Releasing faster with Kotlin multiplatform

@jw@jakewharton.com

Previously On Cash App

Previously On Cash App

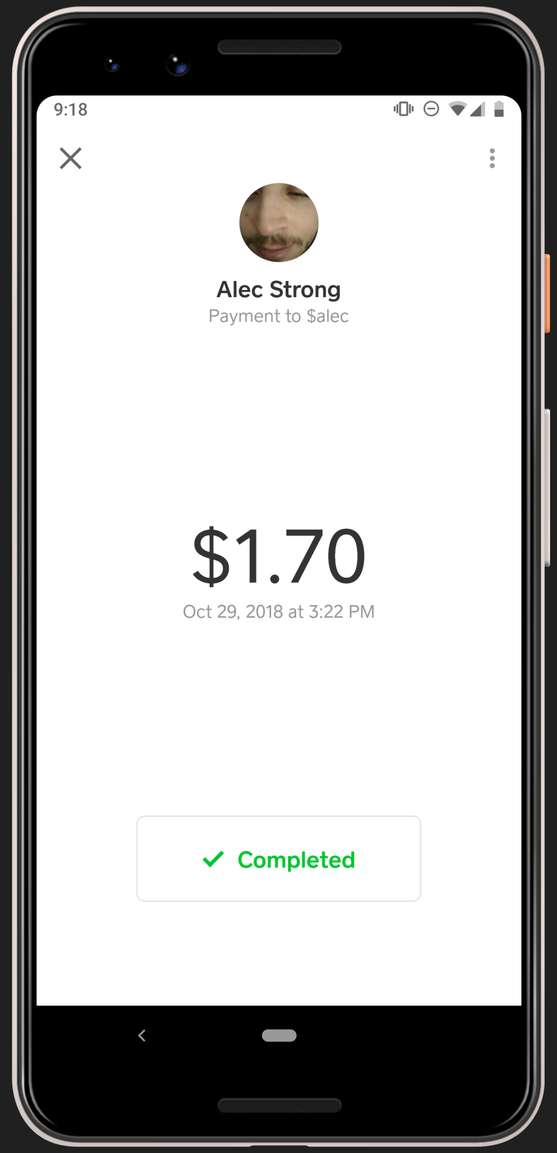
- Android, iOS, and web apps all developed natively

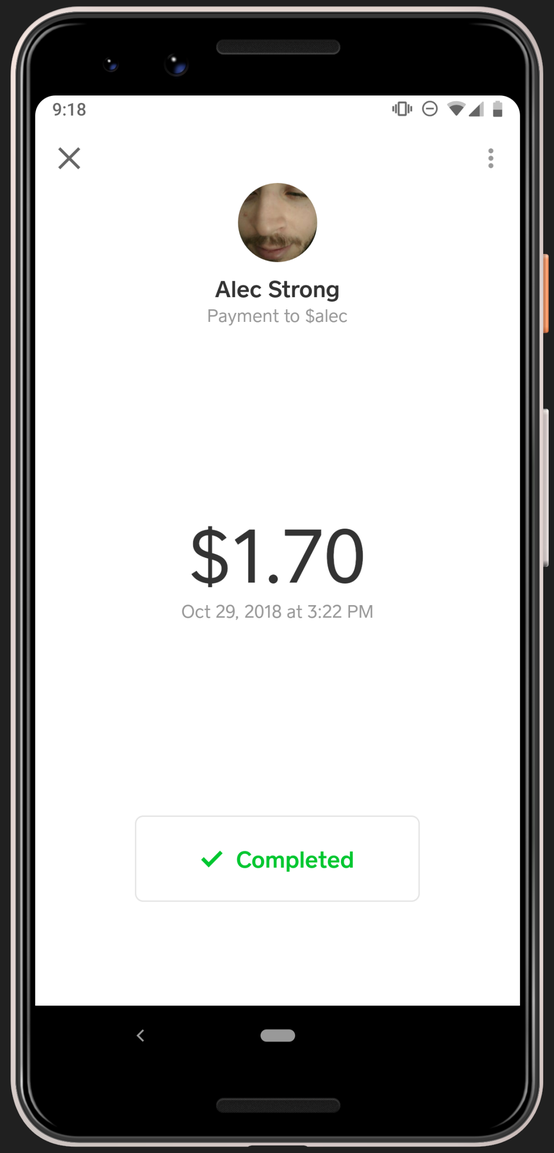
Previously On Cash App

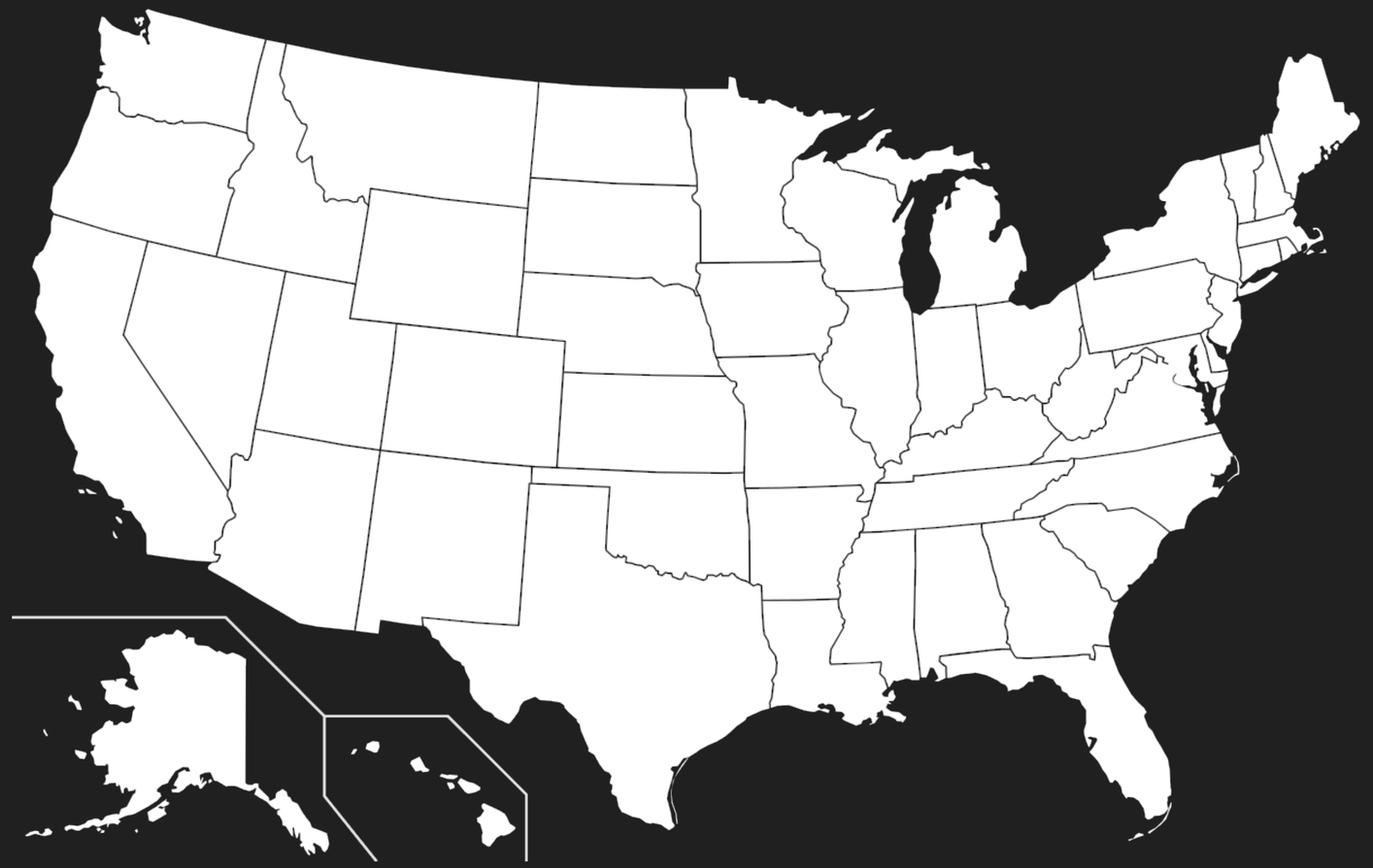
- Android, iOS, and web apps all developed natively
- Two week release trains for mobile apps

