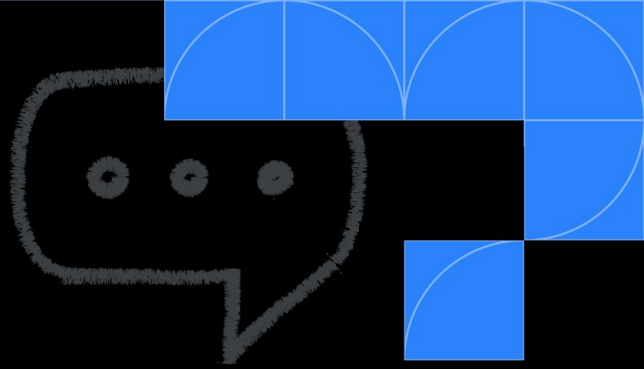
```
Text(
  'Simple Statement or URL',
  style: TextStyle(
    color: Colors.blue[200],
  ),
),
```

devfest

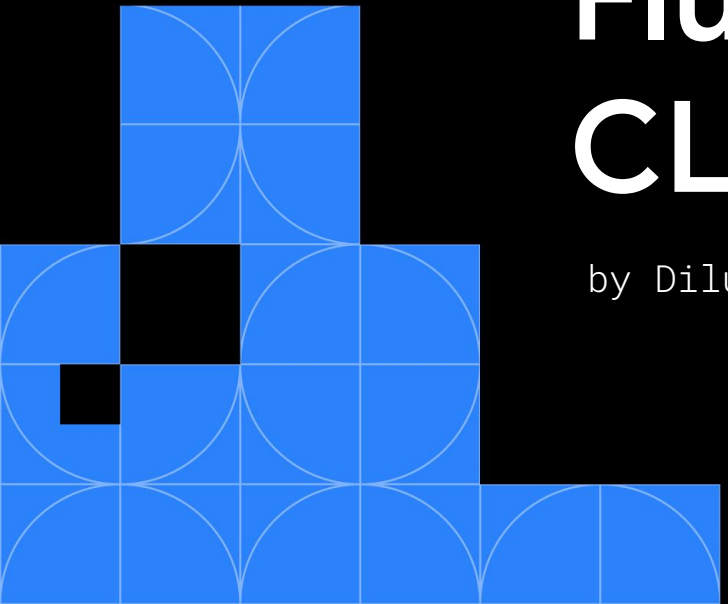
Sri Lanka

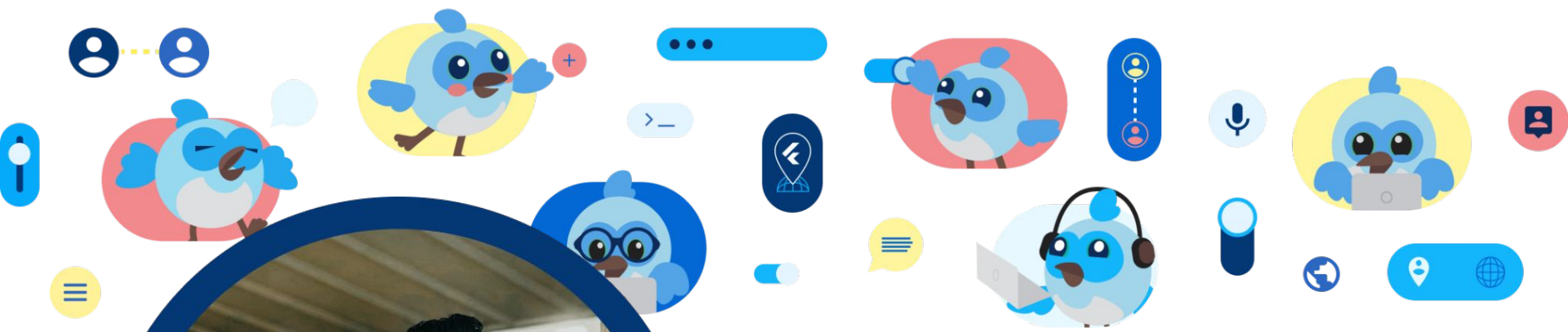
```
s.star,
r: Colors.blue[500],

Text('23'),
```



Flutter meets CLEAN Architecture

by Dilum De Silva





Kia ora

I'm **Dilum De Silva,**

Co-Founder at **Evolza**
Manager **GDG Auckland**
Lead **Flutter Auckland**
Mobile Engineer at **Hectre NZ**

Architectures

MVC, MVP, MVVM, VIPER ..

```
lookup.KeyValue
f.constant(['em
=tf.constant([0
.lookup.StaticV

_buckets=5)
```

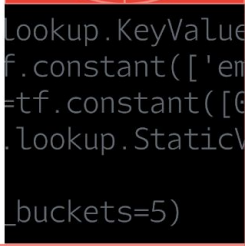
Flutter & clean

Flutter is Google’s portable UI toolkit for crafting, natively compiled applications for mobile, web, and desktop from a single codebase.

```
lookup.KeyValue
f.constant(['em
=tf.constant([0
lookup.StaticV

_buckets=5)
```

What is CLEAN?

