Google Developers

Developer Student Clubs

Flutter Study Jam İzmir

Flutter ve Dart'a Giriş

Gazihan ALANKUŞ, PhD
Google Developer Expert for Dart
Assistant Professor, Izmir University of Economics
Founder gbot.dev
@gazialankus

Neden?

Flutter ve Dart

Popüler, hızla yayılan bir teknoloji

- Hızla yayılıyor - potansiyel iş imkanları
- Çok keyifli ve verimli bir geliştirici deneyimi
- Çok platformlu mobil (Android ve iOS), web ve yakında masaüstü uygulamalar geliştirebilme
- Google tarafından geliştiriliyor, Fuschia'nın merkezinde

Dart

Dart: Yeni Bir Dil

Kullanması çok keyifli

- Java ve JavaScript arasında
- Java gibi strongly typed, JavaScript gibi esnek
- Güçlü IDE desteği: IntelliJ and VSCode
- Harika bir standart kütüphane
- Asenkron özellikler dilin ve kütüphanenin içerisinde
- JIT, AOT, tree shaking, vs.

Dart: Yeni Bir Dil

Güçlü fakat basit


```
List<String> aListOfStrings = ['one', 'two', 'three'];
Set<String> aSetOfStrings = {'one', 'two', 'three'};
Map<String, int> aMapOfStringsToInts = {
  'one': 1,
  'two': 2,
  'three': 3,
};
enableFlags(bold: true, hidden: false);

querySelector('#confirm')
..text = 'Confirm'
..classes.add('important')
..onClick.listen((e) => window.alert('Confirmed!'));
```

```
final newList = [
      anElement,
      ...anotherList,
      if (condition) aConditionalElement,
];
```

Dart: Yeni Bir Dil

Dilde ve standart kütüphanede asenkron desteği

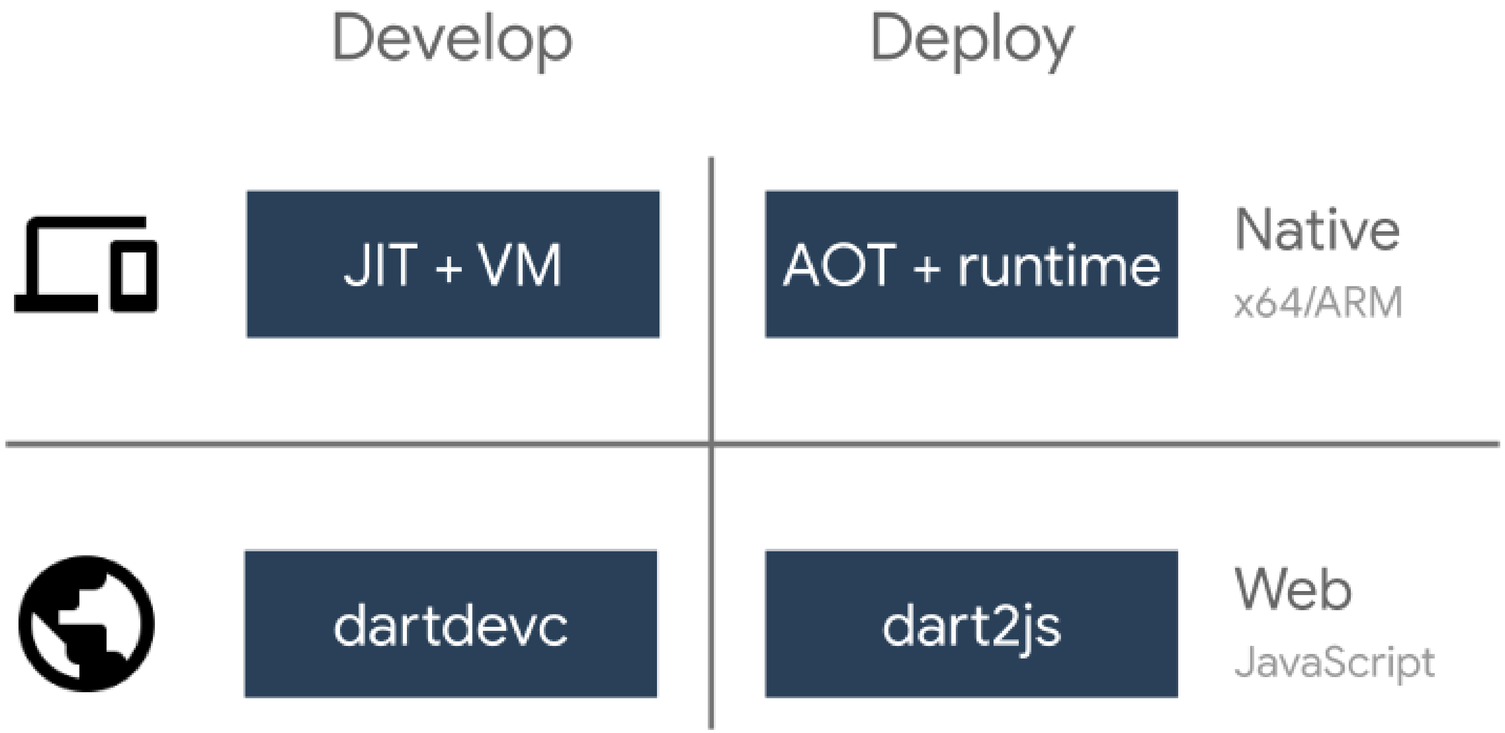

```
Future result = costlyQuery(url);
result
    .then((value) => expensiveWork(value))
    .then((_) => lengthyComputation())
    .then((_) => print('Done!'))
    .catchError((exception) {
  /* Handle exception... */
});
```

```
try {
  final value = await costlyQuery(url);
  await expensiveWork(value);
  await lengthyComputation();
  print('Done!');
} catch (e) {
  /* Handle exception... */
}
```

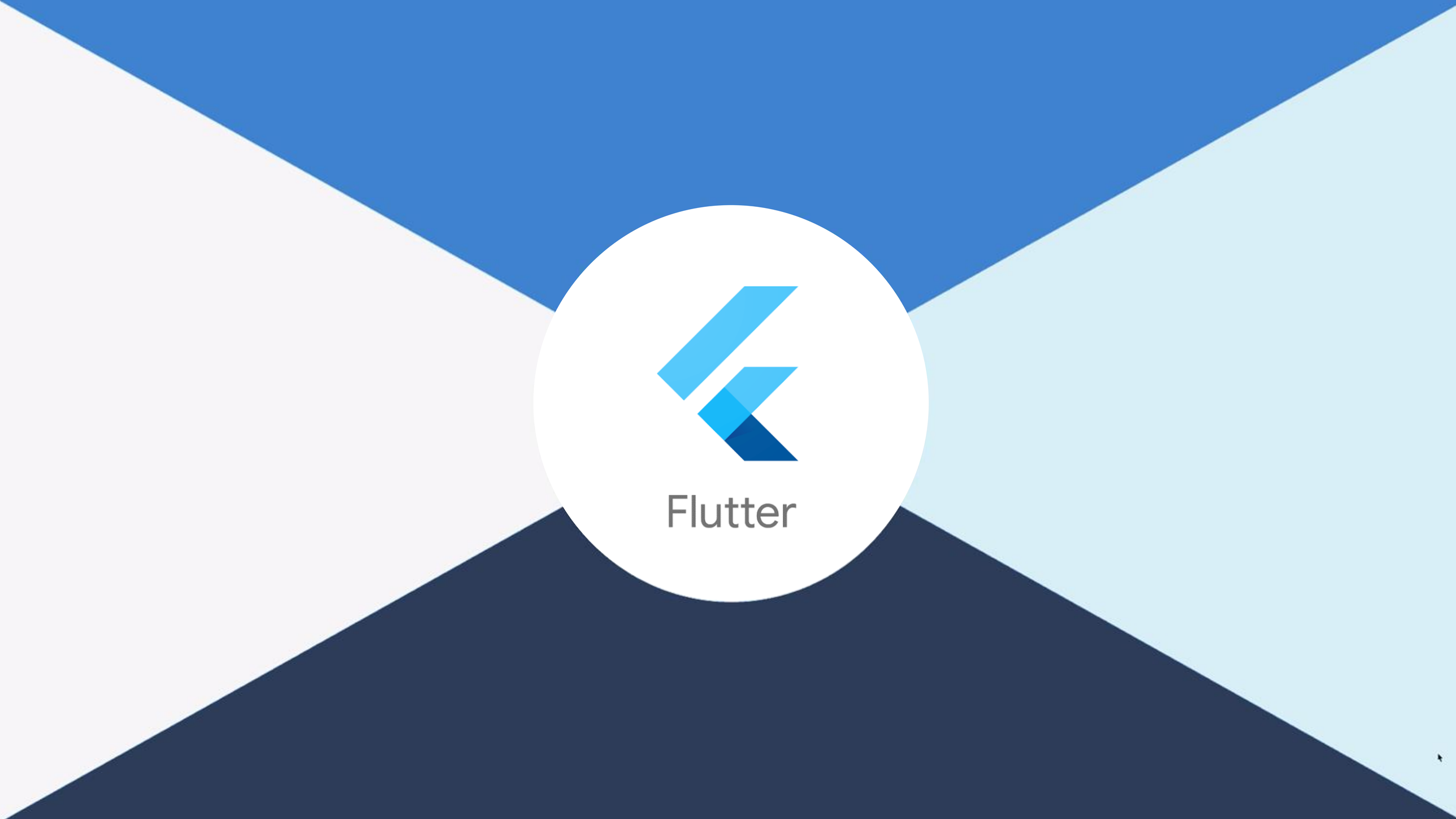
```
var config = File('config.txt');
var contents = await config.readAsString(); // don't make the thread wait here
```

Dart: Yeni Bir Dil

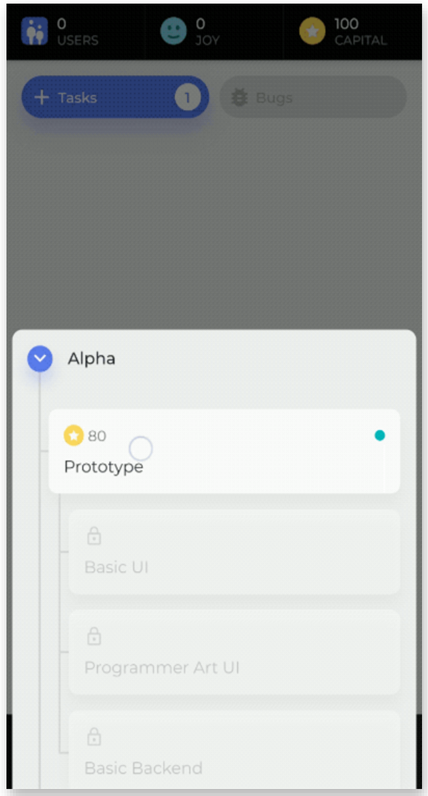
Çevik geliştirme deneyimi, kullanıcı önünde çok hızlı

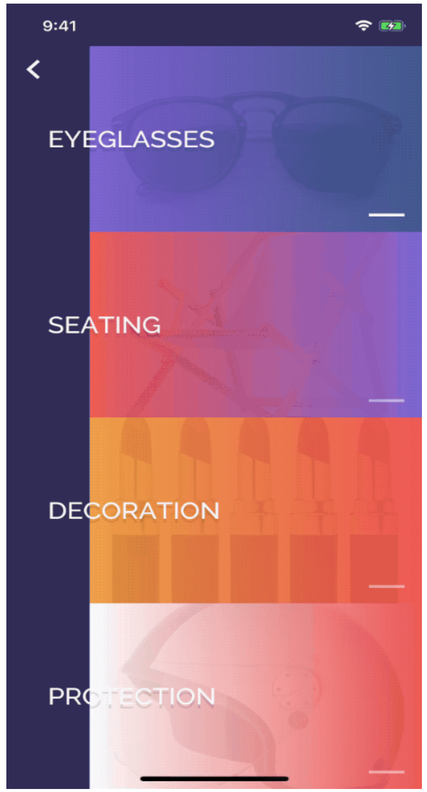


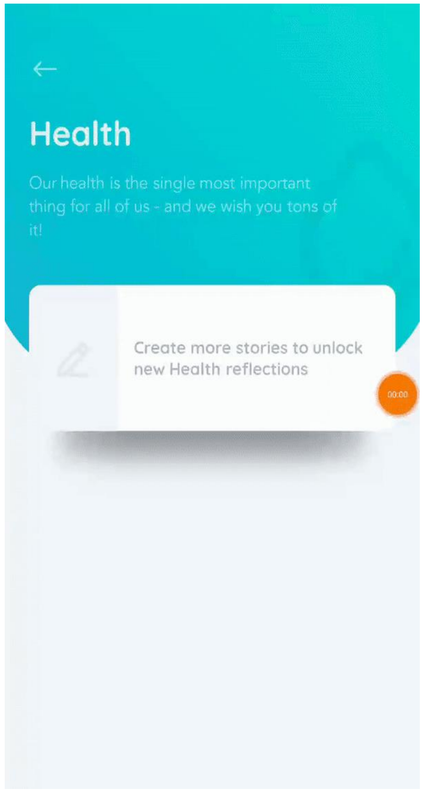
Flutter


Flutter

Tek bir kod projesi ile mobil (Android ve iOS), web ve masaüstü uygulaması geliştirmek mümkün