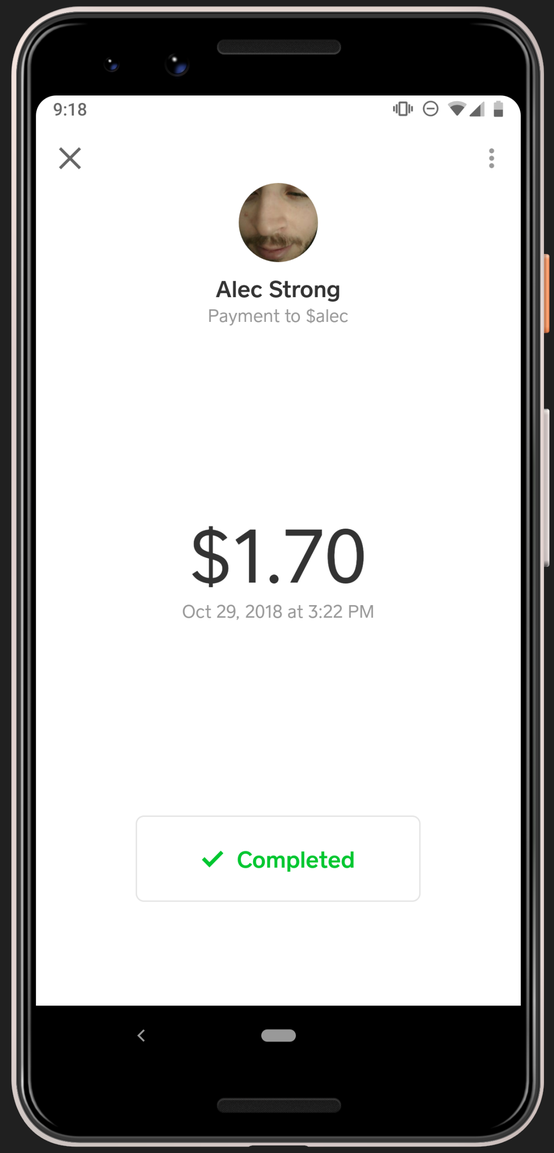
Previously On Cash App

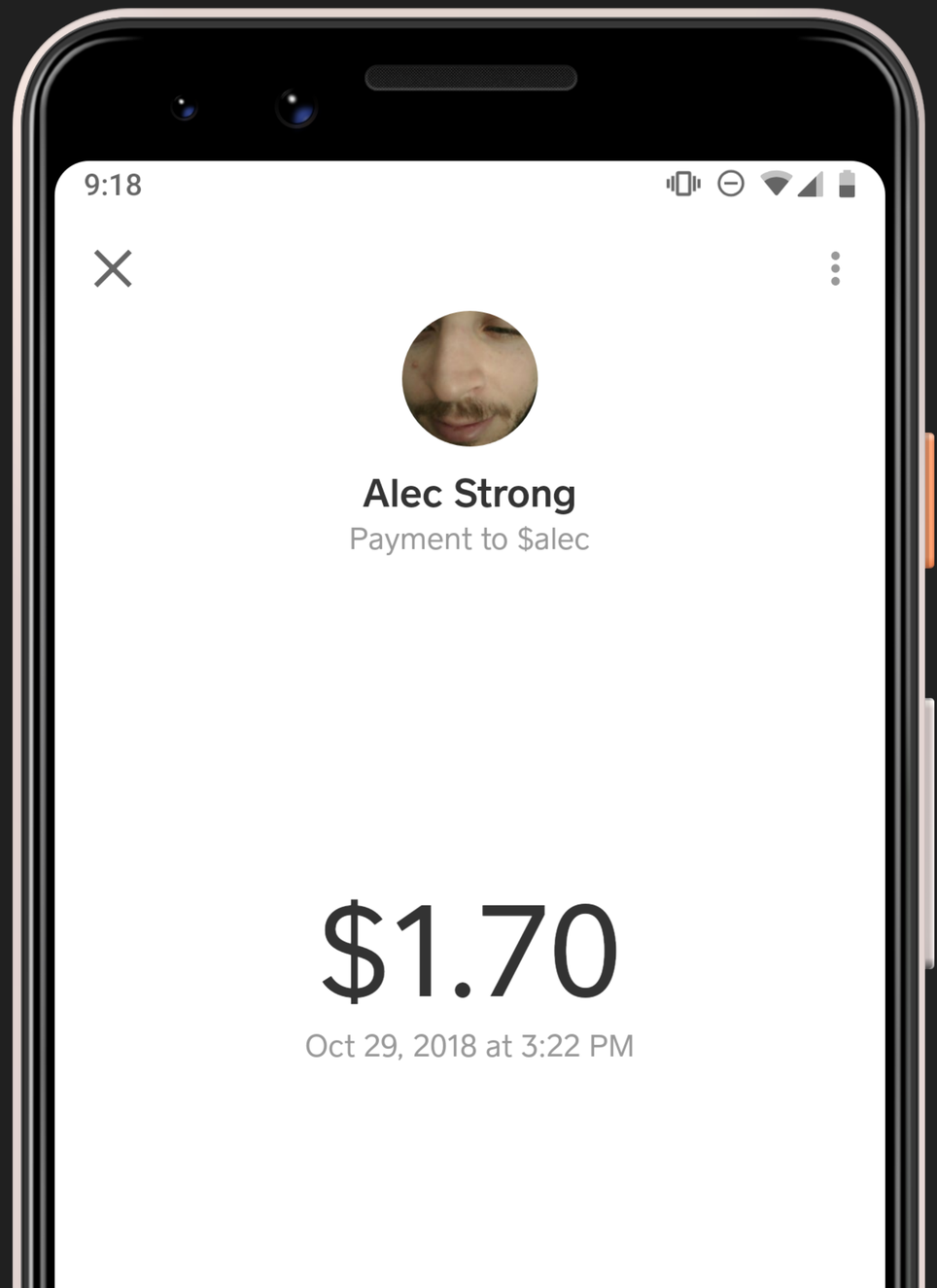
- Android, iOS, and web apps all developed natively
- Two week release trains for mobile apps
- Rollout over one to two week period

