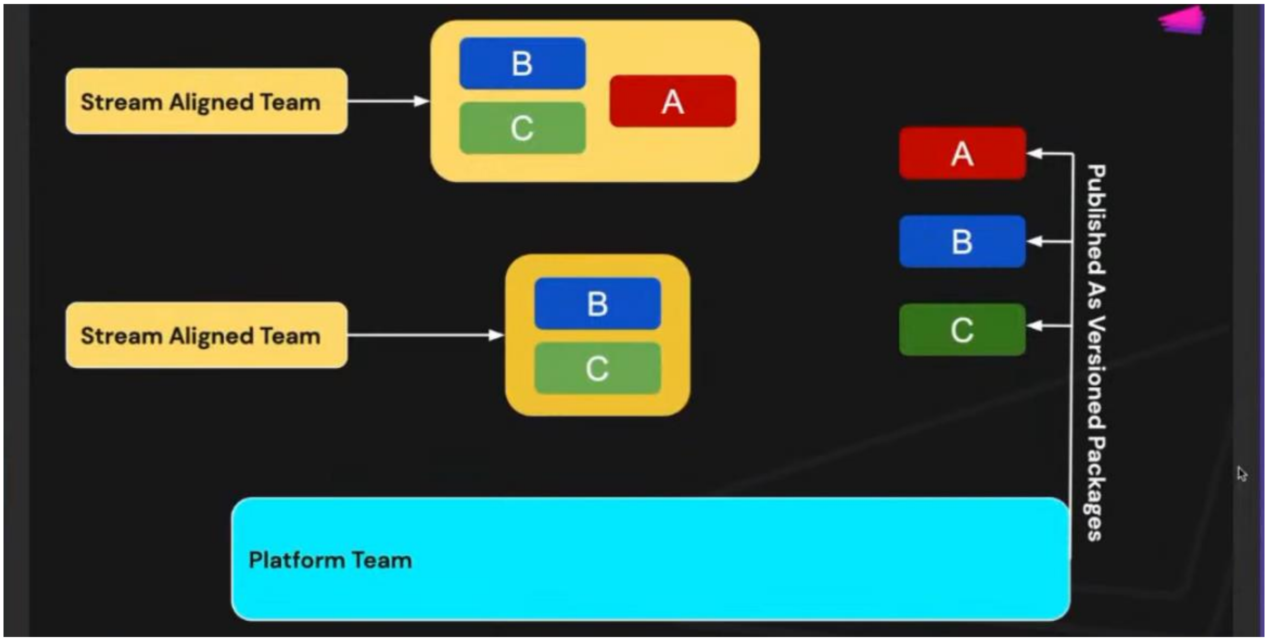
Bucket

+ BlockPublicAccess

+ Specific Encryption Standard

+ Standard ACL

These are constructs that wrap those resources with a few more business specific best practices like security rules, encryption standards and ACL access rules like zero S3 bucket access by default.