Güzel

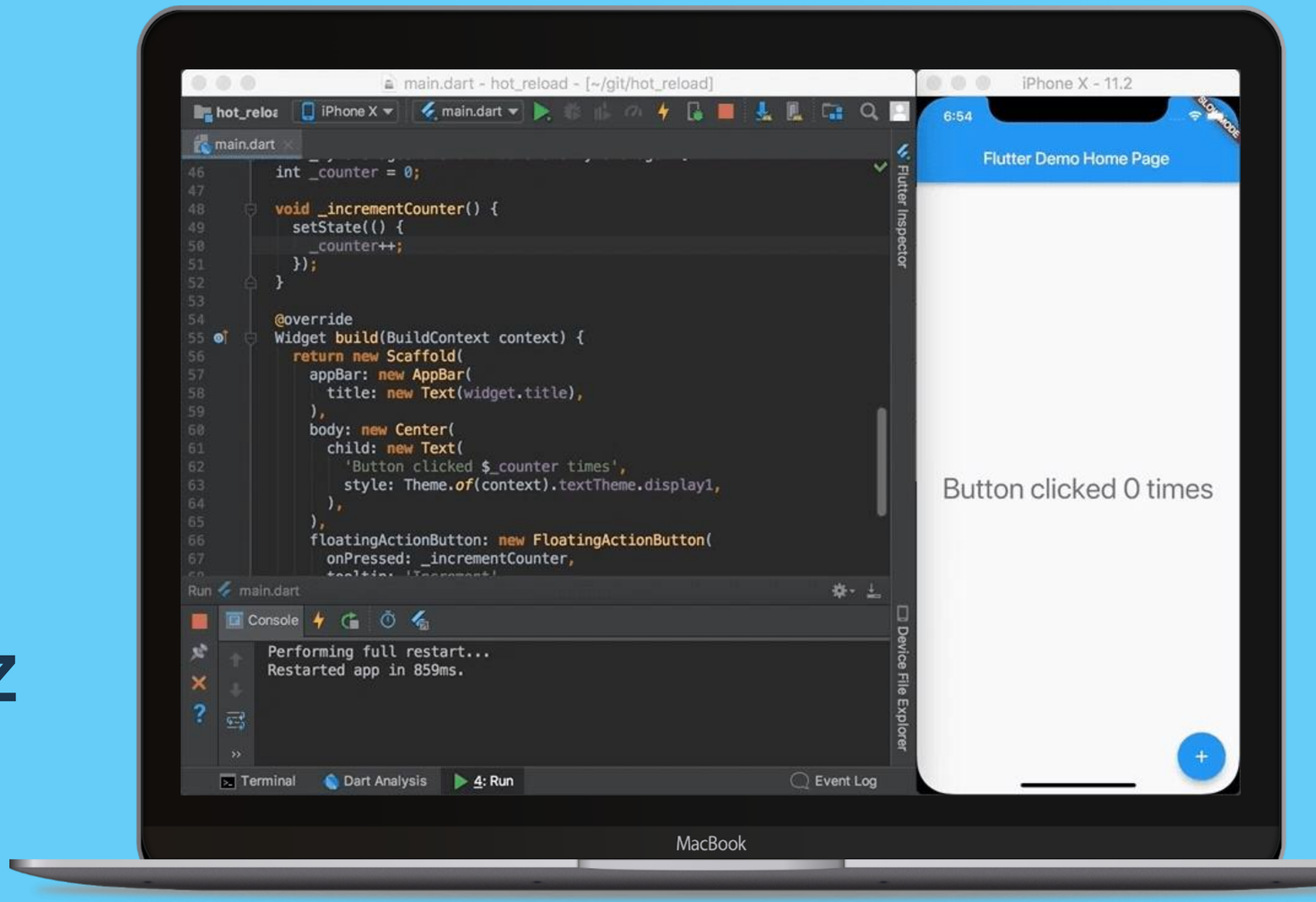
Tasarımcılara hayır demeyin

Hızlı

Akıcı uygulamalar

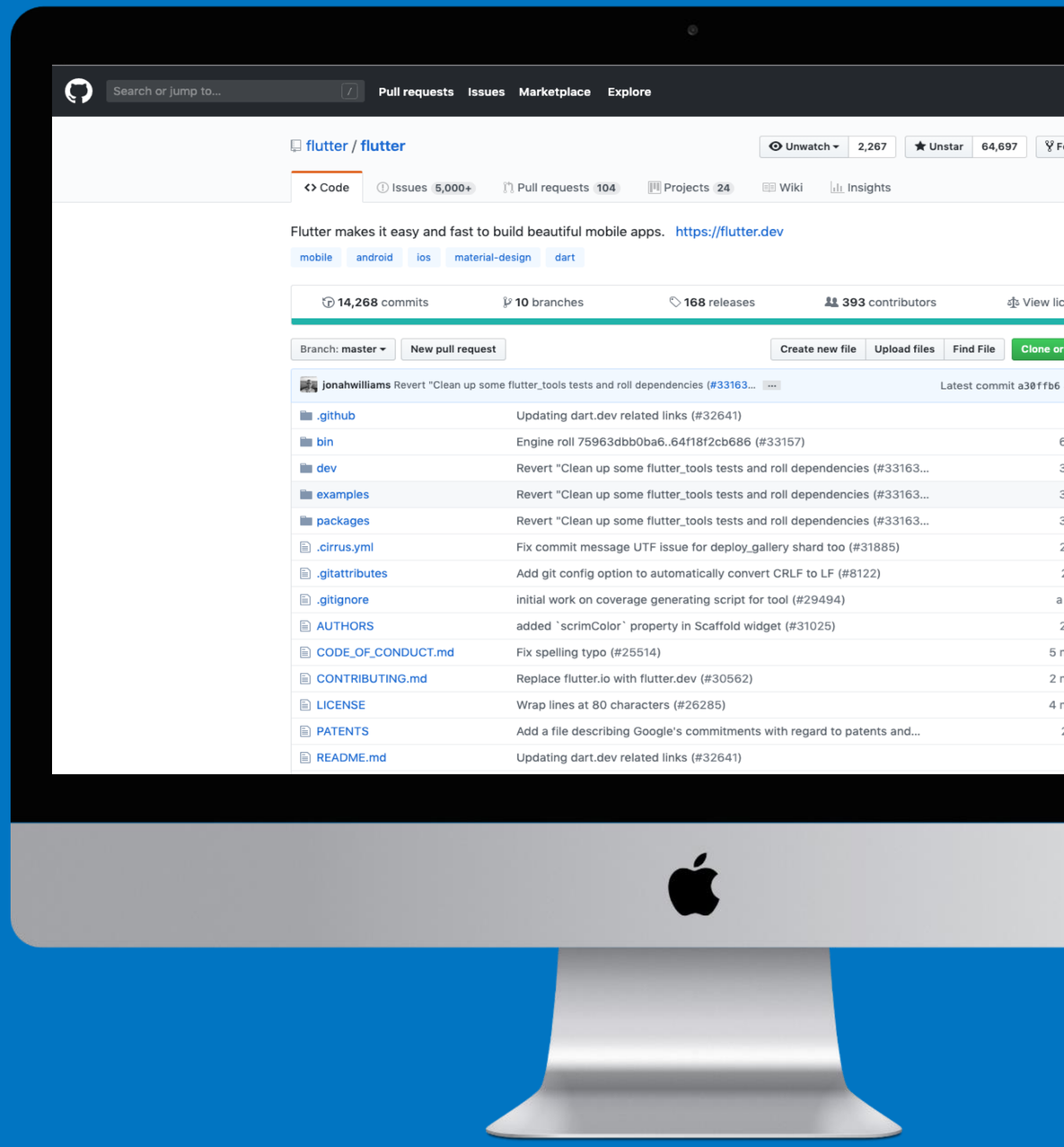
Üretken

Uygulamanız çalışıyorken onu geliştirebiliyorsunuz



Açık

Hepsi open source



Hızlı

Güzel

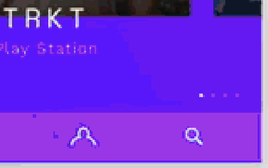
Açık

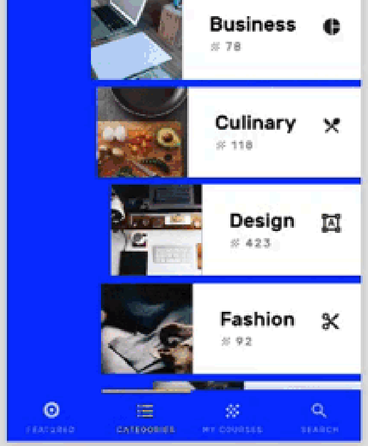
Üretken

Günümüzdeki Durum

- LinkedIn verilerine göre Flutter yazılım mühendisleri arasındaki en hızlı büyüyen teknoloji
- GitHub’da 10. sıradaki repo
- Yüzlerce milyon kullanıcıya ulaşan uygulamalar Flutter ile yazılmış durumda

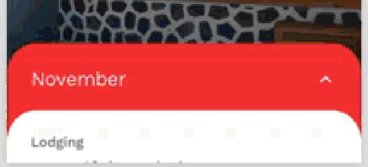
flutter.dev/showcase

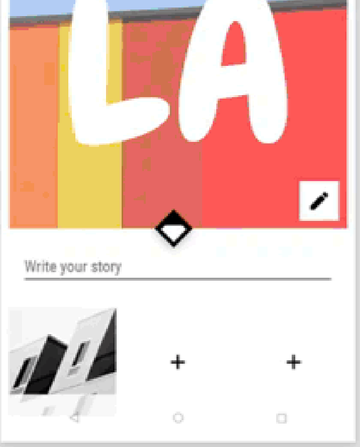

TRKT
Play Station

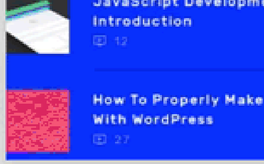

Business
78
Culinary
116
Design
423
Fashion
92


Travel
Magazine

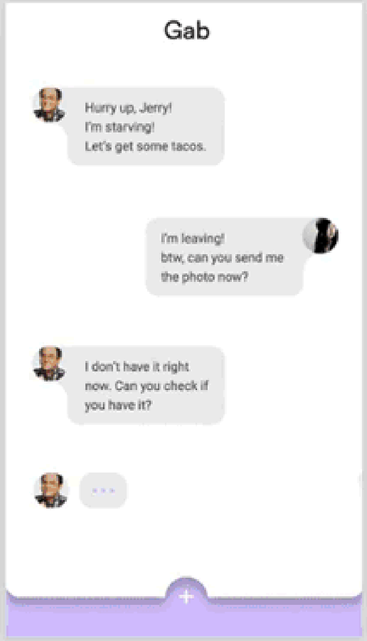

Shows a Crack
By Carl Hulse
After the massacre in Orlando Florida some Republicans finally showed a willingness to support gun control
A widely publicized sit-in led by House Democrat, John Lewis - a bipartisan compromise proposal in the Senate. Neither is very likely to lead to any legislative action in Congress on gun safety this year. Election Day is too close, and m
opposition
John Lewis
D-GA 5th District

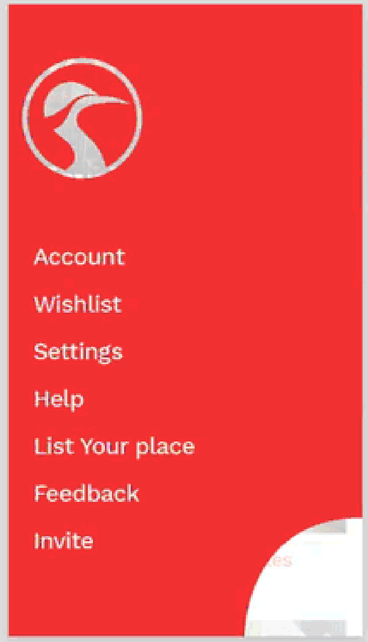

November
Lodging


LA
Write your story


How To Properly Make
With WordPress

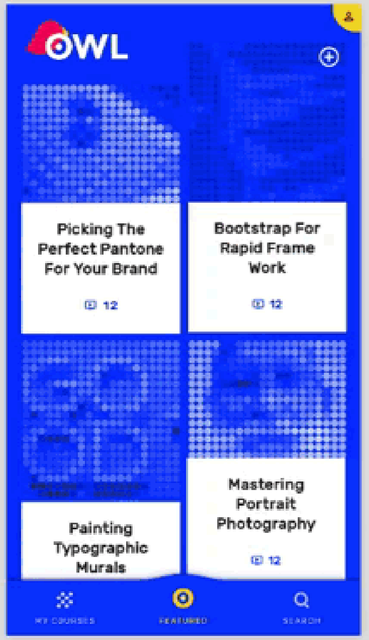

Build Apps
MY COURSES


Gab
Hurry up, Jerry!
I'm starving!
Let's get some tacos.
I'm leaving!
btw, can you send me the photo now?
I don't have it right now. Can you check if you have it?