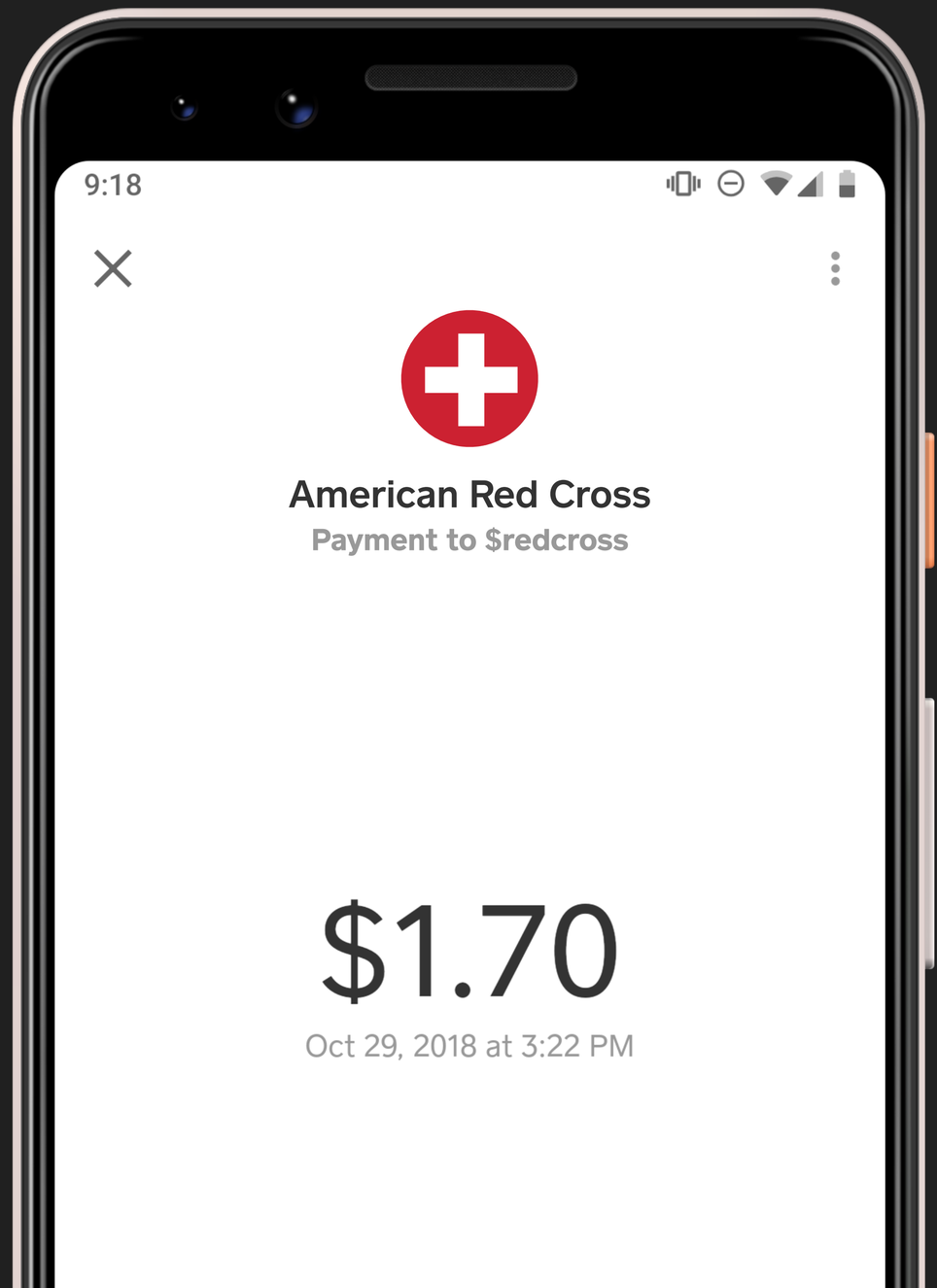

9:18
Alec Strong
Payment to $alec
$1.70
Oct 29, 2018 at 3:22 PM
Completed

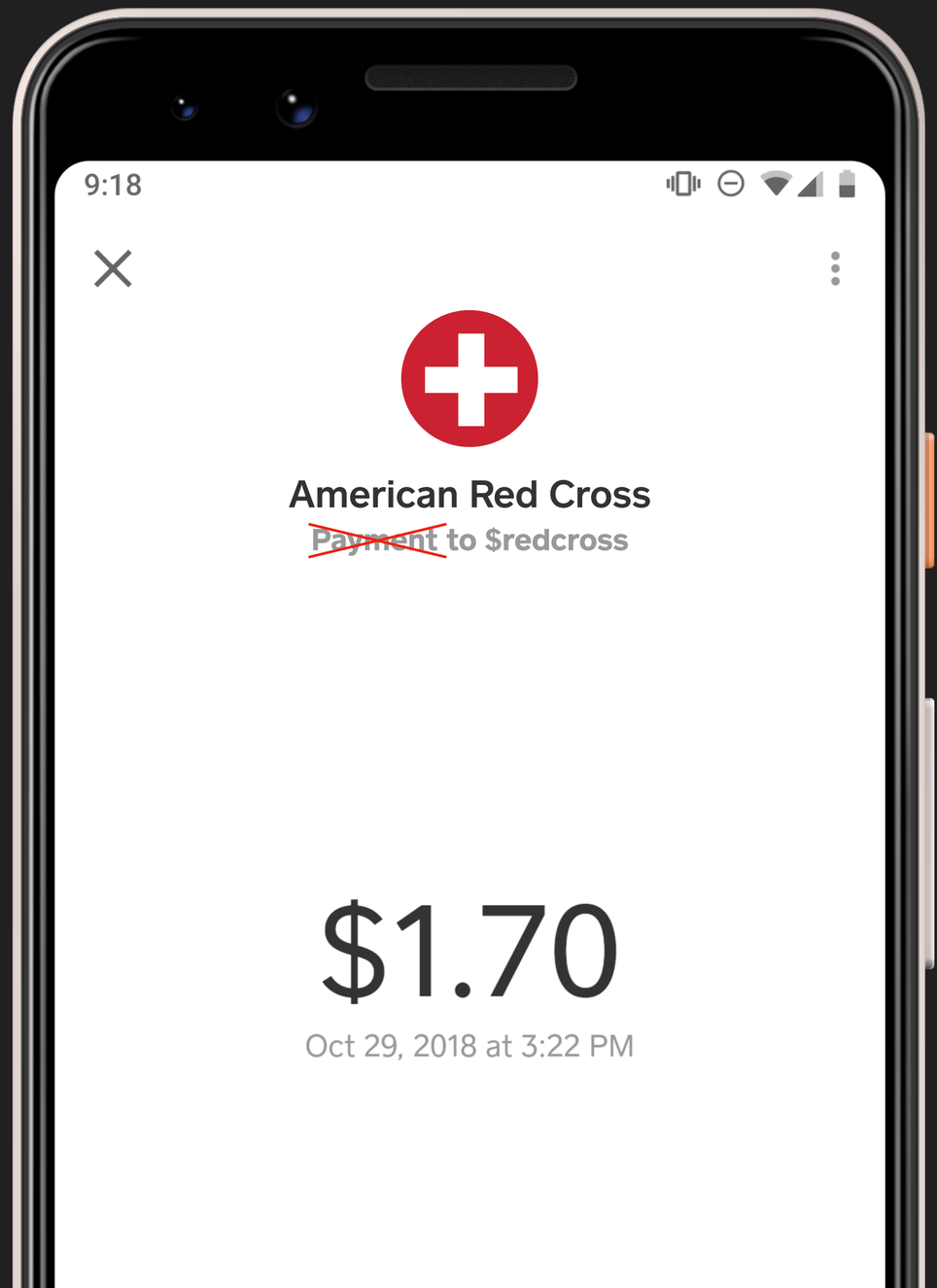

9:18
Alec Strong
Payment to $alec
$1.70
Oct 29, 2018 at 3:22 PM
Completed

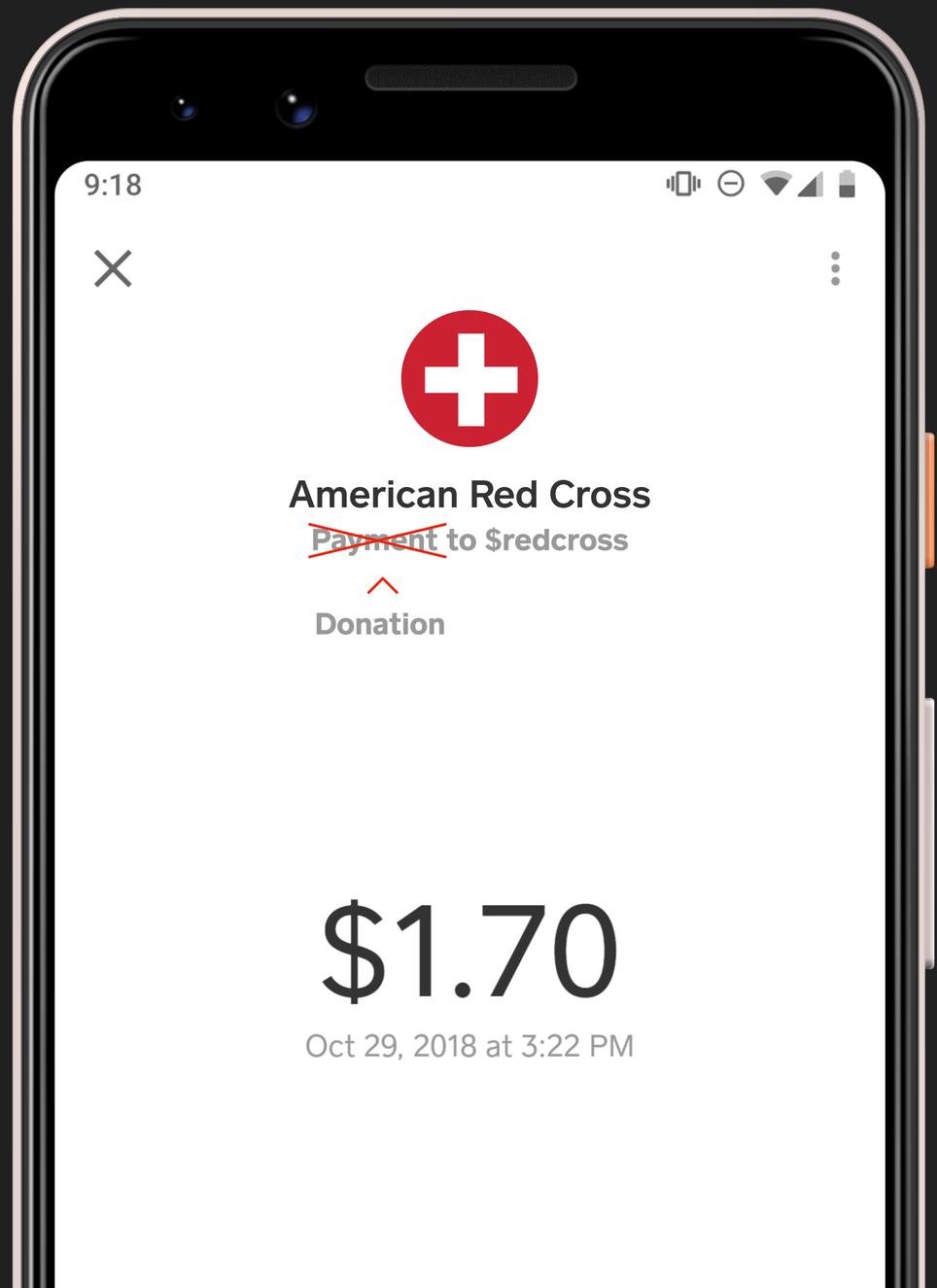


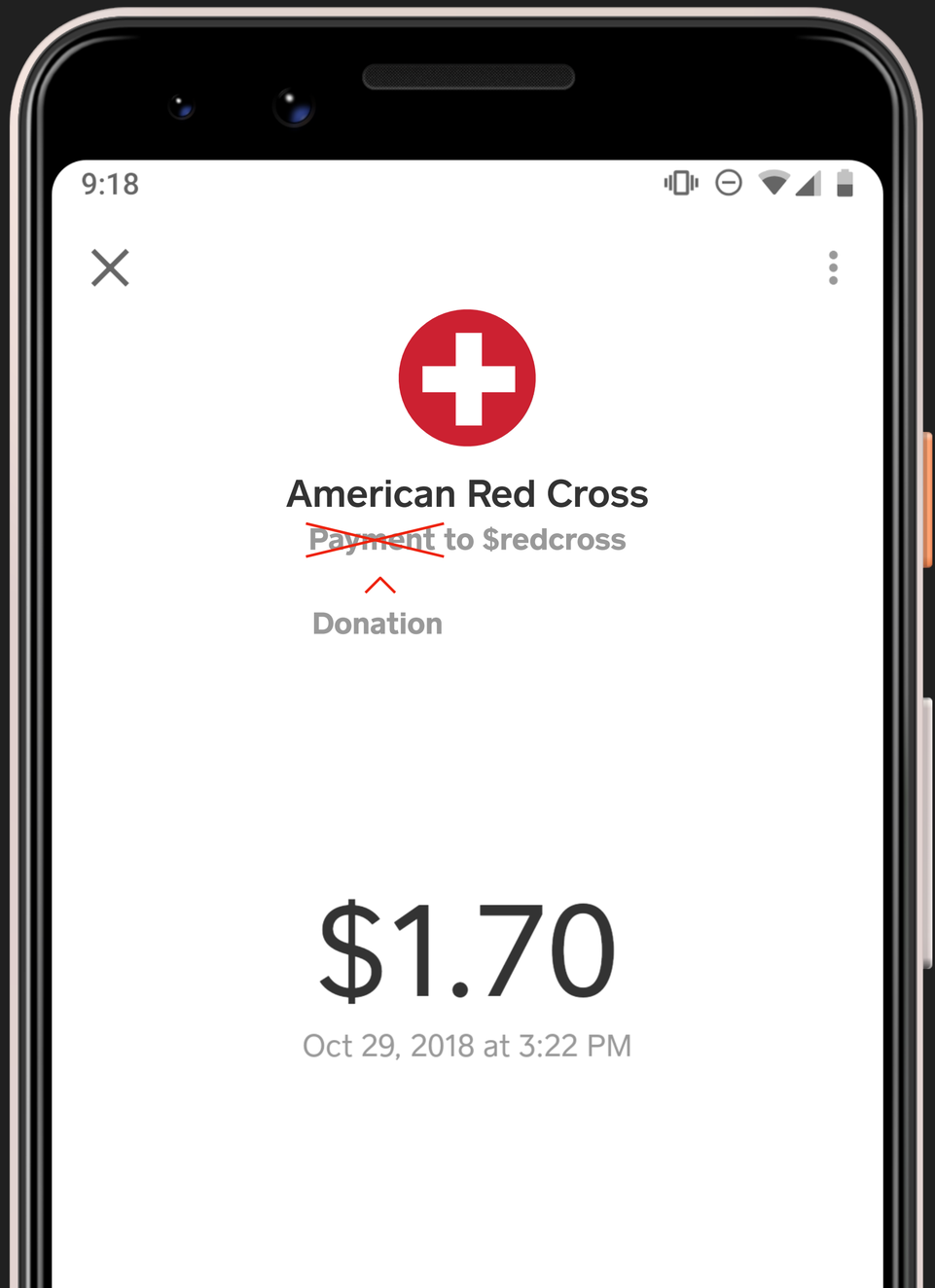

9:18
Alec Strong
Payment to $alec
$1.70
Oct 29, 2018 at 3:22 PM
Completed


9:18
Alec Strong