Stream Aligned Team

B

C

A

Stream Aligned Team

B

C

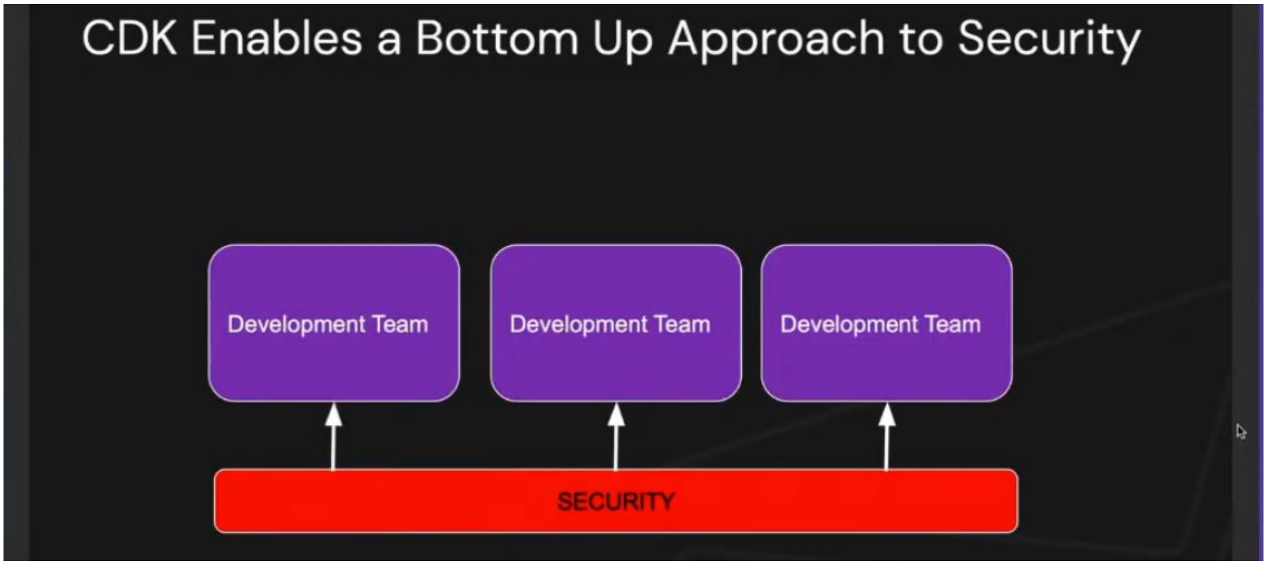
A

B

C

Published As Versioned Packages

Platform Team



CDK Enables a Bottom Up Approach to Security

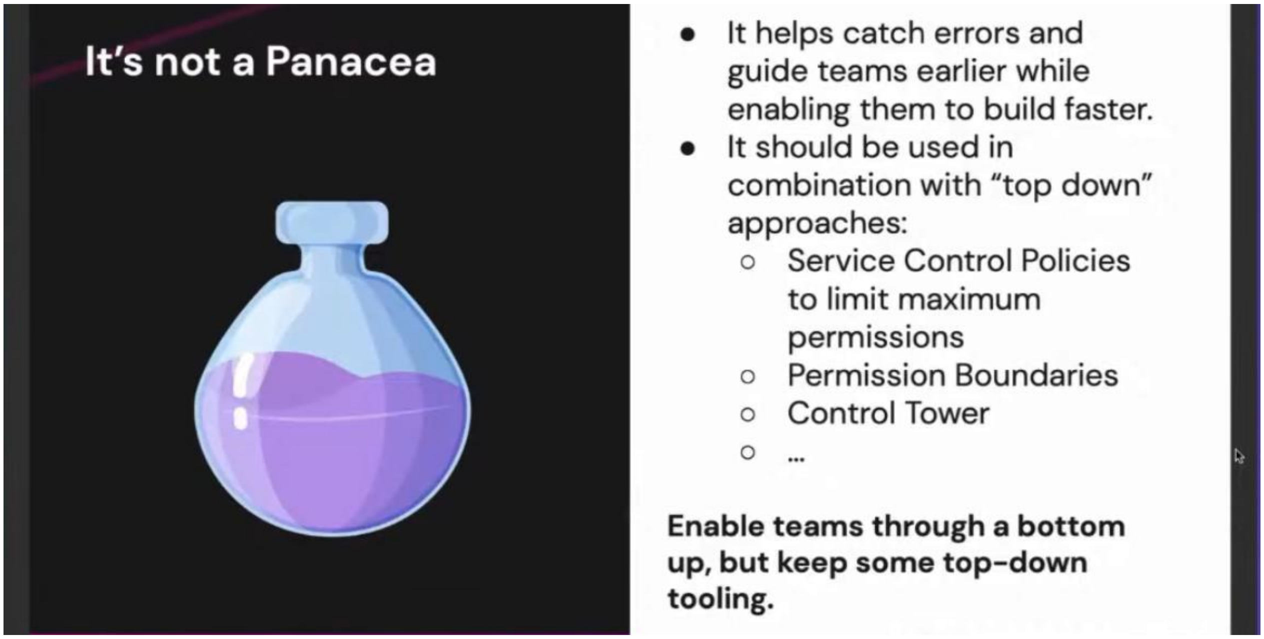
Development Team

Development Team

Development Team

SECURITY

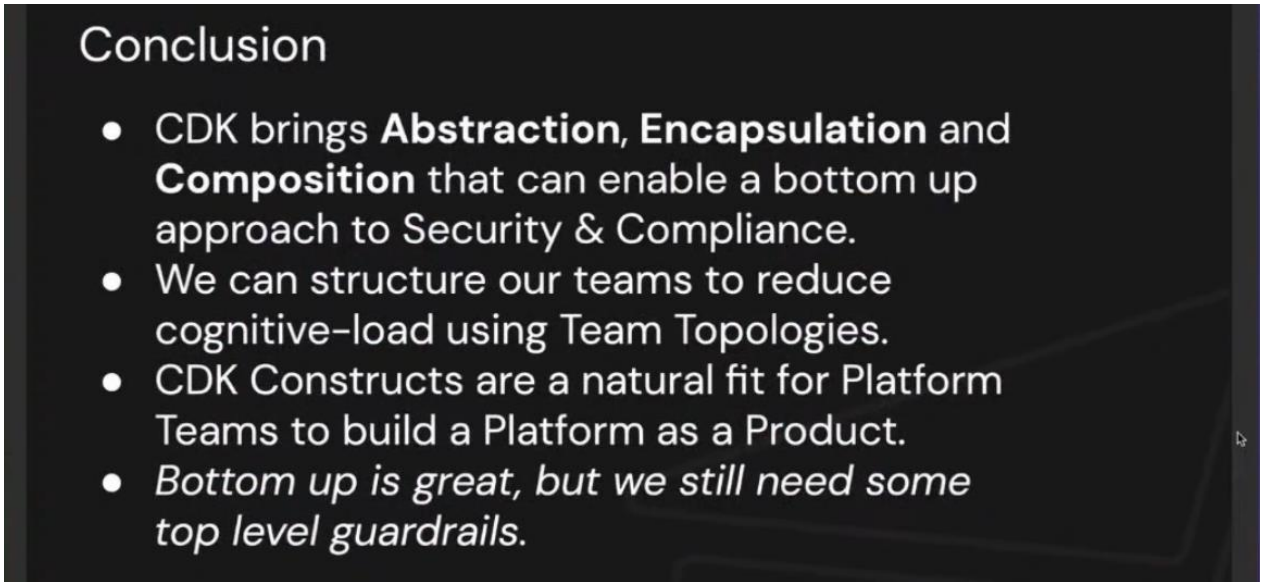
CDK Enables Complexity to be Abstracted by a Platform Team



It’s not a Panacea

- It helps catch errors and guide teams earlier while enabling them to build faster.
- It should be used in combination with “top down” approaches:
  - Service Control Policies to limit maximum permissions
  - Permission Boundaries
  - Control Tower
  - ...

Enable teams through a bottom up, but keep some top-down tooling.



Conclusion

- CDK brings **Abstraction**, **Encapsulation** and **Composition** that can enable a bottom up approach to Security & Compliance.
- We can structure our teams to reduce cognitive-load using Team Topologies.
- CDK Constructs are a natural fit for Platform Teams to build a Platform as a Product.
- *Bottom up is great, but we still need some top level guardrails.*