Account
Wishlist
Settings
Help
List Your place
Feedback
Invite


Books in Los Angeles, Zackary
and Rhys Ernst perched on stoo
discuss their new book of photo
"Relationship." It is by far the m
personal of the many projects t
worked on together, the people
about to move into her basemen
after their photographs were fi
the Whitney, they began workin
first season of "Transparent"
Transparent is a comedy web televisi
created by Amazon Studios. The story
around a Los Angeles family and thei
following the discovery that the perso
knew as their father Mort (Jeffrey Ta
transgender.
Front Page

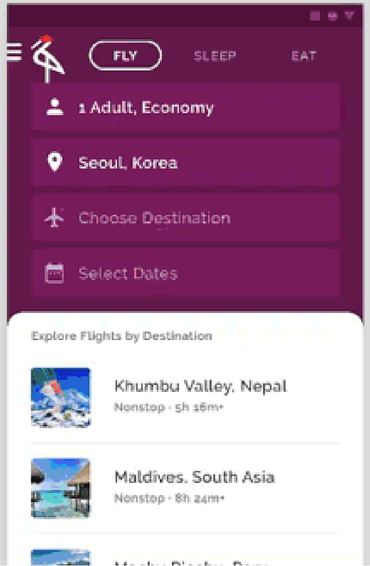

BASIL
Herb Roasted Chicken


OWL
Picking The Perfect Pantone For Your Brand
12
Bootstrap For Rapid Frame Work
12
Painting Typographic Murals
Mastering Portrait Photography
12

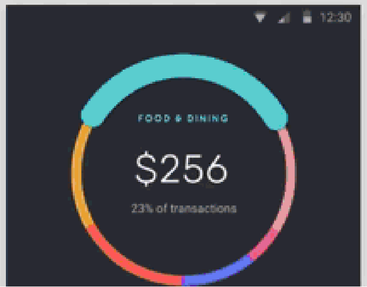

HYPE
boy

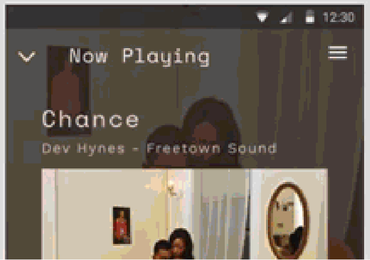

Search
Succulents & Air plants
The best from our winter botanical collection.
Succulent collection
$69
Stone pot succuler
$500

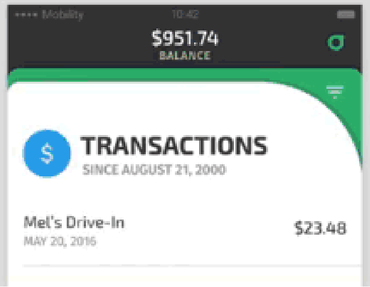

Brunch this weekend?
To me, Rachel and Zach
I'll be in your neighborhood doing errands this weekend if you want to meet up for brunch.
Let me know!


FLY
SLEEP
EAT
1 Adult, Economy
Seoul, Korea
Choose Destination
Select Dates
Explore Flights by Destination
Khumbu Valley, Nepal
Nonstop · 5h 16m
Maldives, South Asia
Nonstop · 8h 24m


Rae Sremmur
Play Station
POPULAR TRACKS
No Type
No Flex Zone
Look Alive


FOOD & DINING
$256
23% of transactions


Now Playing
Chance
Dev Hynes - Freetown Sound


$951.74
BALANCE
TRANSACTIONS
SINCE AUGUST 21, 2000
Mel's Drive-In
MAY 20, 2016
$23.48

Flutter: çabucak, güzel

Cross-platform mobil uygulamalar

Cross-platform ve “to the metal”

Mobil Geliştirme Zorlukları

Diğer frameworkler

“To the metal” yaklaşımlar

- ✓ **Yüksek kalite uygulama**
  Platform ve system entegrasyonu
- ✓ **Yüksek performanslı arayüzler**
  Native kod, GPU ile hızlı
- ✗ **İki uygulama maliyeti**
  İki takım, iki kod projesi, yatırım
- ✗ **Tutarsızlıklar**
  Farklı cihazlarda farklı deneyim

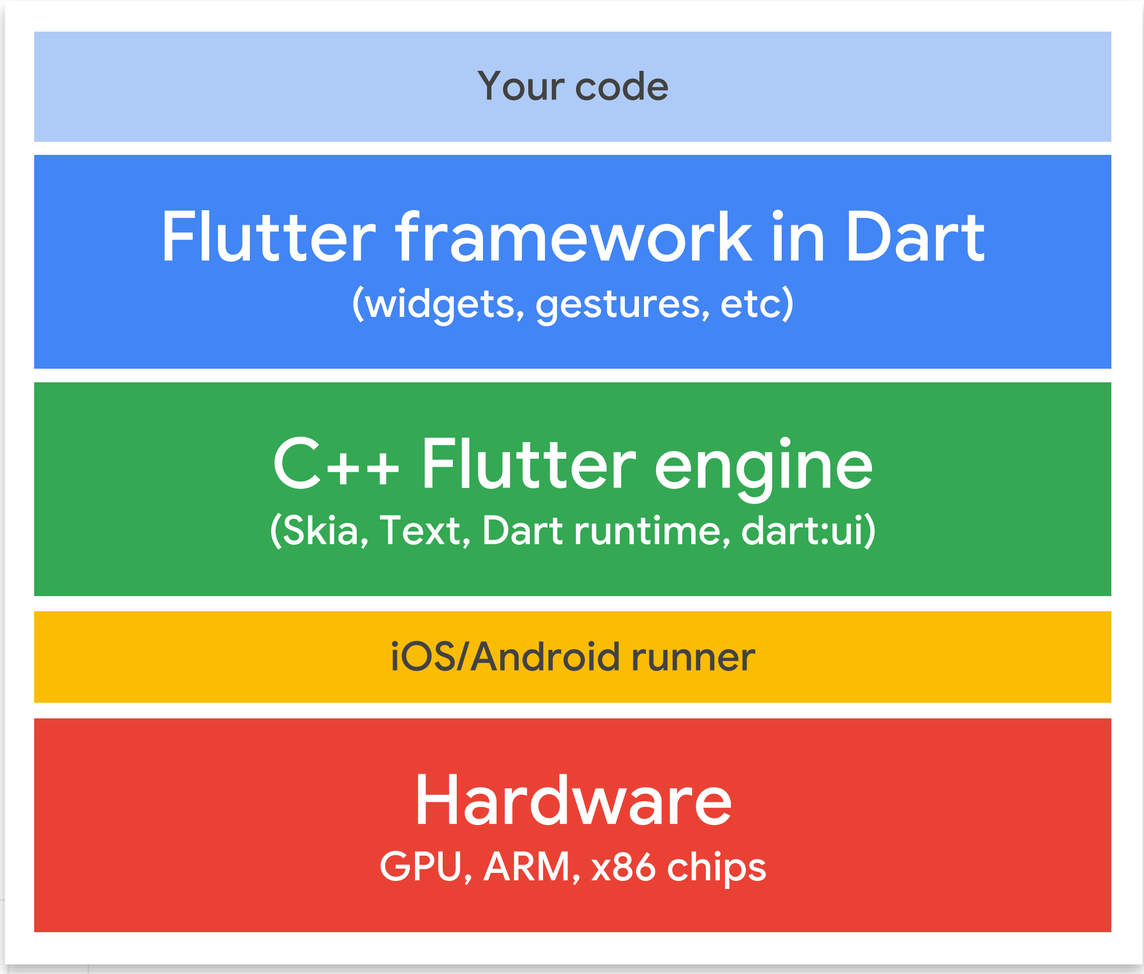
“Cross platform” yaklaşımlar

- ✓ **Hızlı geliştirme**
  Hızlı iterasyonlar
- ✓ **Taşınabilirlik**
  Tek bir kod projesi
- ✗ **Düşük performans**
  Yavaşlık farkedilebilir
- ✗ **Native olmayan deneyim**
  Kullanıcılar farkı hissediyor

Flutter iki tarafın iyi yanlarını içeriyor

> Flutter combines **native performance and quality** with **high-velocity development** and **multi-platform reach**.