Payment to $alec
$1.70
Oct 29, 2018 at 3:22 PM

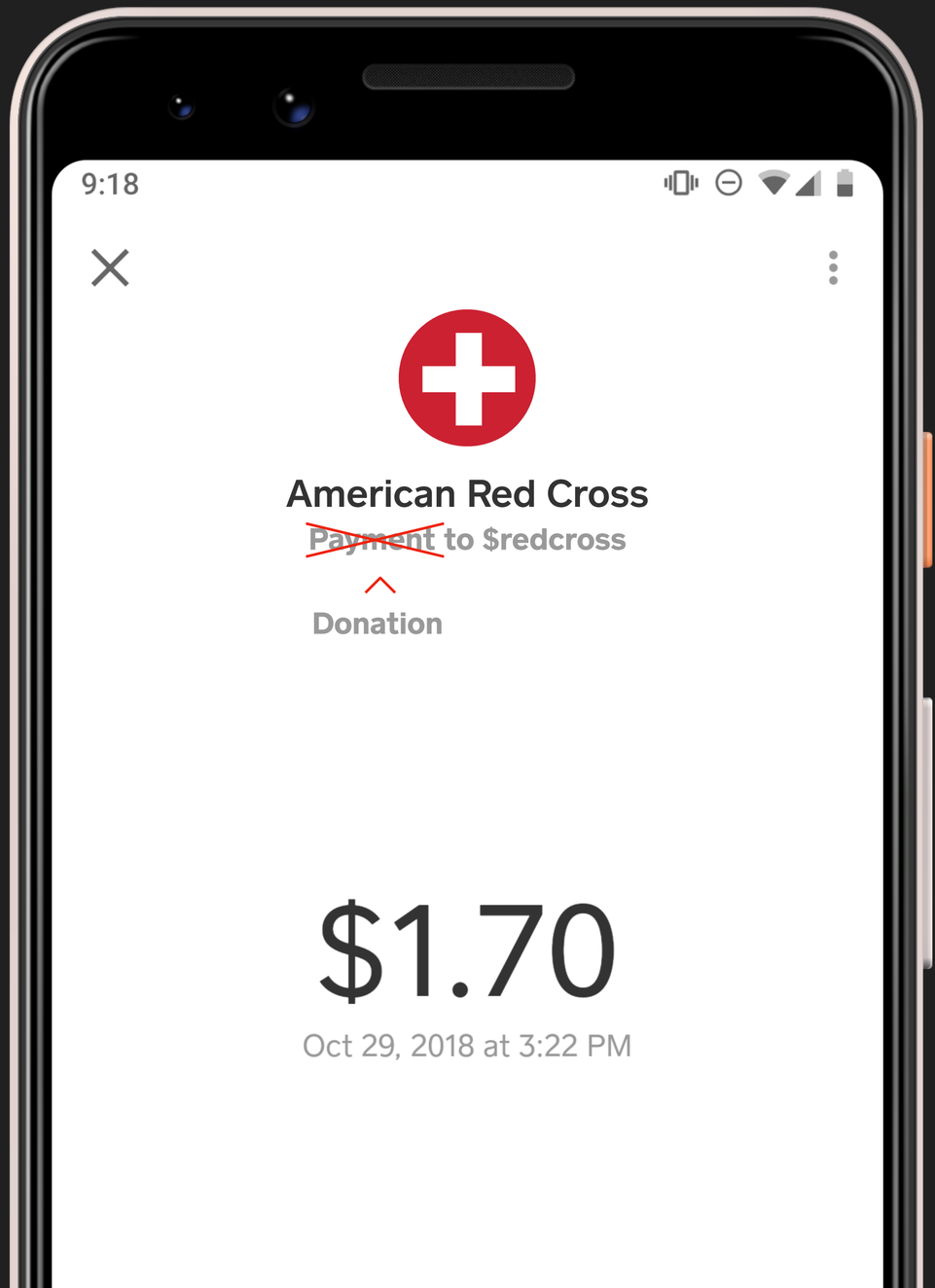

9:18
American Red Cross
Payment to $redcross
$1.70
Oct 29, 2018 at 3:22 PM


9:18
American Red Cross
Payment to $redcross
$1.70
Oct 29, 2018 at 3:22 PM


9:18
American Red Cross
Payment to $redcross
Donation
$1.70
Oct 29, 2018 at 3:22 PM


9:18
American Red Cross
Payment to $redcross
Donation
$1.70
Oct 29, 2018 at 3:22 PM