The Clean Architecture, developed by **Robert C. Martin**, "Uncle Bob" is a design philosophy that promotes the **separation of concerns** within a software. The main goal is to produce a system that is independent of UI, databases, frameworks, and external agencies, making it more **maintainable**, **testable**, and **scalable**.



Why CLEAN for Flutter?

More Separation of concerns, loosely coupled, isolation of business logic & thus better long term maintainability,testability & scalability of projects.

```
lookup.KeyValue
f.constant(['em
=tf.constant([0
.lookup.StaticV

_buckets=5)
```

Real world benefits of CLEAN

Enhancements & Maintainability: Features can be added or modified without disturbing other parts of the system. Bugs can be identified and fixed more rapidly,This modular approach ensures that individual components can evolve independently.

Replaceability: Replaceability refers to the ease with which a specific component, module, or piece of software can be replaced with an alternative or substitute without causing significant disruption to the overall system's functionality. .

Real world benefits of CLEAN

Readability: Having a well-structured codebase makes code more readable and new developers can be onboarded faster since they can understand the architecture's clear structure.

Improved Collaboration: Since each layer has its own distinct responsibility, multiple developers can work on different layers without stepping on each other's toes. This is especially beneficial in larger teams or projects.

Real world benefits of CLEAN

Testability:

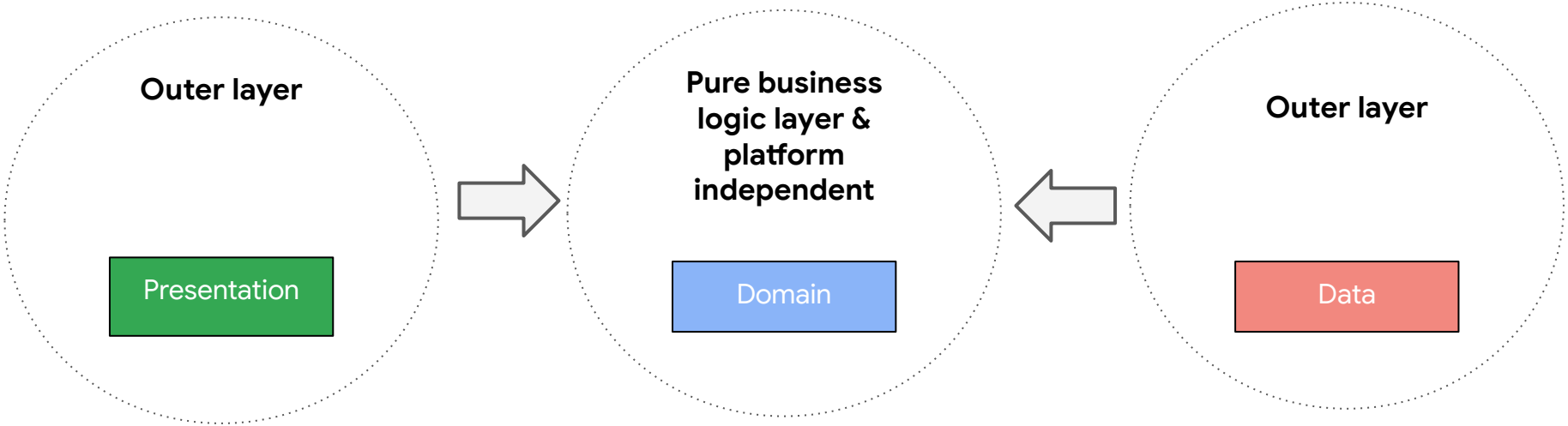
Clean Architecture makes it much easier to write unit tests. The core logic is separated from the framework and external interfaces. This means you can test the core logic (use cases and entities) without any dependency on the Flutter framework or any other external systems.

How to organise your project CLEAN?

presentation, domain, data

```
lookup.KeyValue
f.constant(['em
=tf.constant([0
.lookup.StaticV

_buckets=5)
```

Break down project into different folders also known as layers



- Widgets
- Screens
- State & Cubits
- Theme, styles, fonts.
- Utils, extensions

- Entities
- Use cases
- Abstract classes for repositories
- Utils, extensions

- Repository implementation
- Data sources
- Network classes
- Utils, extensions

Thank you!
Let’s hear from you...

Access session materials at

bit.ly/flutter-meets-clean-architecture