Flutter native uygulama üretiyor



Arayüz ağaç yapısı

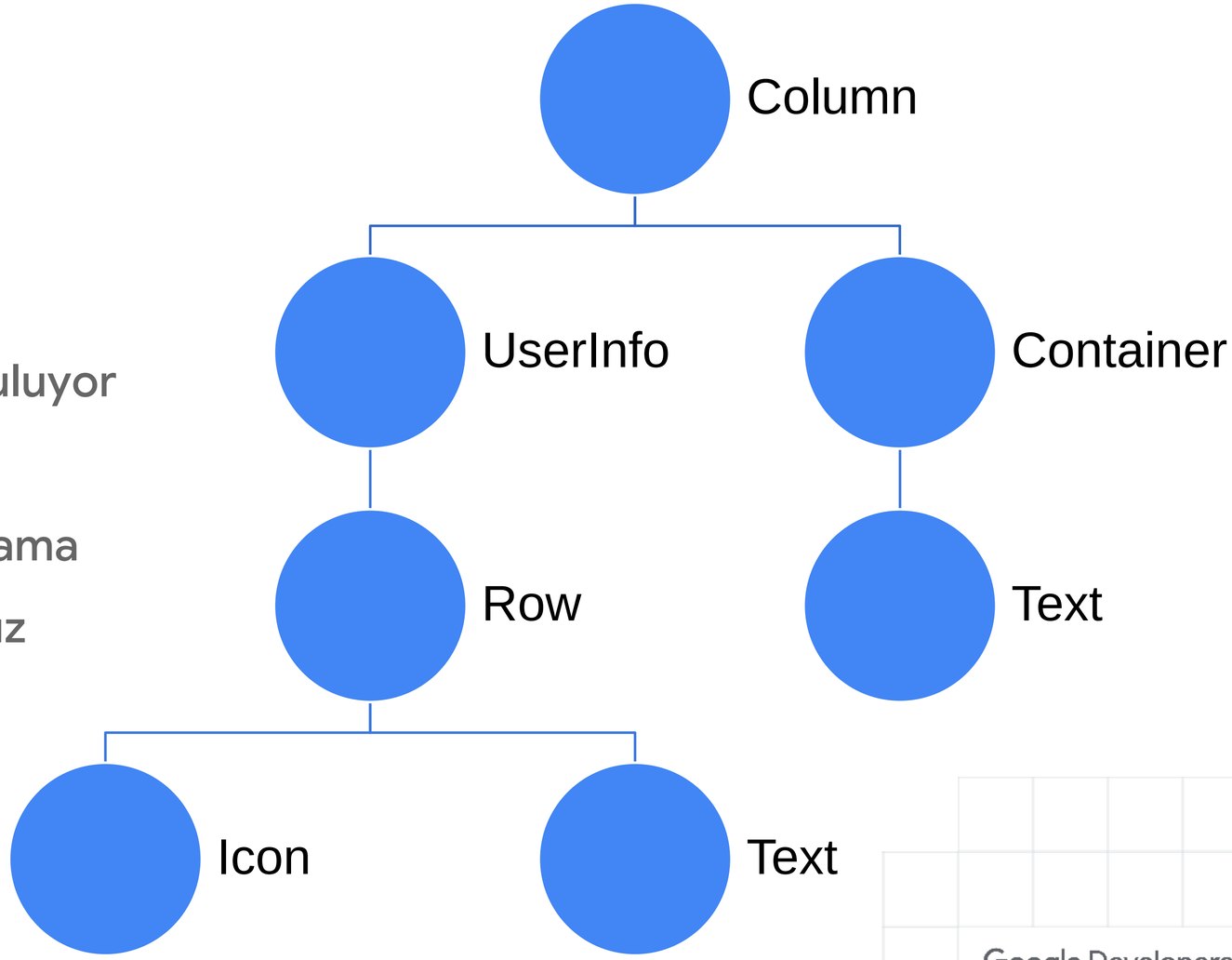
Ağacı yaratarak tanımlıyoruz

Tekrar tekrar yaratıyoruz, sorun değil

Arka planda diffliyor ve değişiklikleri buluyor

Alıştığınızdan farklı, "declarative" programlama

- Her yere istediğiniz gibi uzanamıyorsunuz



Arayüz ağaç yapısı

Stateful ve Stateless widgetlar

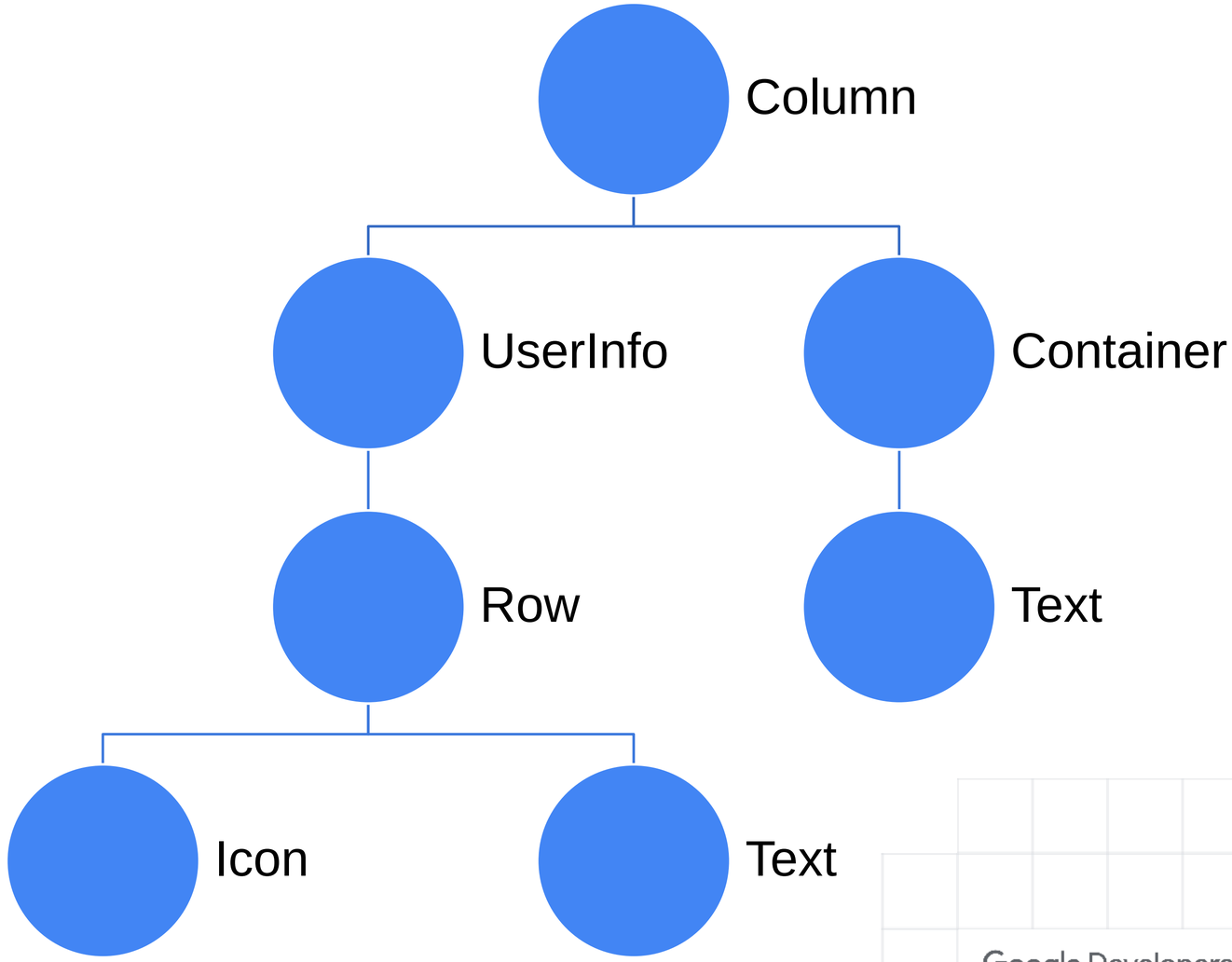
StatelessWidget

- build()

StatefulWidget

State

- initState()
- build()



Arayüz ağaç yapısı

Stateful ve Stateless widgetlar

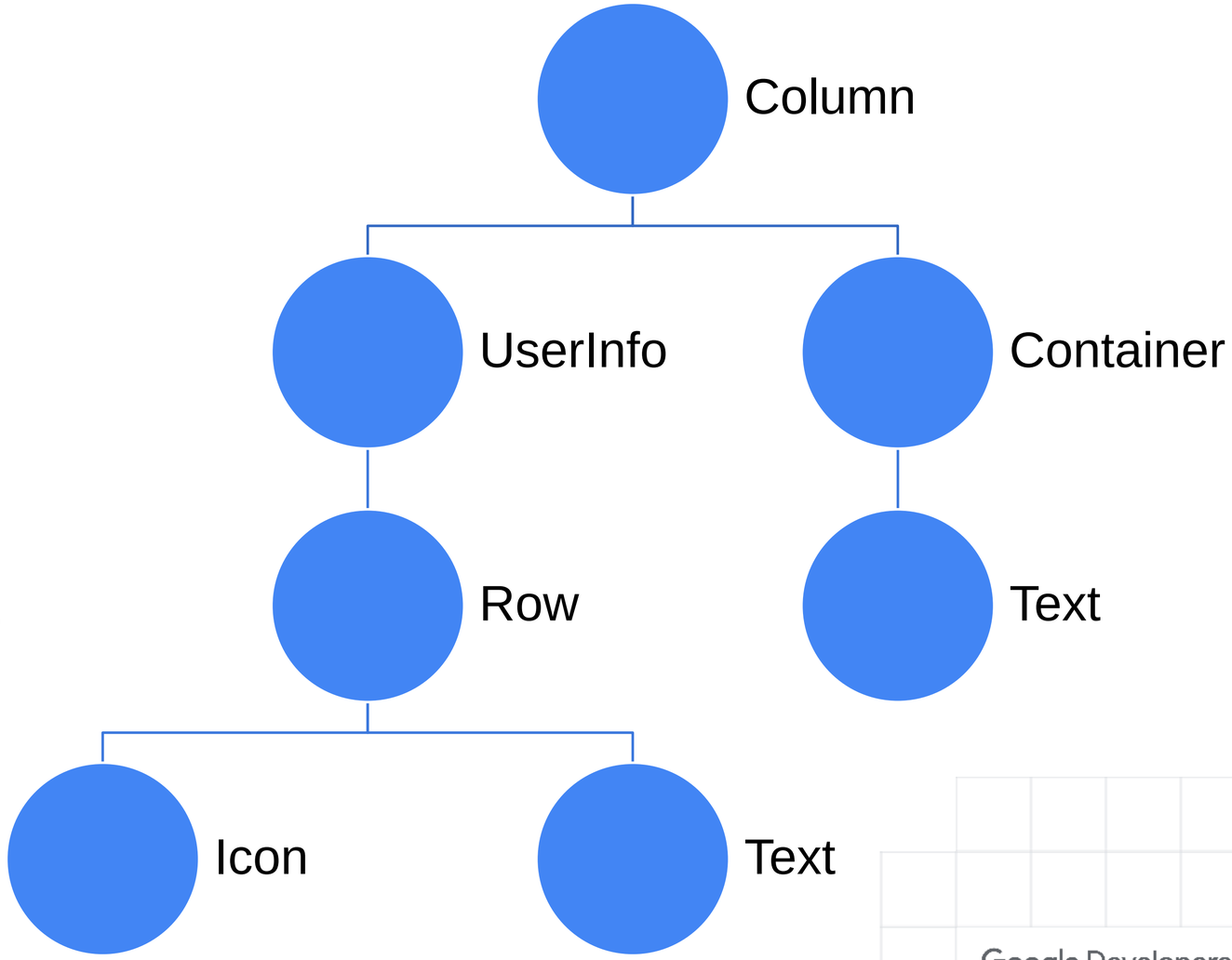
Widgetlar immutable

Widget’a constructor’da veriler iletiliyor

Stateless o verileri ekrana basıyor

Stateful o verileri ilk değer olarak kullanıyor

- State classında değişimi yönetiyor



Arayüz ağaç yapısı

Stateful ve Stateless widgetlar

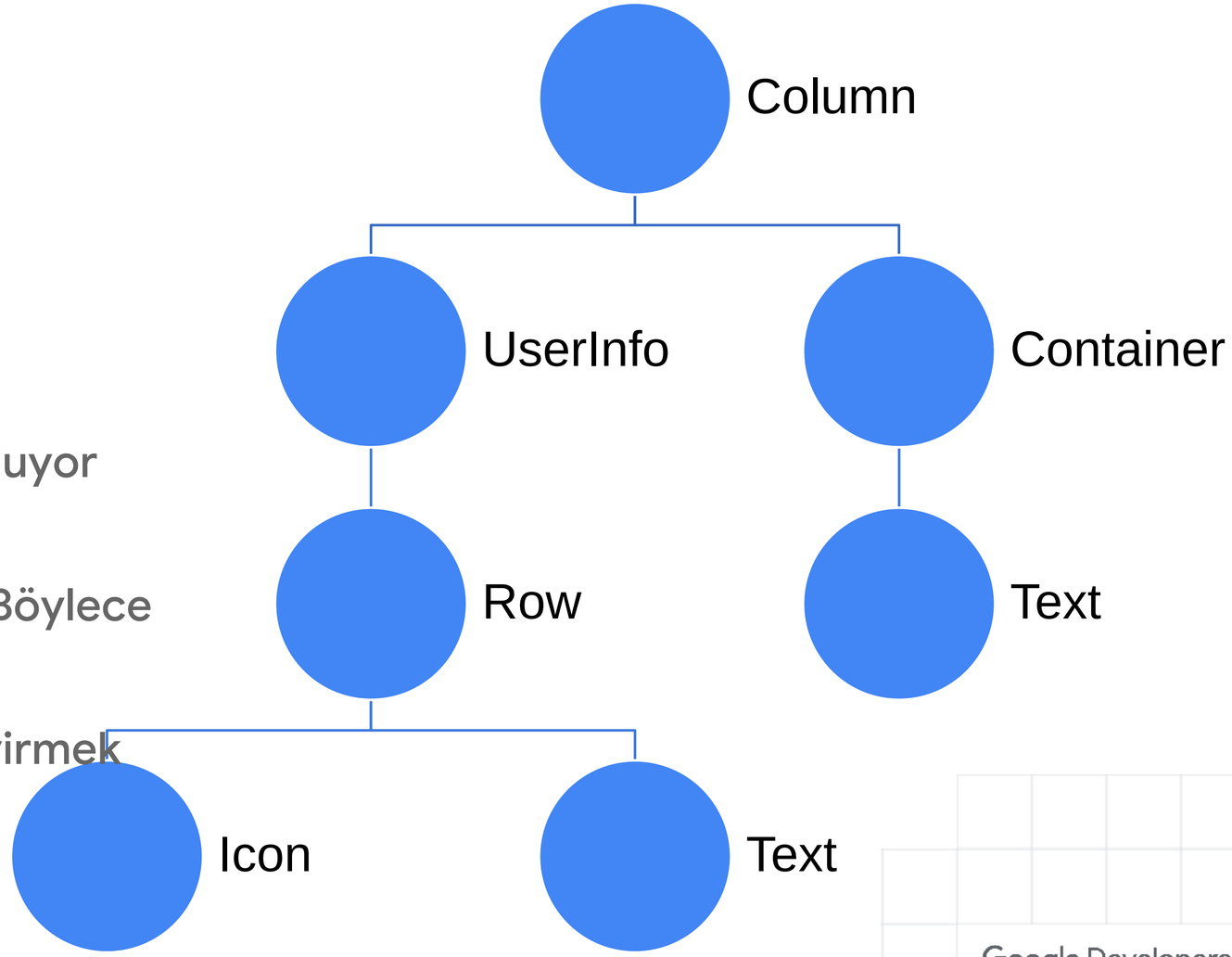
Widgetlar immutable

State classının fieldları state değişkenleri oluyor

Bu değişkenleri değiştirirken setState()'e verdiğimiz bir fonksiyon ile değiştiriyoruz. Böylece bu build()'i tetikliyor

build() fonksiyonunun görevi onları UI'a çevirmek

içinde başka işler yapmayın!