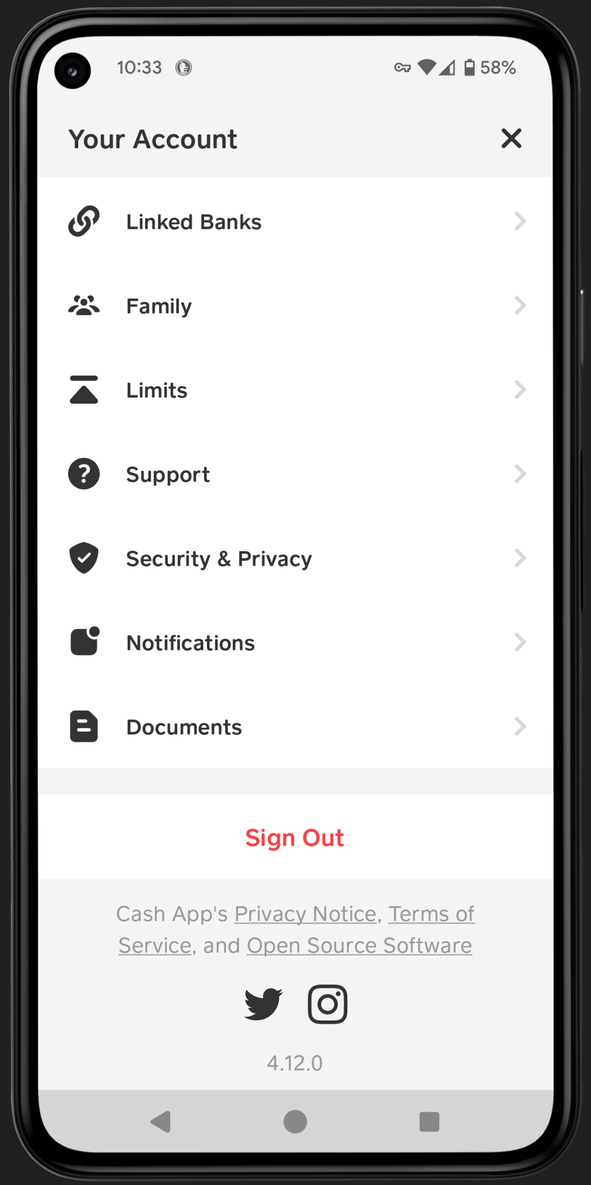
~1 day

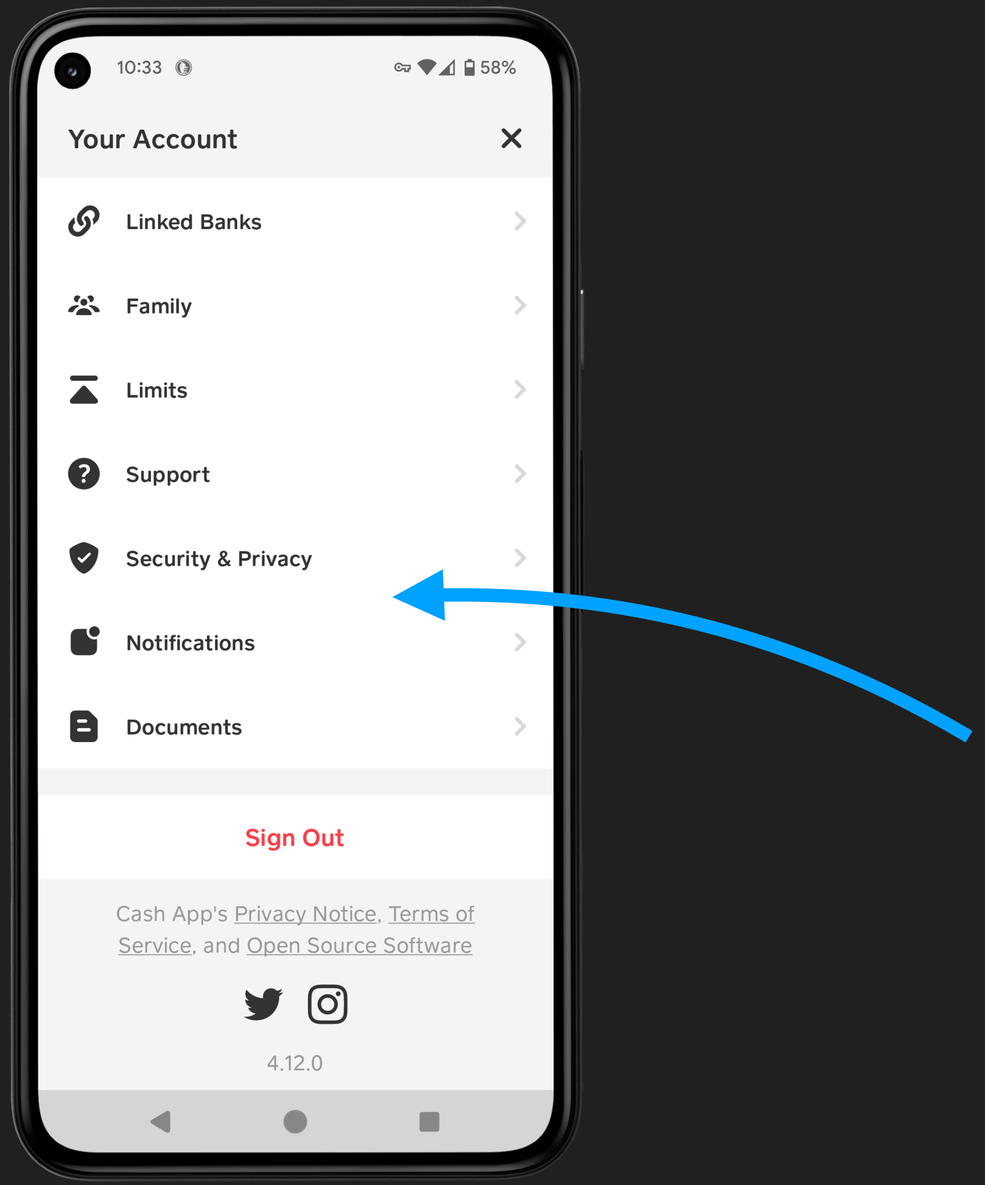


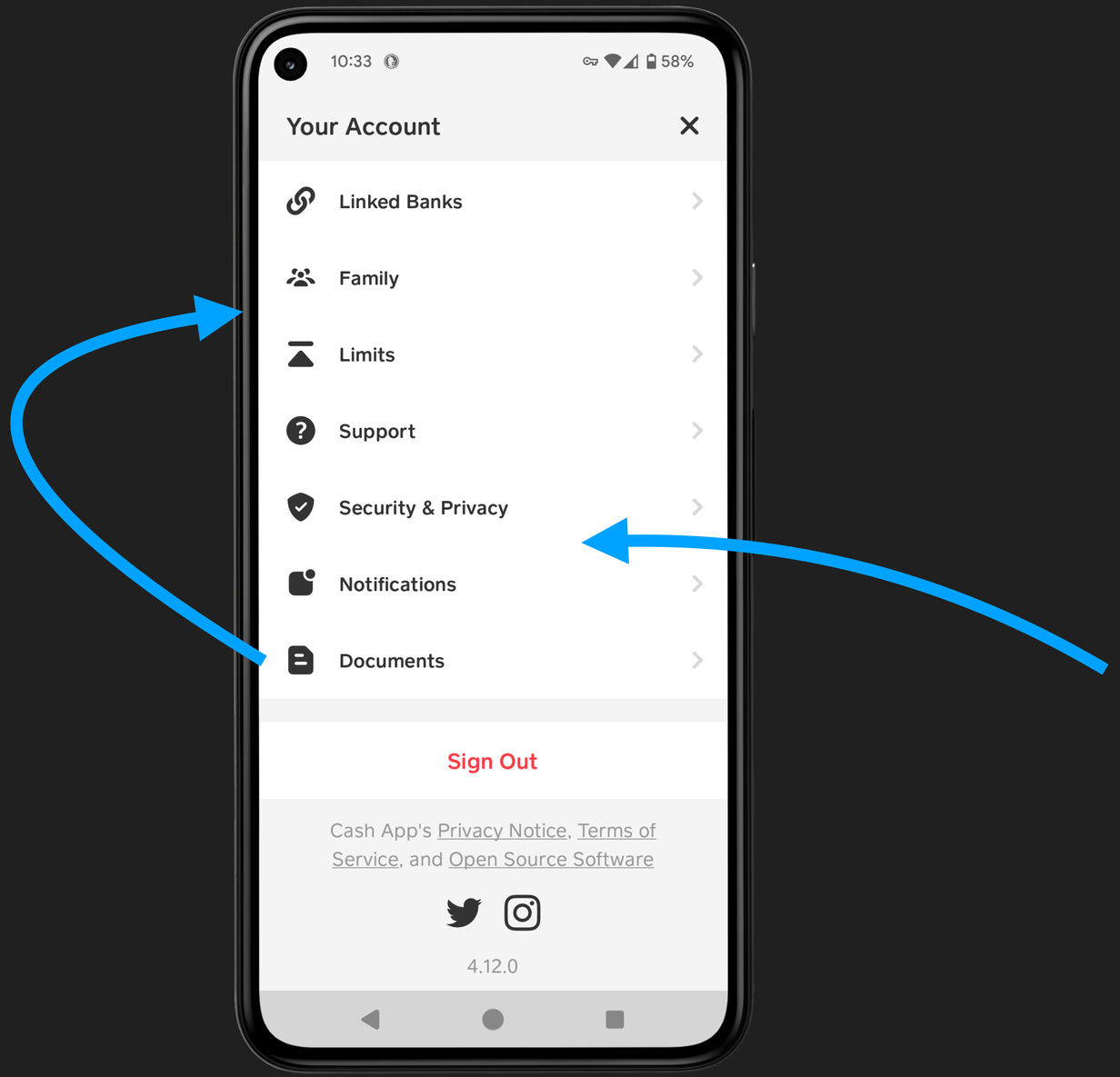
~1 day

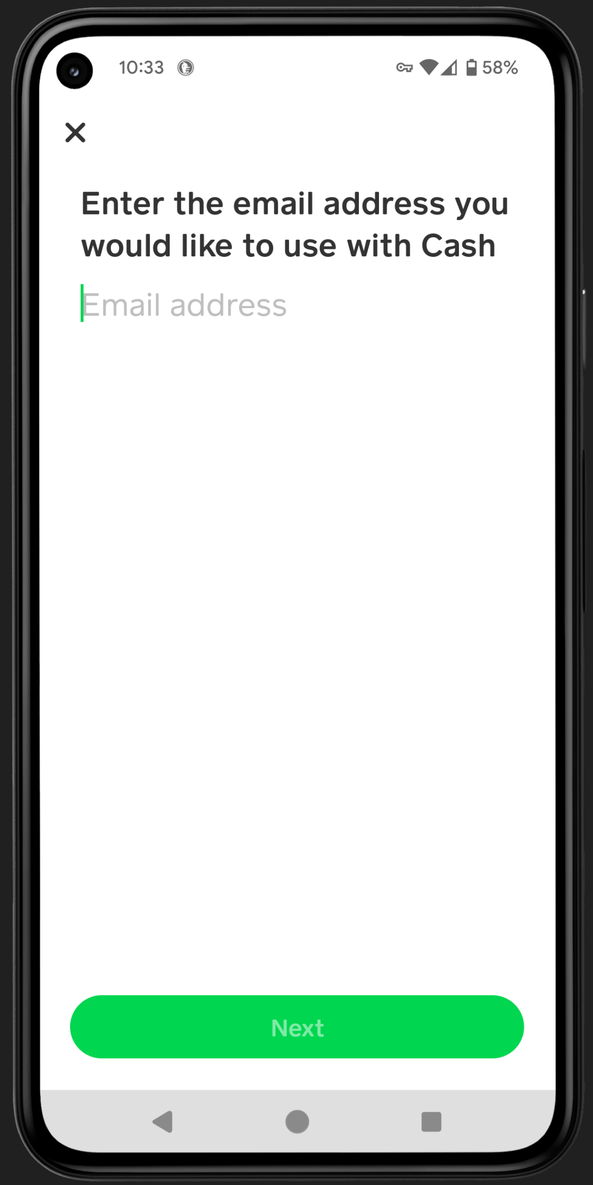
3 - 7 days

3 - 7 days


10:33
58%
Your Account
Linked Banks
Family
Limits
Support
Security & Privacy
Notifications
Documents
Sign Out
Cash App's Privacy Notice, Terms of Service, and Open Source Software
4.12.0


10:33
58%
Your Account
Linked Banks
Family
Limits
Support
Security & Privacy
Notifications
Documents
Sign Out
Cash App's Privacy Notice, Terms of Service, and Open Source Software
4.12.0


10:33
58%
Your Account
Linked Banks
Family
Limits
Support
Security & Privacy
Notifications
Documents
Sign Out
Cash App's Privacy Notice, Terms of Service, and Open Source Software
4.12.0


10:33
58%
Enter the email address you would like to use with Cash
Email address
Next

58%

ail address you
use with Cash

Next

11:04 55%

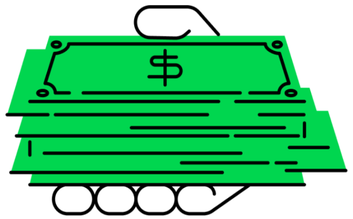
How much do you want to save?

$0

1 2 3
4 5 6
7 8 9
. 0

Continue

11:04 54%



Use your Cash Card to withdraw at ATMs

Withdraw anywhere
Withdraw at any ATM in the world, and pay lower fees at the ATMs found on our map.

Get fees instantly reimbursed when you direct deposit
We'll instantly reimburse all fees for in-network withdrawals and 1 out-of-network withdrawal each month you direct deposit $300+.

Get started

Set up direct deposit

11:01 55%

Please enter your current address

Current Address

Next

11:02

How to make a b
transfer

Transfers in 2-3 busines
external bank may charg
from $0 to $15 per transf

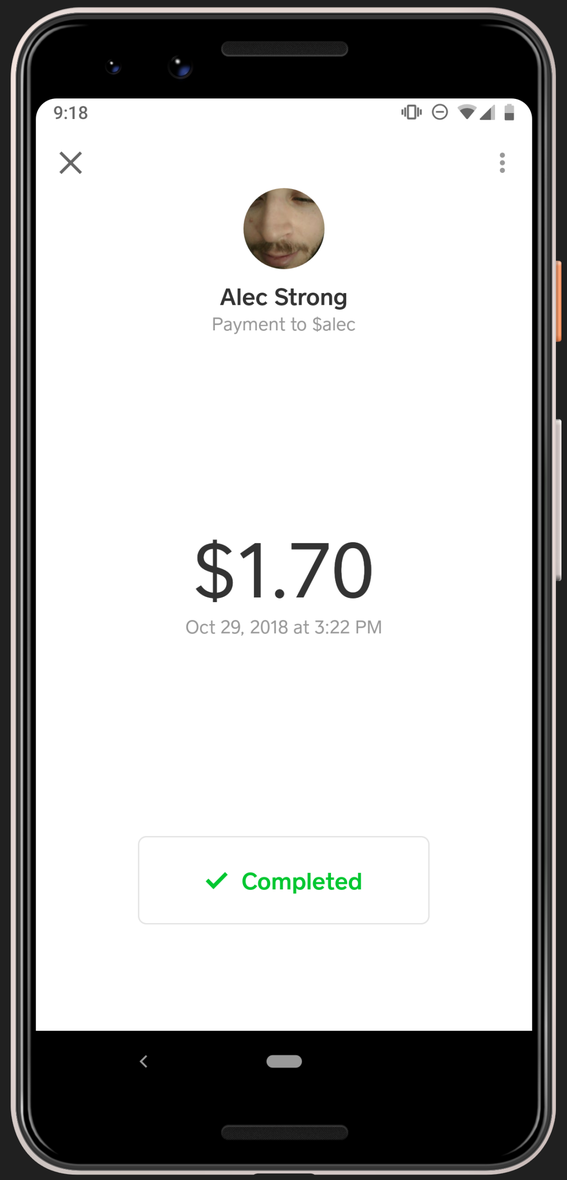
1 Go to the transfe
the bank's websi
Many banks call thi

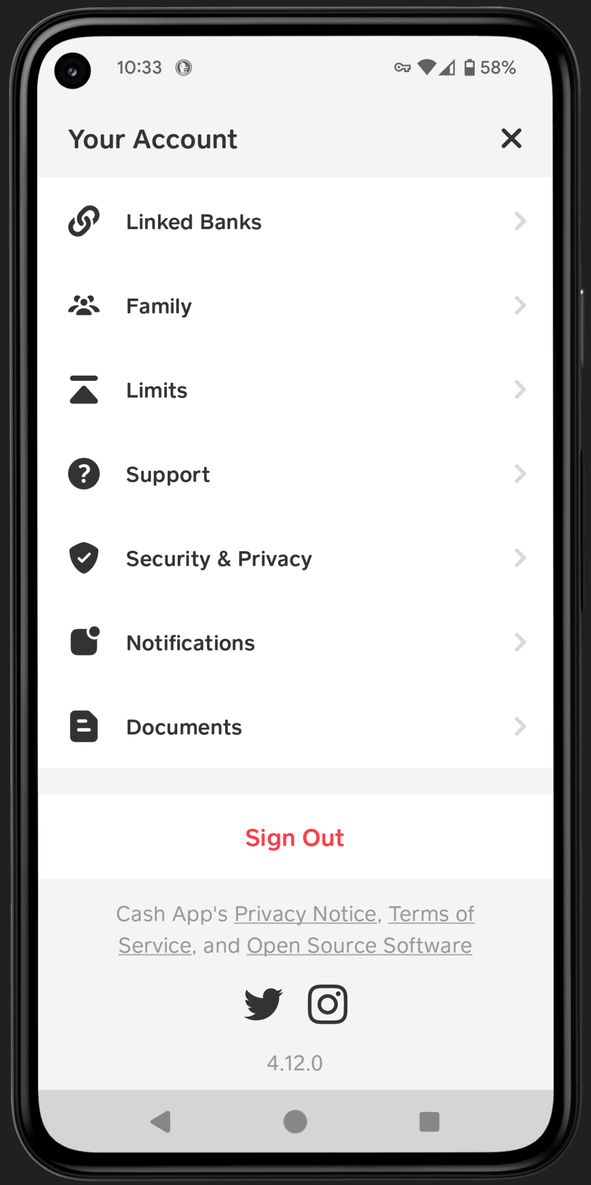
2 Use your routing
numbers to make
If it's your first tran
add Cash App as a
account. Select Che
account type.