Kod Demosu

Asenkron Programlama

Beklemeyi kimse sevmez

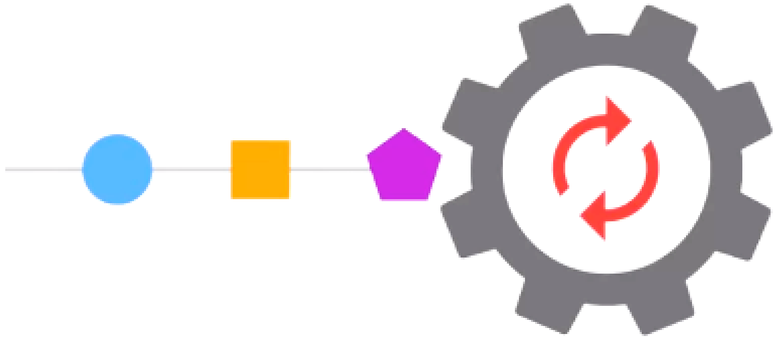
Beklememek için geçici bir şey verilmesi lazım

– Terzi Ali –

Gazihan Alankuş
1 kot pantolon
Paçaları yapılacak

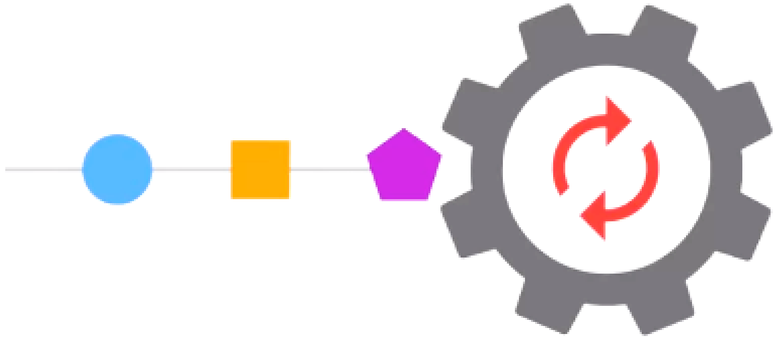
Future nesnesi

Event döngüsü ilgileniyor

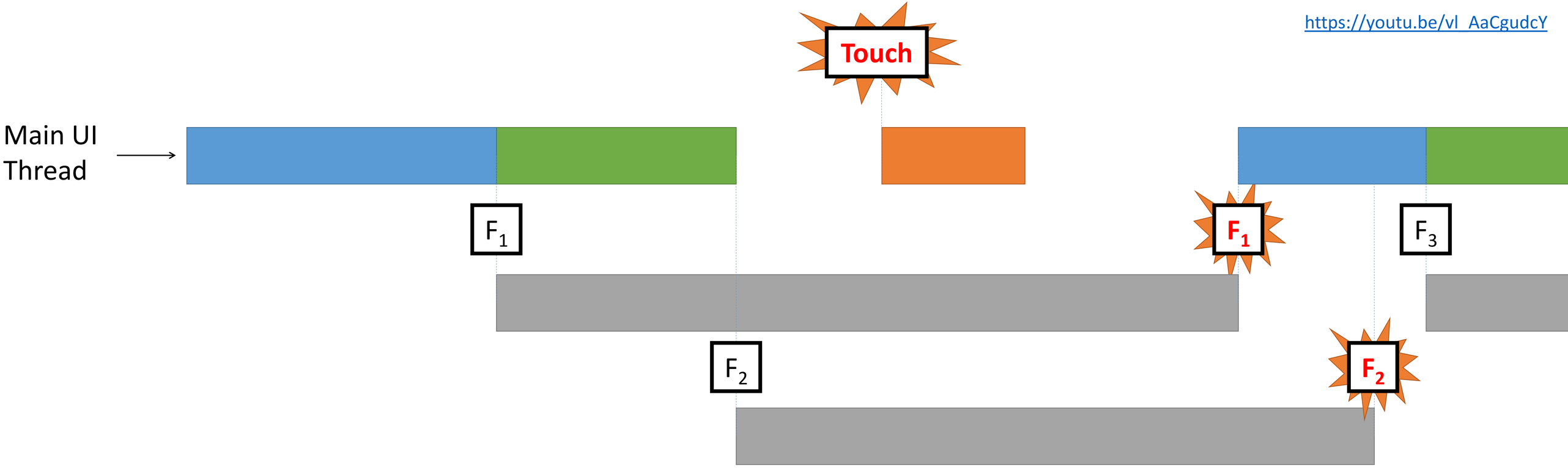


https://youtu.be/vl_AaCgudcY

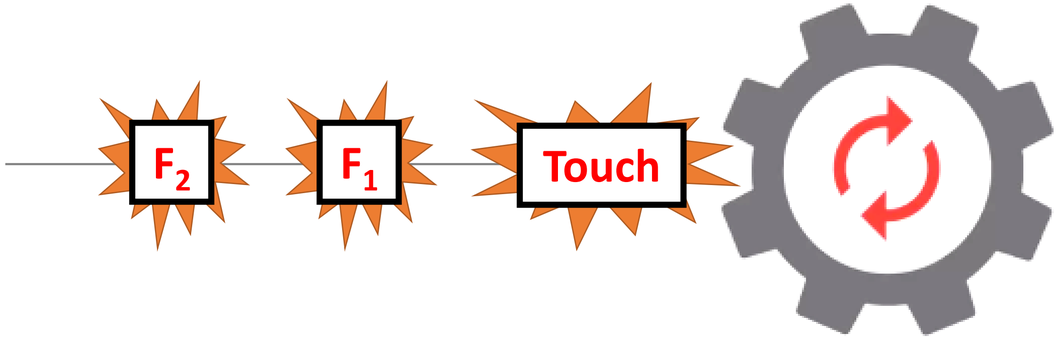
Diğer eventler gibi



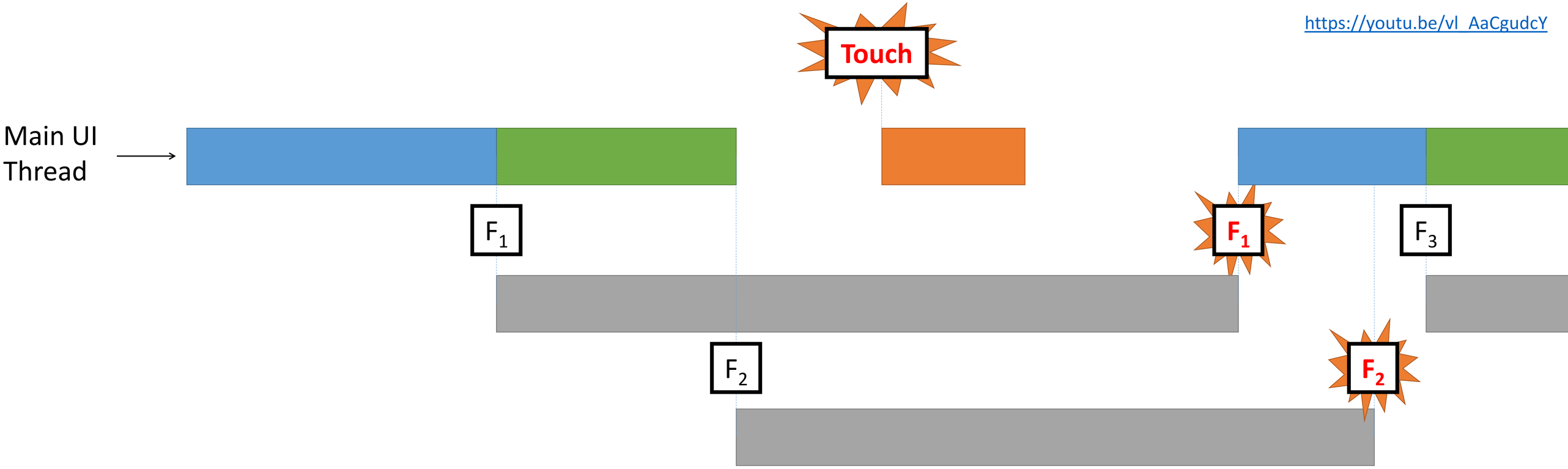
https://youtu.be/vl_AaCgudcY



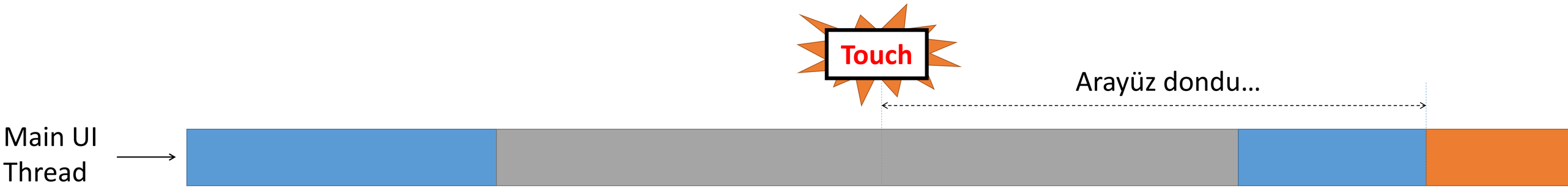
Diğer eventler gibi



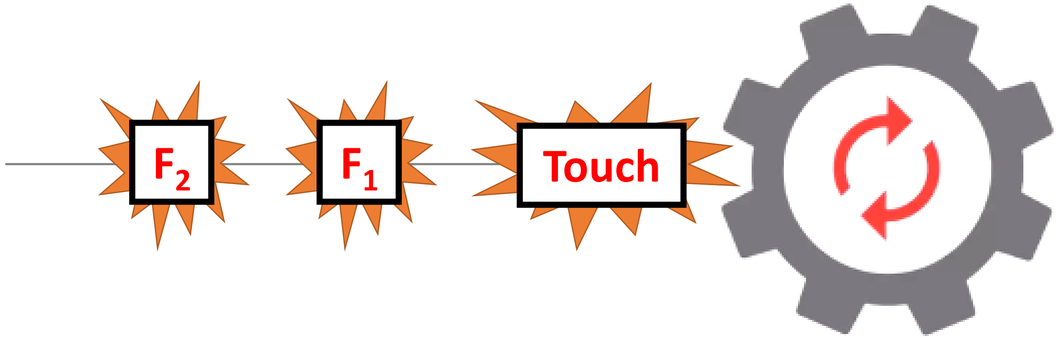
https://youtu.be/vl_AaCgudcY



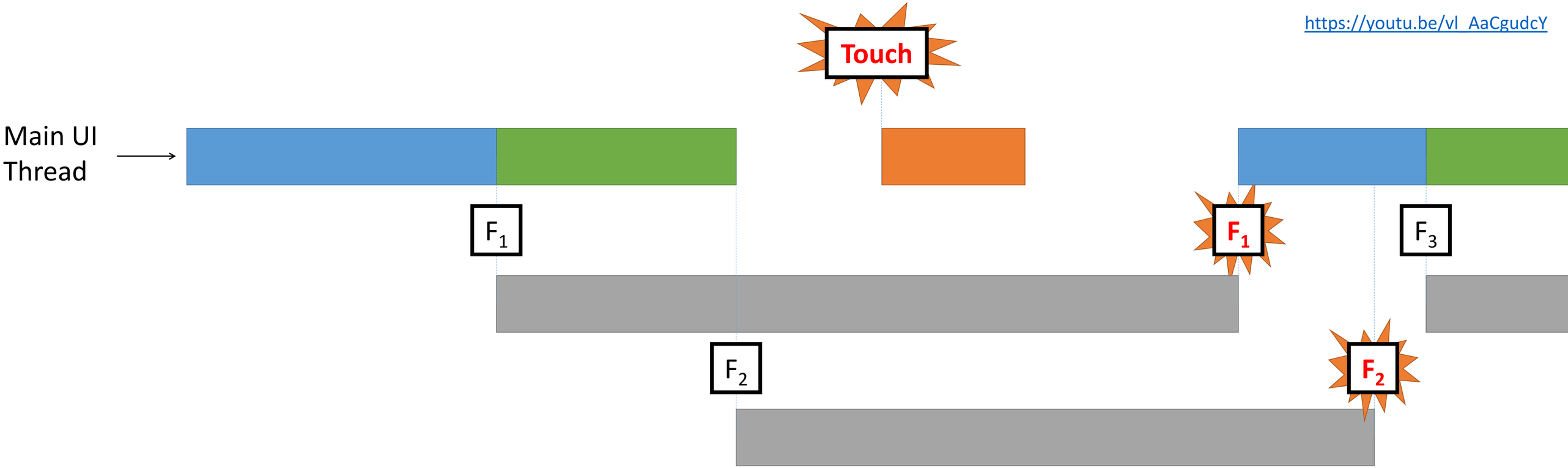
Senkron alternatif



Asenkron -> ayık app

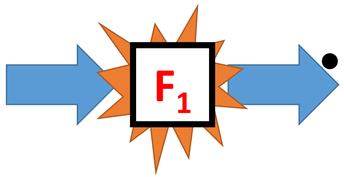


https://youtu.be/vl_AaCgudcY



Future: callback fonksiyonları için bir aracı

- Üreten
  - Şimdilik bu Future'ı al, sonuç hazır olunca onunla vereceğim.
  - *sonucu üretmek için çalışır*
  - Bitti, buyur sana sonuç



- Tüketen
  - Süper, hazır olunca kullanırım bunu
  - *bu arada başka işler yapar*
  - Süper, Future'a taktığım fonksiyon çalıştı. Sonucu onun içinde kullanayım.