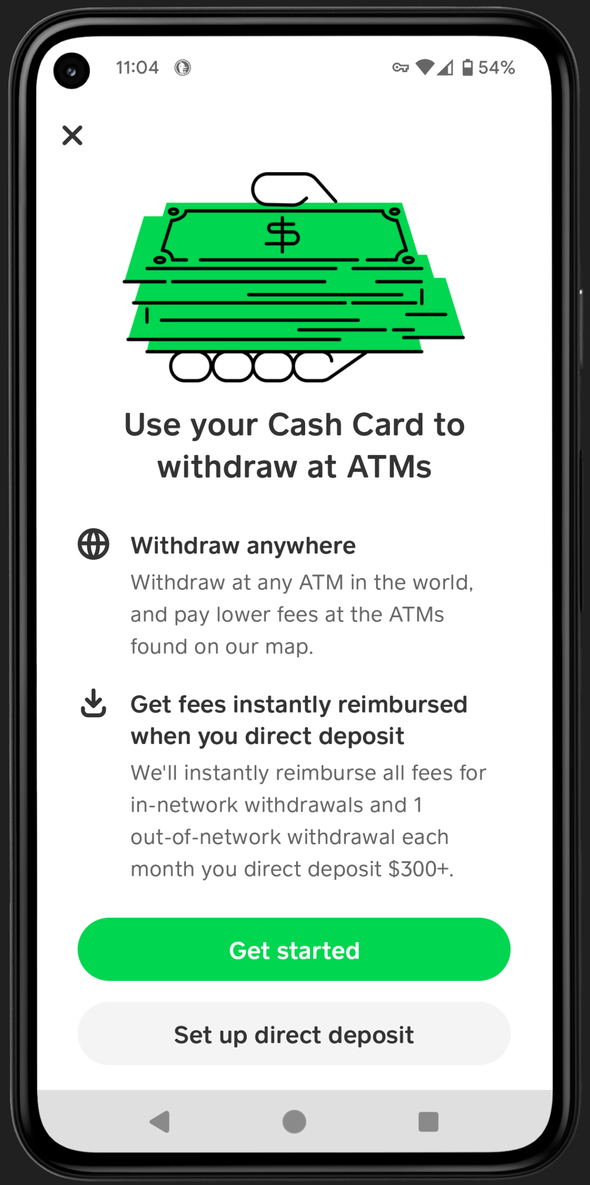
Transfer up to $25,

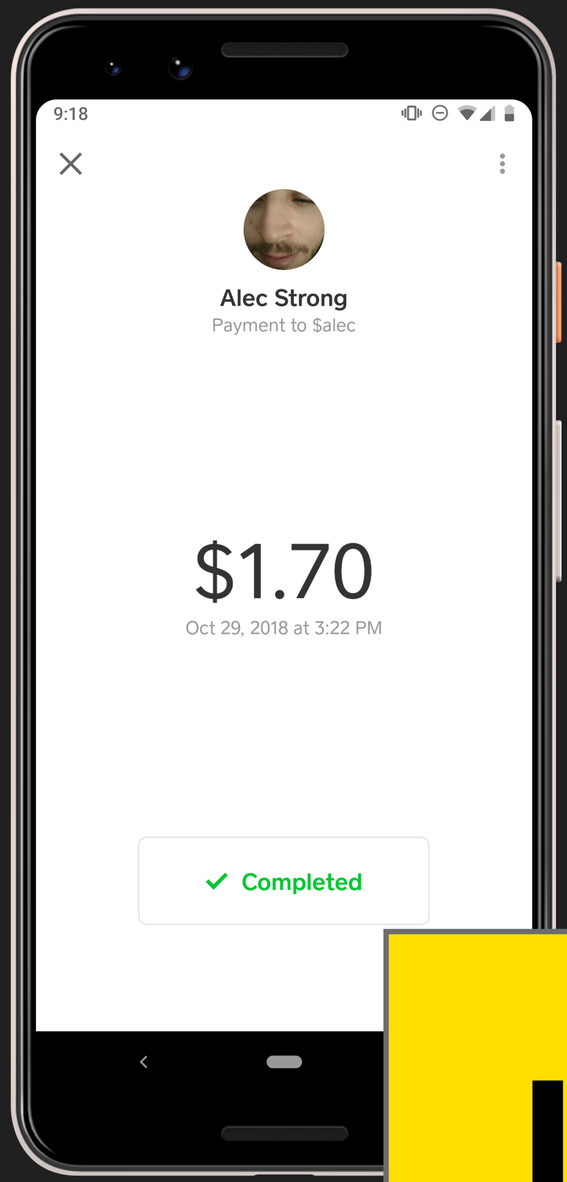
041 215 663

Done

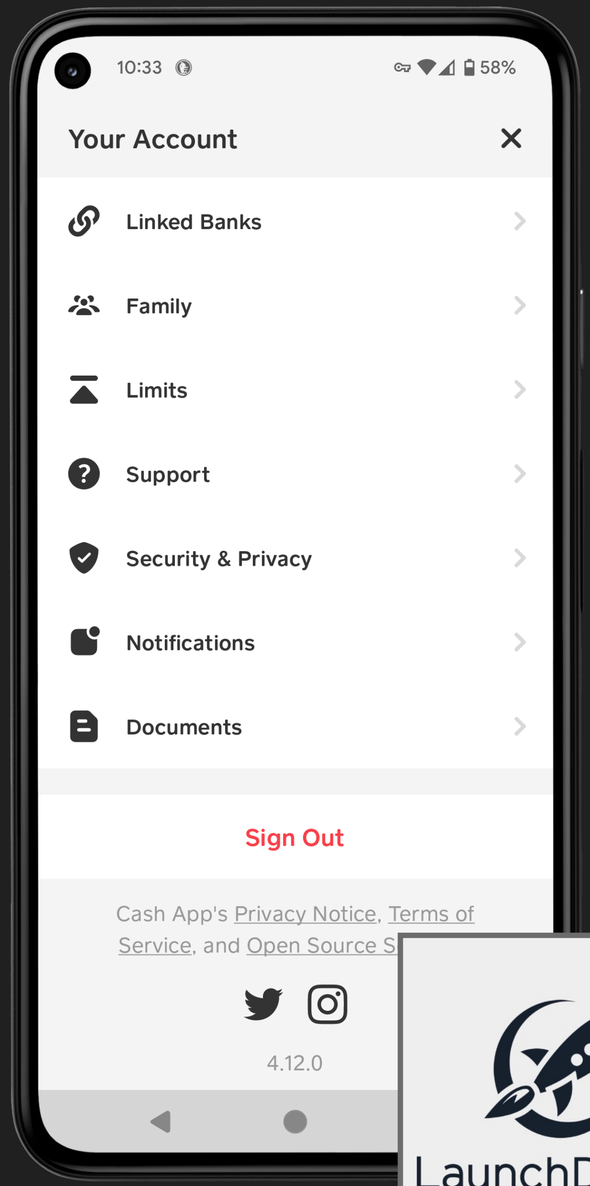

9:18
Alec Strong
Payment to $alec
$1.70
Oct 29, 2018 at 3:22 PM
Completed


10:33
58%
Your Account
Linked Banks
Family
Limits
Support
Security & Privacy
Notifications
Documents
Sign Out
Cash App's Privacy Notice, Terms of Service, and Open Source Software
4.12.0

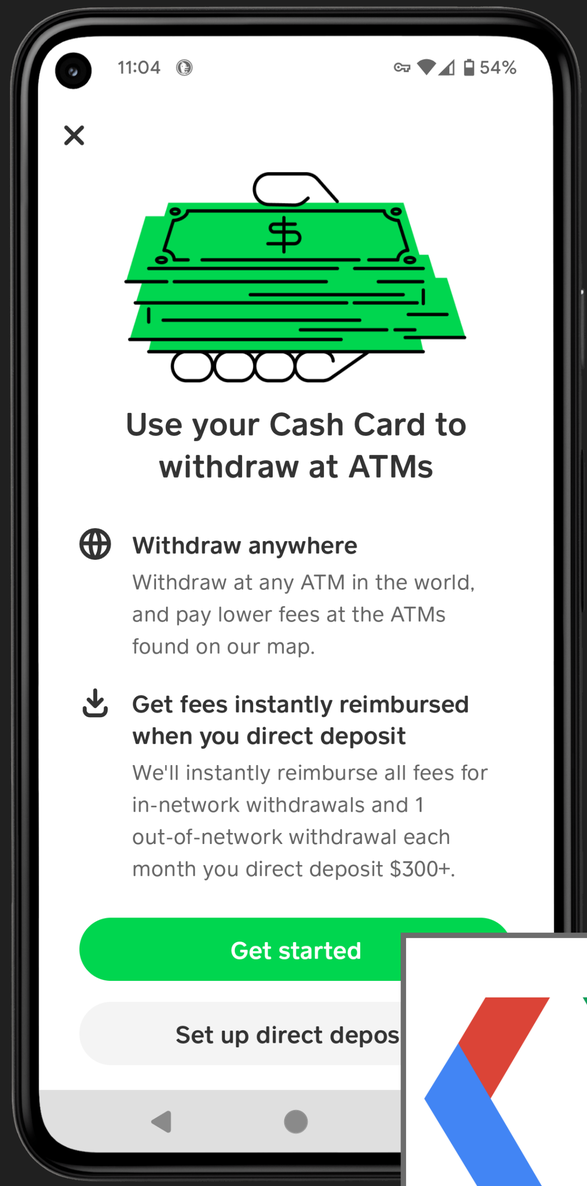

11:04
54%
Use your Cash Card to withdraw at ATMs
Withdraw anywhere
Withdraw at any ATM in the world, and pay lower fees at the ATMs found on our map.
Get fees instantly reimbursed when you direct deposit
We'll instantly reimburse all fees for in-network withdrawals and 1 out-of-network withdrawal each month you direct deposit $300+.
Get started
Set up direct deposit