Future

	Sync	Async
Tekli değer	T	Future<T>

```
int getVal() {
  return 1;
}

void main() {
  int val = getVal();
  print(val);
}
```

```
Future<int> getVal() {
  return Future.value(1);
}

void main() {
  getVal().then((int val) {
    print(val);
  });
}
```

Stream

	Sync	Async
Tekli değer	T	Future<T>
Çoklu değer	Iterable<T>	Stream<T>

```
Iterable<int> iterable = [1, 2, 3];

void main() {
  for (int i in iterable) {
    print(i);
  }
}
```

```
Stream<int> stream = Stream.fromIterable([1, 2, 3]);

void main() {
  stream.listen((i) {
    print(i);
  });
}
```

Future ve async/await

	Sync	Async
Tekli değer	T	Future<T>

```
int getVal() {
  return 1;
}

void main() {
  int val = getVal();
  print(val);
}
```

```
Future<int> getVal() async {
  return 1;
}

Future<void> main() async {
  int val = await getVal();
  print(val);
}
```

```
Future<int> getVal() {
  return Future.value(1);
}

void main() {
  getVal().then((int val) {
    print(val);
  });
}
```

Stream ve async/await

	Sync	Async
Tekli değer	T	Future<T>
Çoklu değer	Iterable<T>	Stream<T>

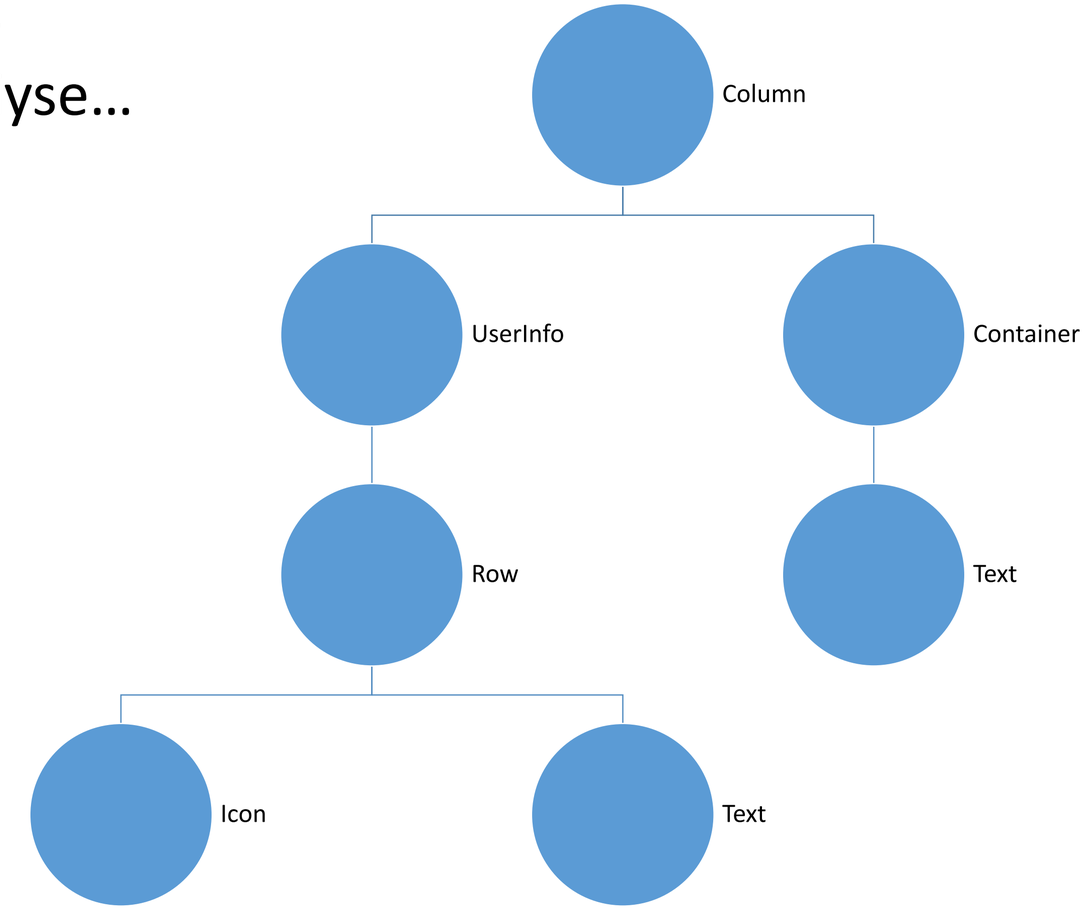
```dart
Iterable<int> iterable = [1, 2, 3];

void main() {
  for (int i in iterable) {
    print(i);
  }
}
```

```dart
Stream<int> stream = Stream.fromIterable([1, 2, 3]);

Future<void> main() async {
  await for (int i in stream) {
    print(i);
  }
}
```

```dart
Stream<int> stream = Stream.fromIterable([1, 2, 3]);

void main () {
  stream.listen((i) {
    print(i);
  });
}
```