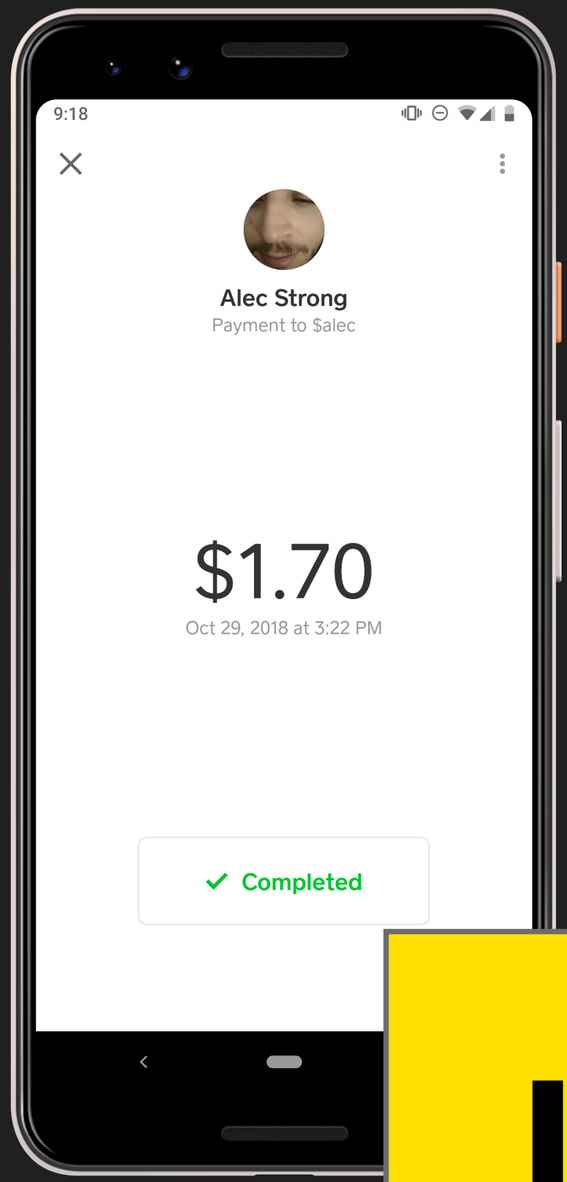

9:18
Alec Strong
Payment to $alec
$1.70
Oct 29, 2018 at 3:22 PM
Completed


JS

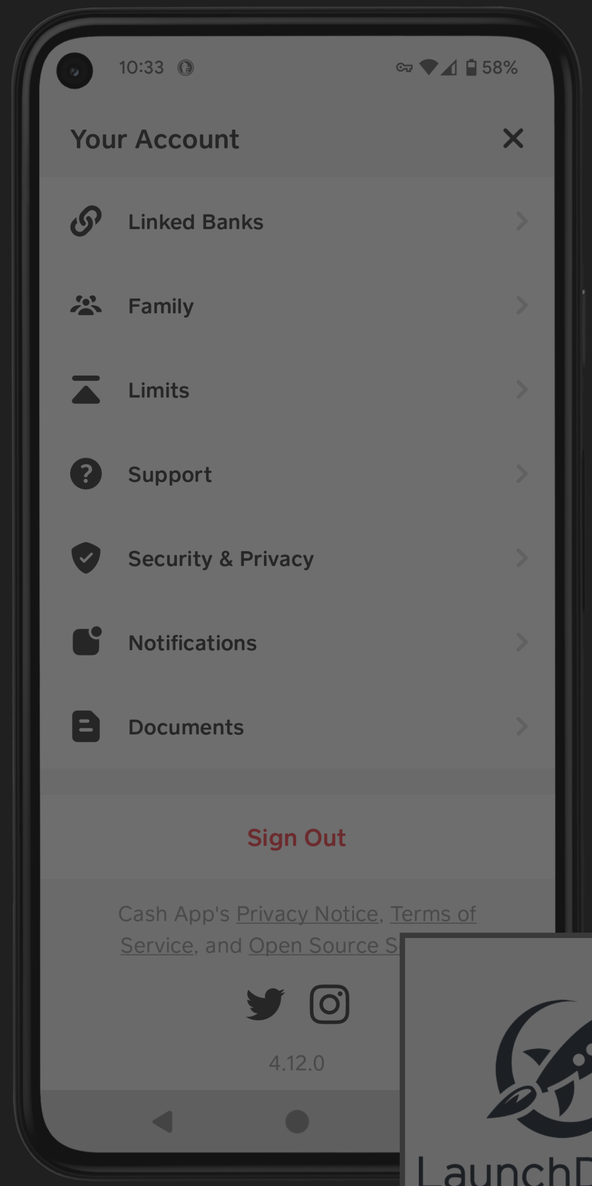

10:33
58%
Your Account
Linked Banks
Family
Limits
Support
Security & Privacy
Notifications
Documents
Sign Out
Cash App's Privacy Notice, Terms of
Service, and Open Source S
4.12.0


LaunchDarkly

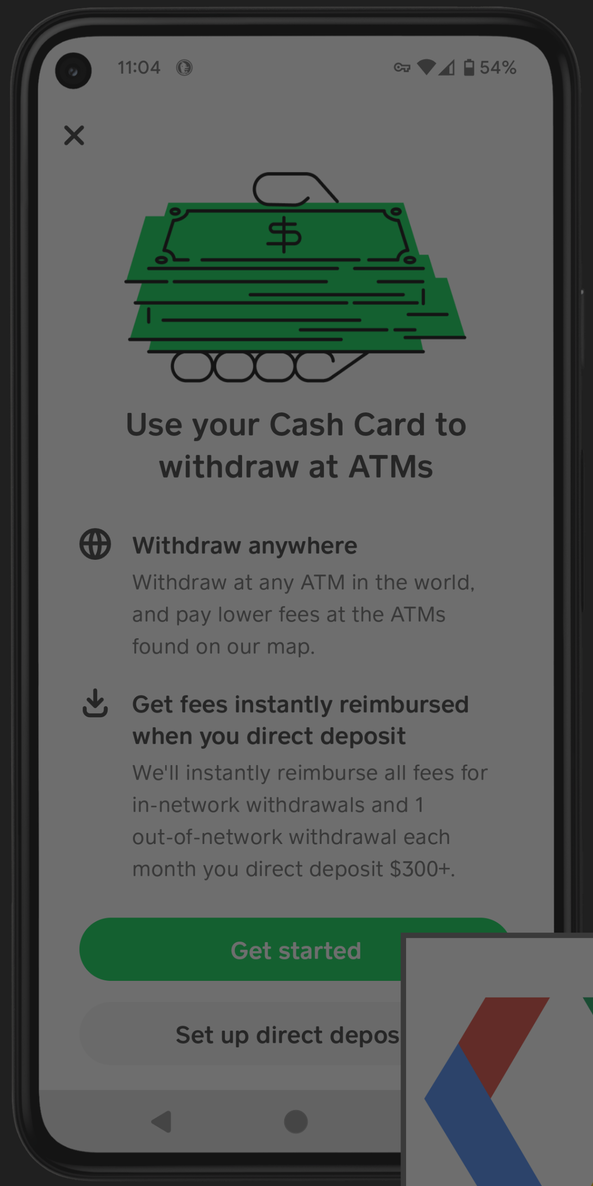

11:04
54%
Use your Cash Card to
withdraw at ATMs
Withdraw anywhere
Withdraw at any ATM in the world,
and pay lower fees at the ATMs
found on our map.
Get fees instantly reimbursed
when you direct deposit
We'll instantly reimburse all fees for
in-network withdrawals and 1
out-of-network withdrawal each
month you direct deposit $300+.
Get started
Set up direct depos