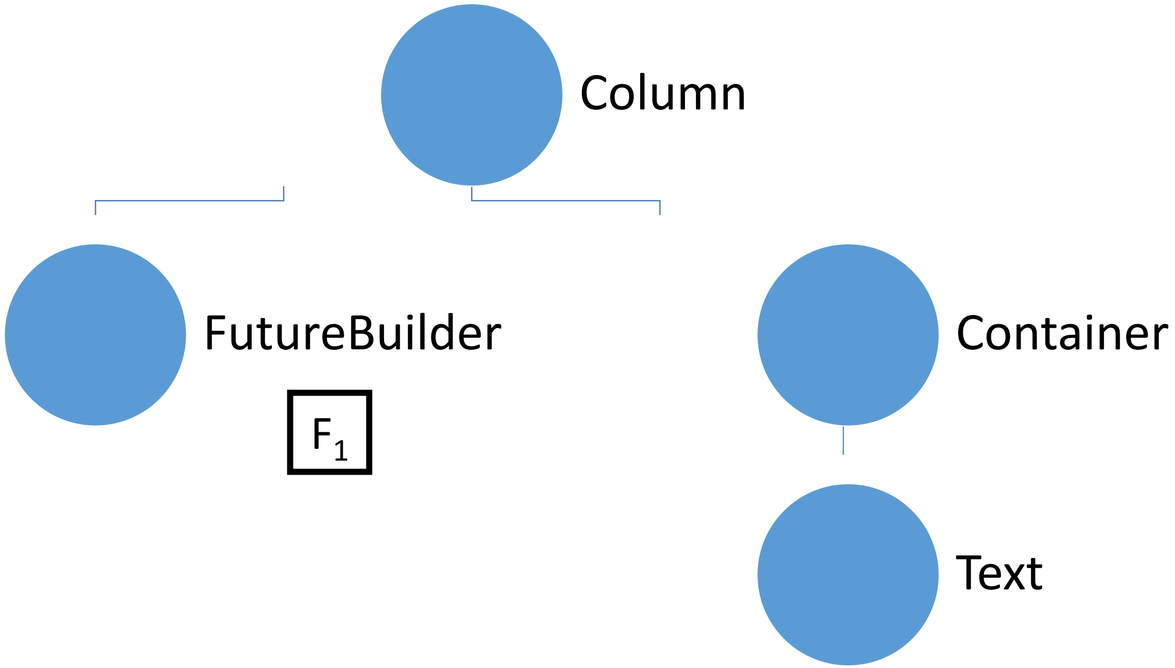
Flutter UI Mimarisi

- It’s all widgets!
  - Tabi veri geldiyse…



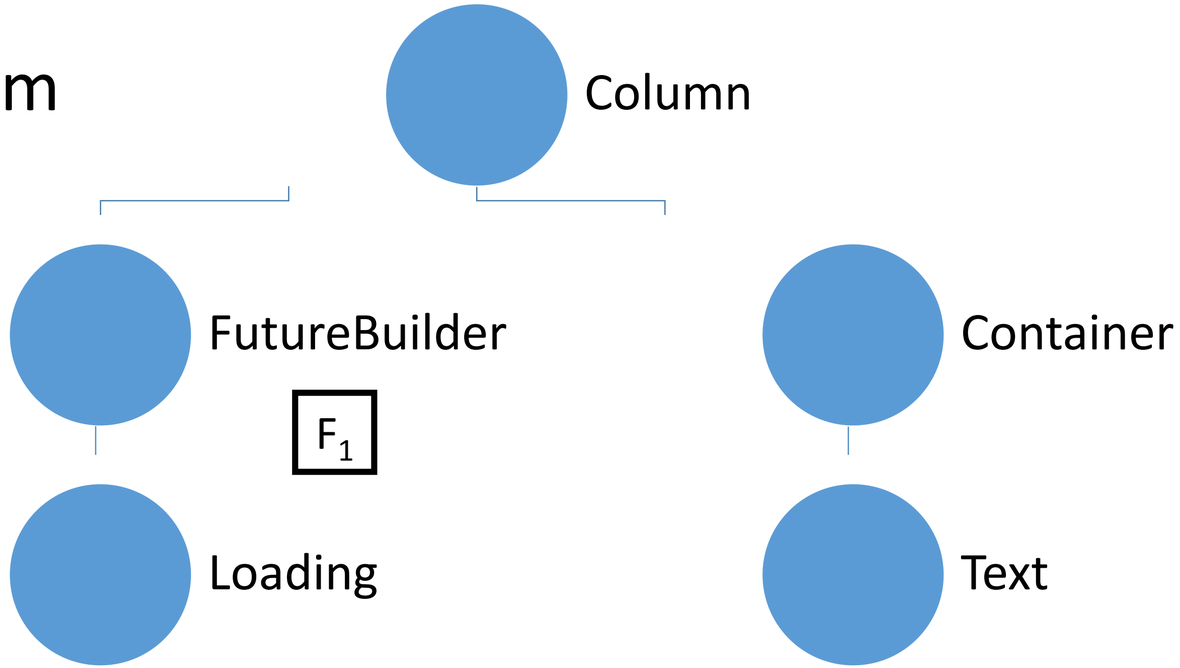
FutureBuilder widget

- Veri sonra geliyor



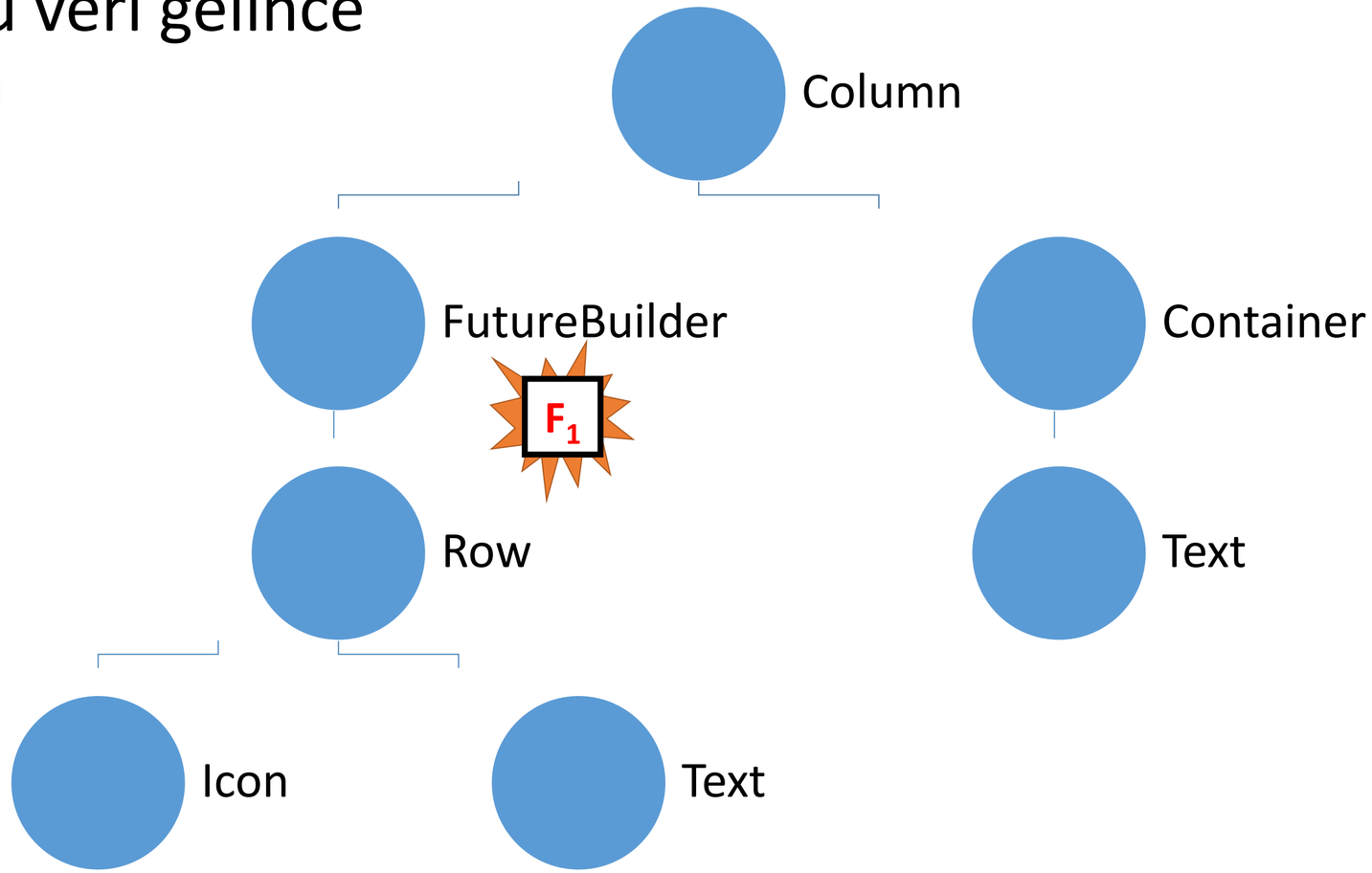
FutureBuilder widget

- Veri henüz yok,
  yükleniyor gösterelim



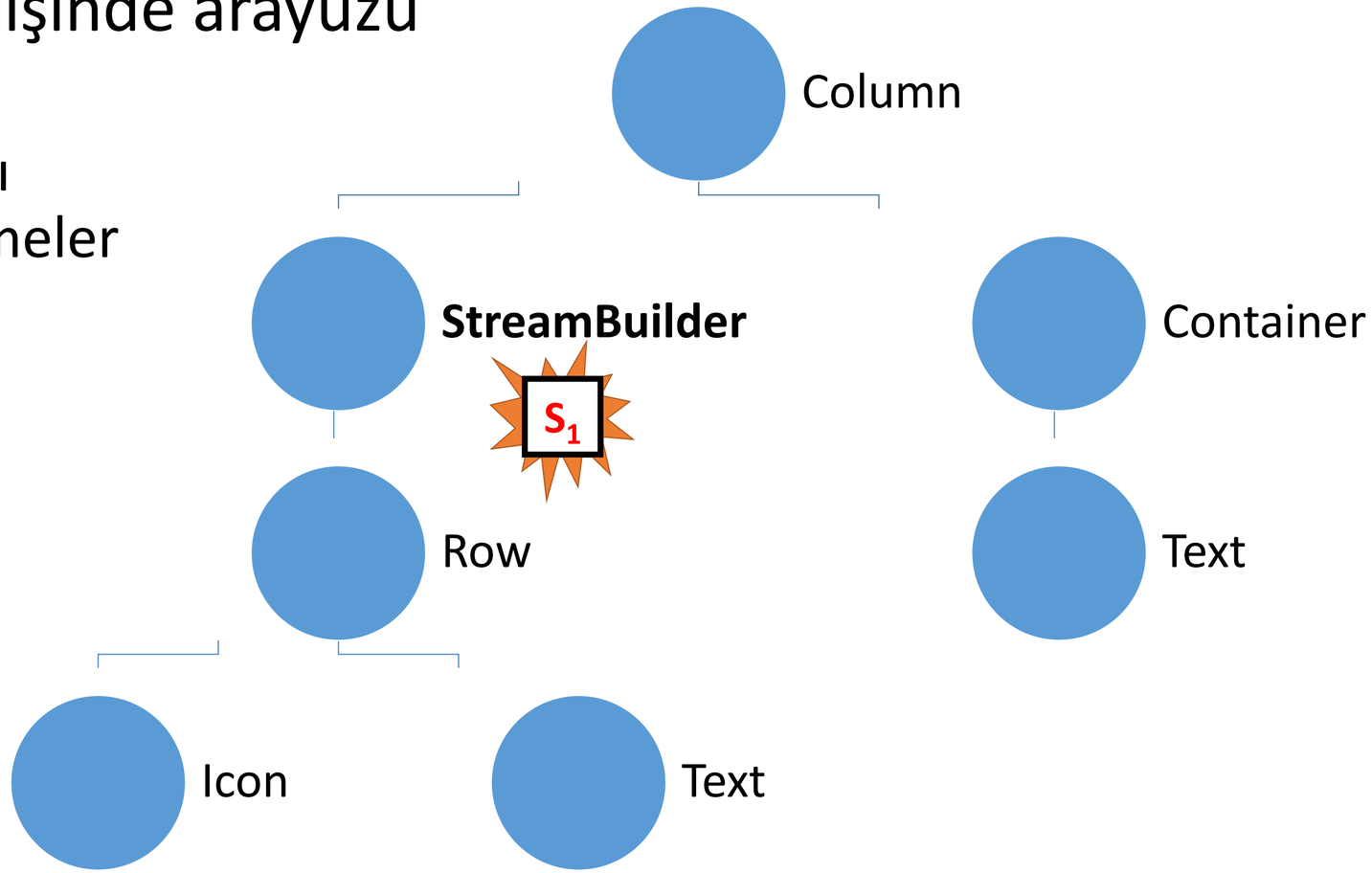
FutureBuilder widget

- Asıl arayüzü veri gelince inşa edelim



StreamBuilder widget

- Her veri gelişinde arayüzü inşa edelim
  - Örn. Canlı güncellemeler



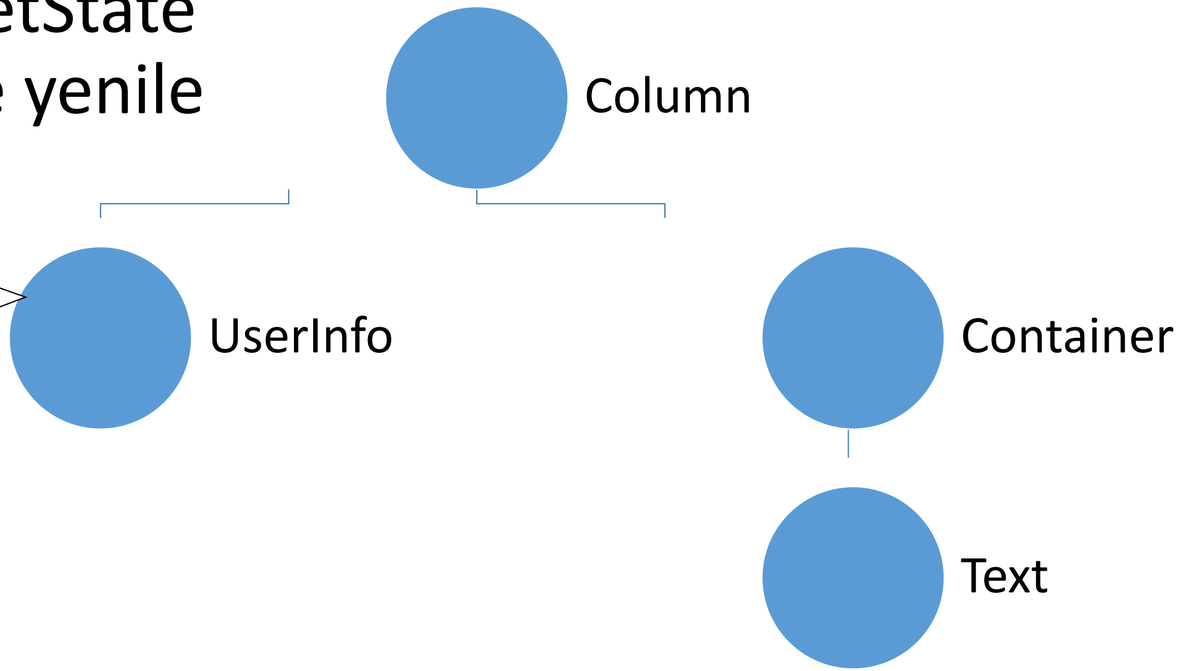
```dart
@override
Widget build(BuildContext context) {
  return Column(
    children: <Widget>[
      FutureBuilder(
        future: _f1,
        builder: (context, snapshot) {
          if (snapshot.hasError) {
            return Text("Error: ${snapshot.error}");
          }
          if (!snapshot.hasData) {
            return CircularProgressIndicator();
          }
          return Row(
            children: <Widget>[
              Icon(Icons.person),
              Text(snapshot.data)
            ],
          );
        },
      ),
      Container(
        child: Text("Devfest Veneto"),
      )
    ],
  );
}
```

```dart
@override
Widget build(BuildContext context) {
  return Column(
    children: <Widget>[
      StreamBuilder(
        stream: widget.nameStream,
        builder: (context, snapshot) {
          if (snapshot.hasError) {
            return Text("Error: ${snapshot.error}");
          }
          if (!snapshot.hasData) {
            return CircularProgressIndicator();
          }
          return Row(
            children: <Widget>[
              Icon(Icons.person),
              Text(snapshot.data)
            ],
          );
        },
      ),
      Container(
        child: Text("Devfest Veneto"),
      )
    ],
  );
}
```

Alternatif: Future/Stream dinle ve setState kullan

- Ağacın bu kısmını setState ile yenile

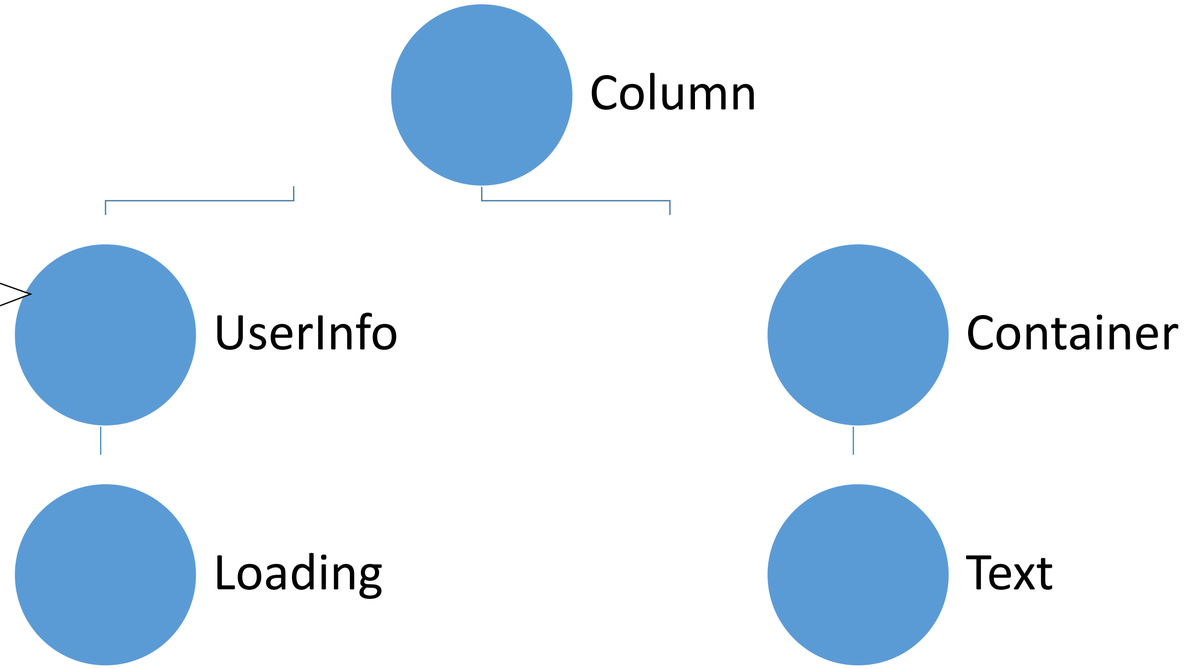
```
initState() {
  startAsyncJob()
}

startAsyncJob() async {
  var r = await obtain();
  setState(() {
    state = r;
  })
}

build() {
  if (state == null) {
    return Loading();
  } else {
    return createUI(state);
  }
}
```



Alternatif: Future/Stream dinle ve setState kullan

- State başta null

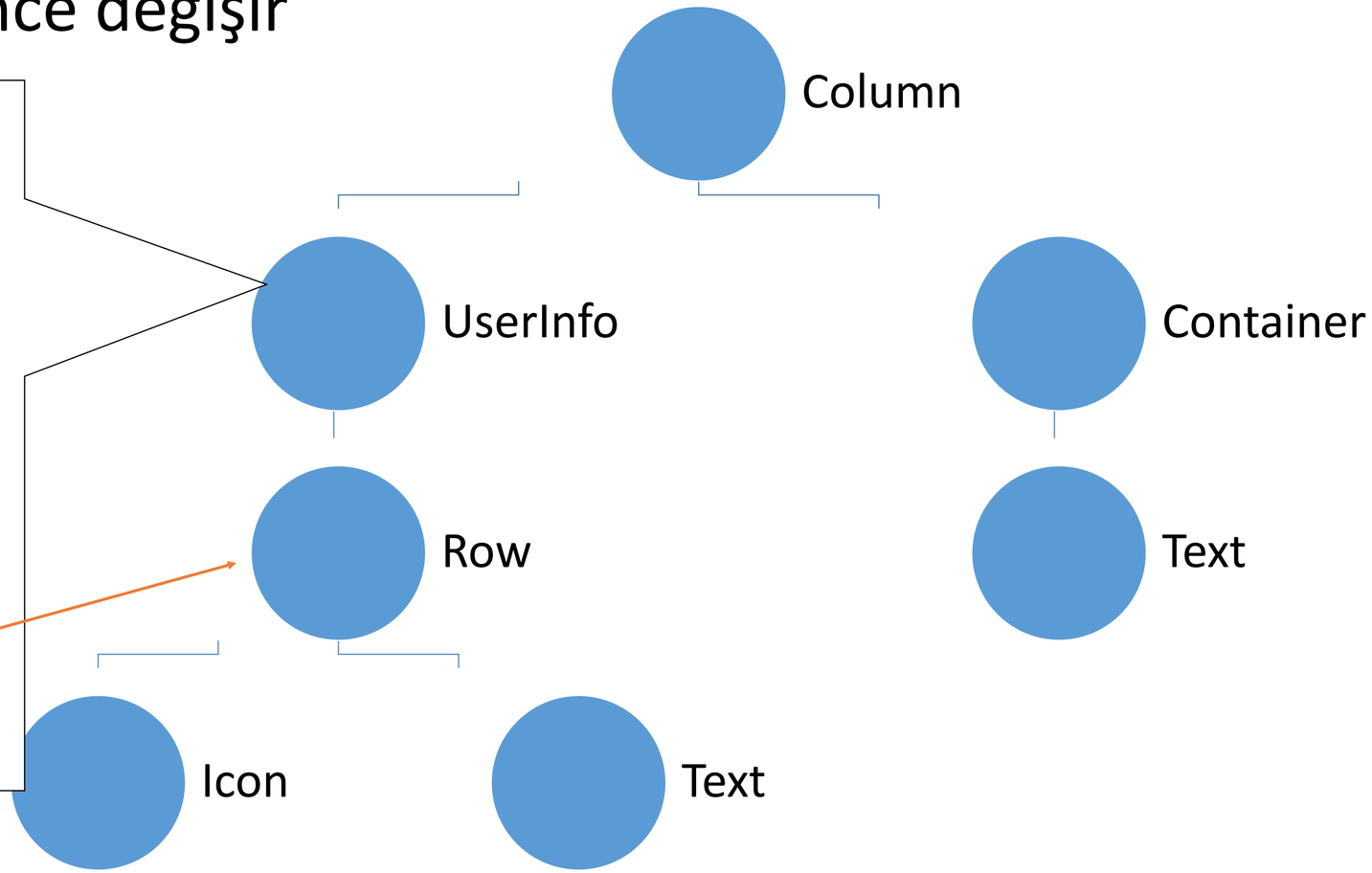
```
initState() {
  startAsyncJob()
}

startAsyncJob() async {
  var r = await obtain();
  setState(() {
    state = r;
  })
}

build() {
  if (state == null) {
    return Loading();
  } else {
    return createUI(state);
  }
}
```



Alternatif: Future/Stream dinle ve setState kullan

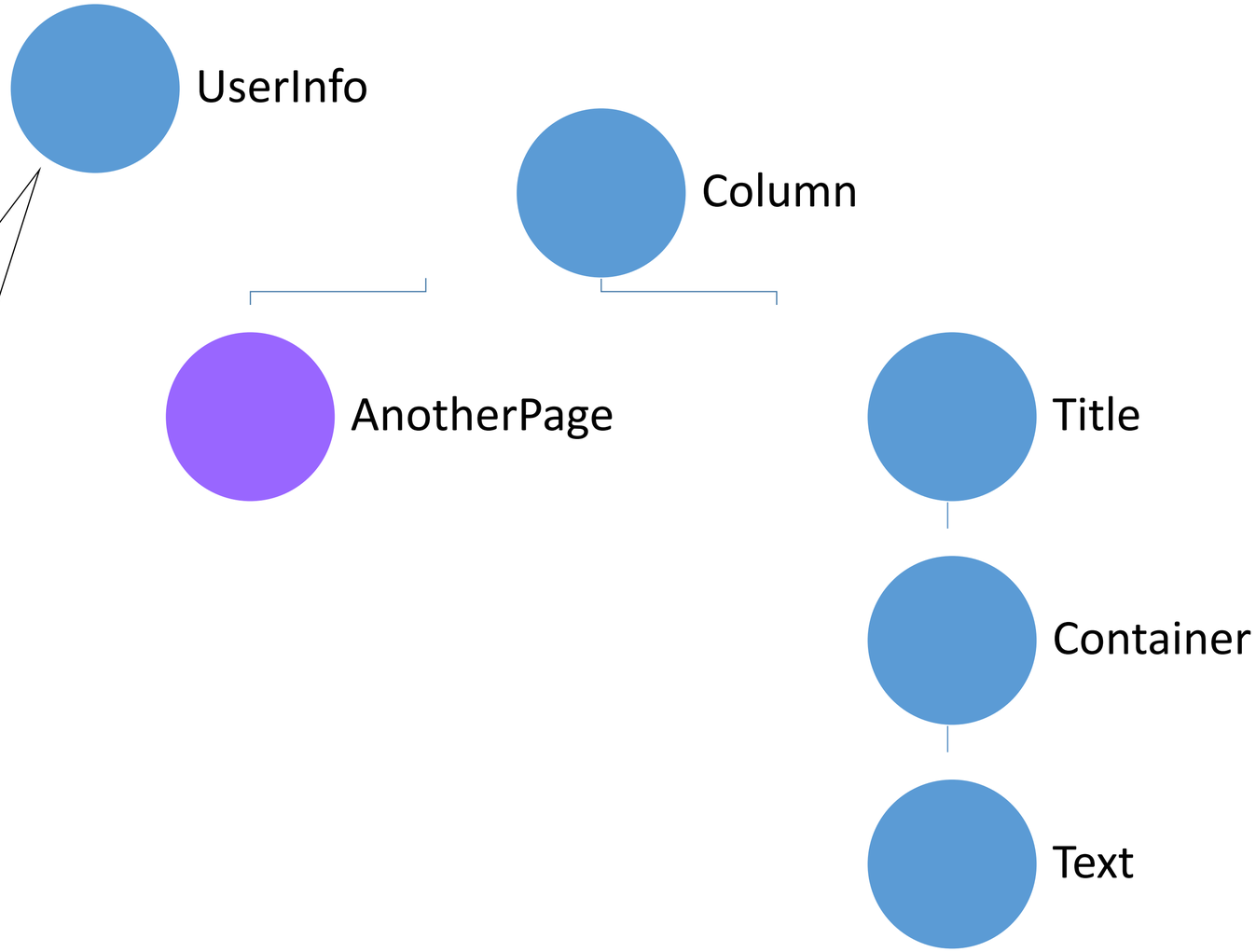
- Future bitince değişir

```
initState() {
  startAsyncJob()
}

startAsyncJob() async {
  var r = await obtain();
  setState(() {
    state = r;
  })
}

build() {
  if (state == null) {
    return Loading();
  } else {
    return createUI(state);
  }
}
```



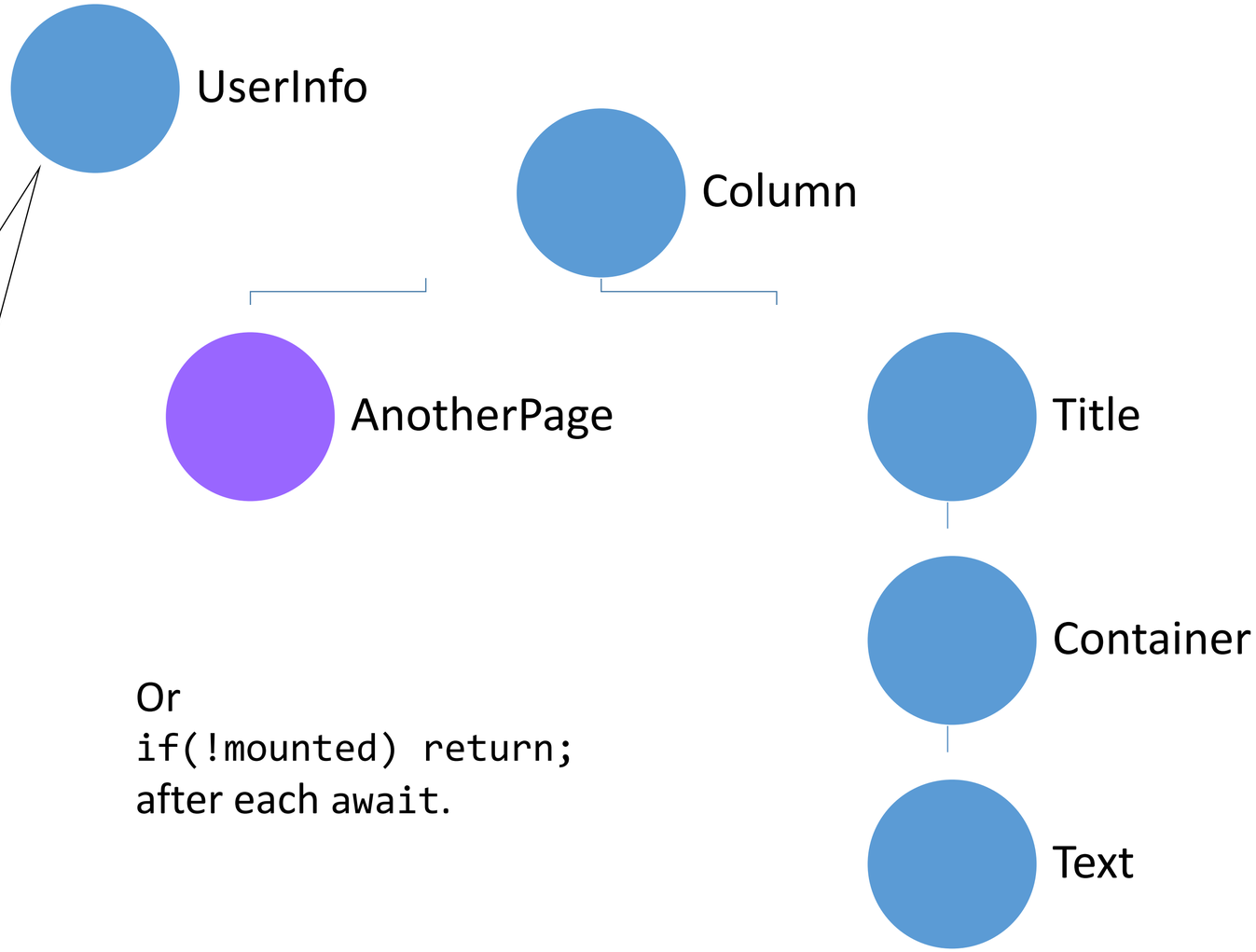
Dikkat: widget unmount olabilir

```
initState() {
  startAsyncJob()
}

startAsyncJob() async {
  var r = await obtain();
  setState(() {
    state = r;
  })
}

build() {
  if (state == null) {
    return Loading();
  } else {
    return createUI(state);
  }
}
```



Dikkat: widget unmount olabilir

```
initState() {
  startAsyncJob()
}

startAsyncJob() async {
  var r = await obtain();
  setStateIfMounted(() {
    state = r;
  })
}

build() {
  if (state == null) {
    return Loading();
  } else {
    return createUI(state);
  }
}

setStateIfMounted(f) {
 if (mounted) setState(f);
}
```



Or
`if(!mounted) return;`
after each `await`.

Unmount’a karşı koruma

```
class _FastCounterState extends State<FastCounter> {
  int counter = 0;

  @override
  void initState() {
    super.initState();
    startCounter();
  }

  Future<void> startCounter() async {
    while(true) {
      await Future.delayed(Duration(milliseconds: 500));
      setStateIfMounted(() {
        ++counter;
      });
    }
  }

  void setStateIfMounted(f) {
    if (mounted) setState(f);
  }

  @override
  Widget build(BuildContext context) {
    return Text("Fast counter: $counter");
  }
}
```

UI ve async kod ayrı, daha temiz

- Dart'ta iç içe fonksiyonlar var
- Esas arayüz kodu ve FutureBuilder/ StreamBuilder kodu ayrı

```
Widget build(BuildContext context) {
  Widget reallyBuild(DocumentSnapshot data) {
    return TabBarView(controller: _tabController, children: <Widget>[
      PumpTab(data),
      DriverTab(data),
    ]);
  }

  return StreamBuilder<DocumentSnapshot>(
    stream: _snapshotsOfMeasurementDetails,
    builder:
      (BuildContext context, AsyncSnapshot<DocumentSnapshot> snapshot) {
      if (snapshot.hasError) {
        return Text('Error: ${snapshot.error}');
      }
      if (snapshot.hasData) {
        return reallyBuild(snapshot.data);
      }
      return CircularProgressIndicator();
    },
  );
}
```

build() fonksiyonunda yan etki yok!

- build() çeşitli nedenlerden defalarca çağırılabilir
- İçerisinde network işi yapma

```
class _MyWidgetState extends State<MyWidget> {
  @override
  Widget build(BuildContext context) {
    return FutureBuilder(
      future: httpCall(),
      builder: (context, snapshot) {
        // create some layout here
      },
    );
  }
}
```

```
class _MyWidgetState extends State<MyWidget> {
  Future<int> _future;

  @override
  void initState() {
    super.initState();
    _future = httpCall();
  }

  @override
  Widget build(BuildContext context) {
    return FutureBuilder(
      future: _future,
      builder: (context, snapshot) {
        // create some layout here
      },
    );
  }
}
```

https://stackoverflow.com/questions/52249578/how-to-deal-with-unwanted-widget-build

Happy async programming!

https://www.njpartyentertainment.com/jugglers-juggling-stilt-walkers-nj.html

Resources

- Andrew Brogdon, Async coding with Dart – Flutter in Focus
  - https://www.youtube.com/watch?v=vl_AaCgudcY&list=PLjxrf2q8roU2HdJQDjJzOeO6J3FoFLWr2&index=15&t=0s
- Florian Loitsch, Async in Dart (Dart Developer Summit 2015)
  - https://www.youtube.com/watch?v=MUDOIAssBDs
- Asynchronous Programming in Flutter with Dart (IUE Presentation)
  - https://www.youtube.com/watch?v=yyvTrXCDdYg