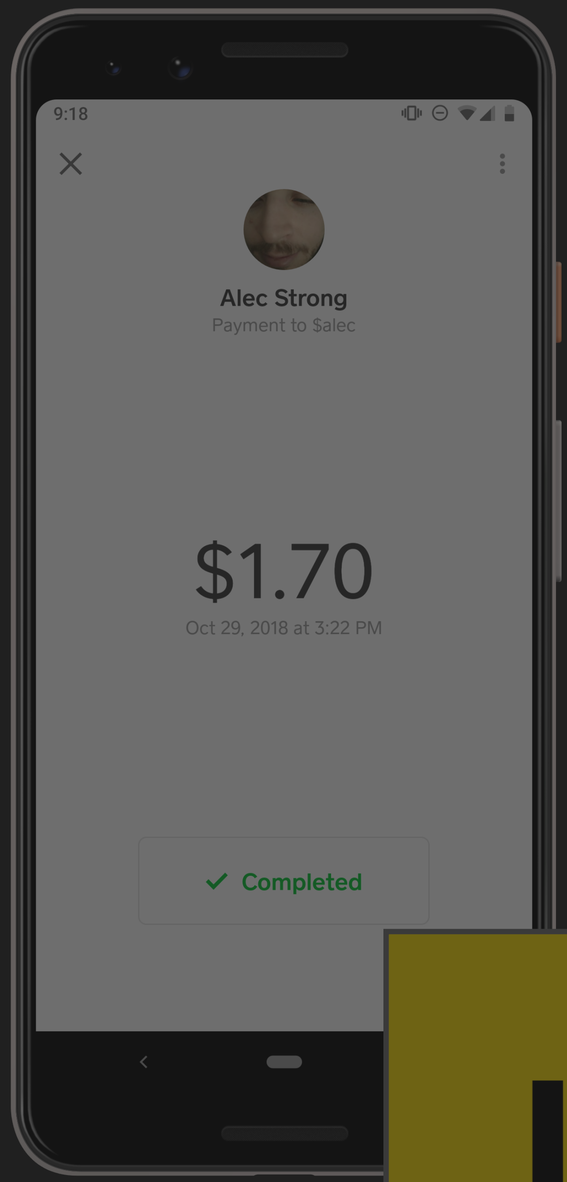

9:18
Alec Strong
Payment to $alec
$1.70
Oct 29, 2018 at 3:22 PM
Completed


JS

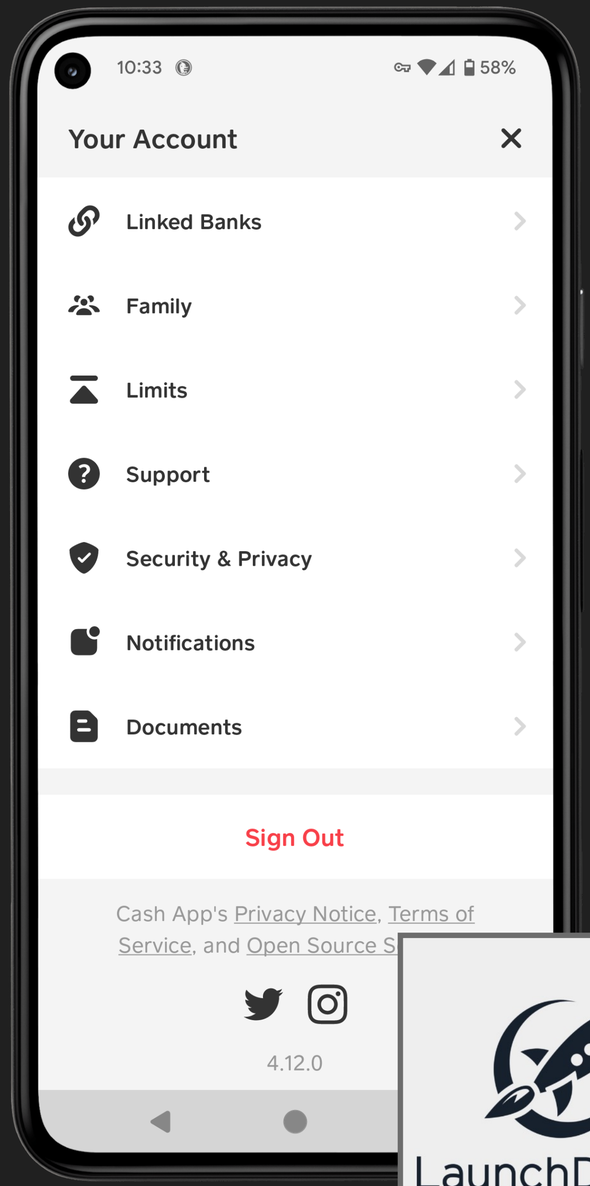

10:33
58%
Your Account
Linked Banks
Family
Limits
Support
Security & Privacy
Notifications
Documents
Sign Out
Cash App's Privacy Notice, Terms of
Service, and Open Source S
4.12.0


LaunchDarkly

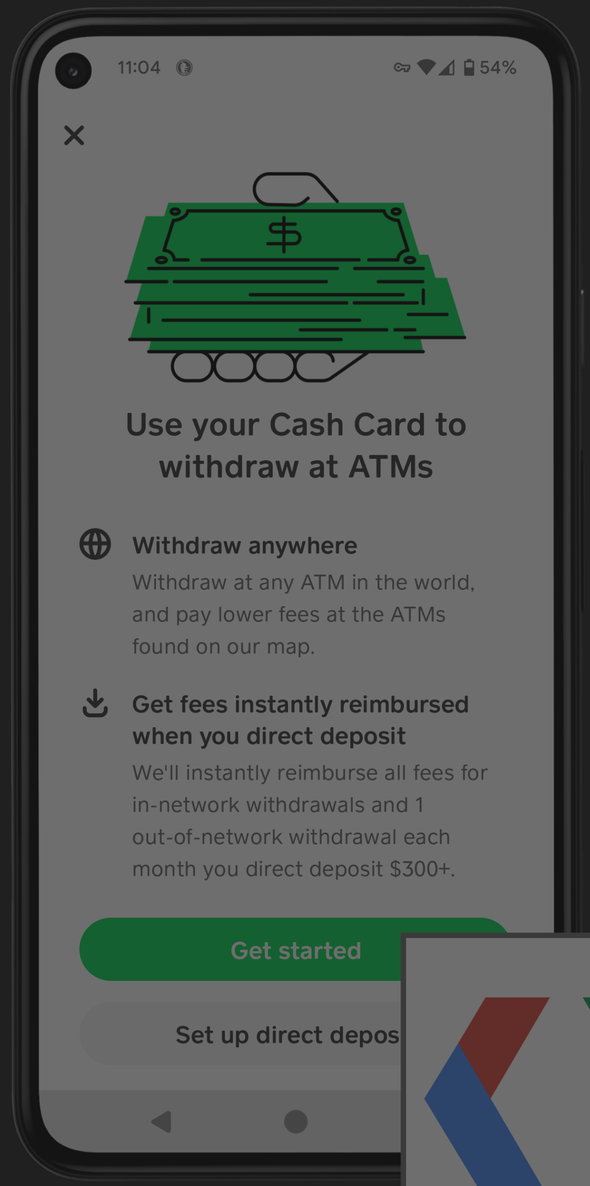

11:04
54%
Use your Cash Card to
withdraw at ATMs
Withdraw anywhere
Withdraw at any ATM in the world,
and pay lower fees at the ATMs
found on our map.
Get fees instantly reimbursed
when you direct deposit
We'll instantly reimburse all fees for
in-network withdrawals and 1
out-of-network withdrawal each
month you direct deposit $300+.
Get started
Set up direct depos

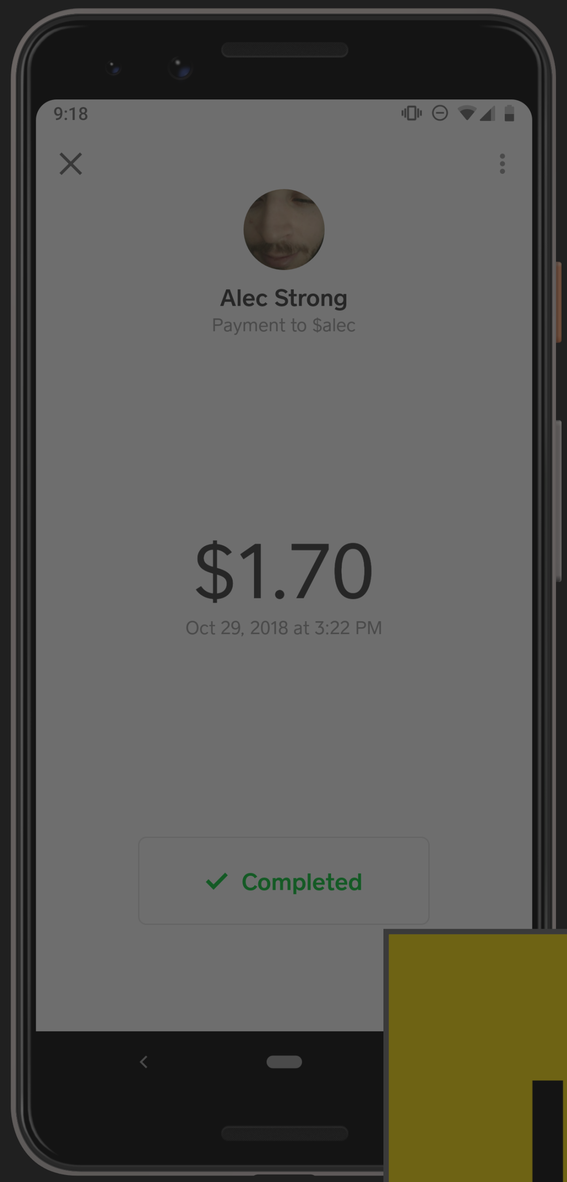

9:18
Alec Strong
Payment to $alec
$1.70
Oct 29, 2018 at 3:22 PM
Completed

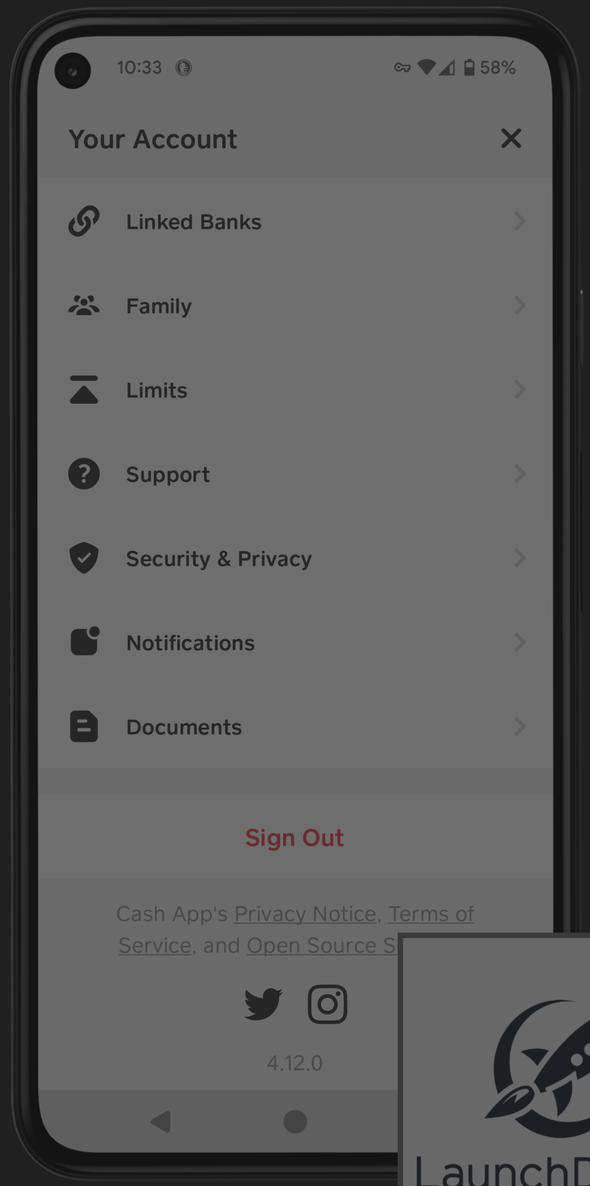

JS

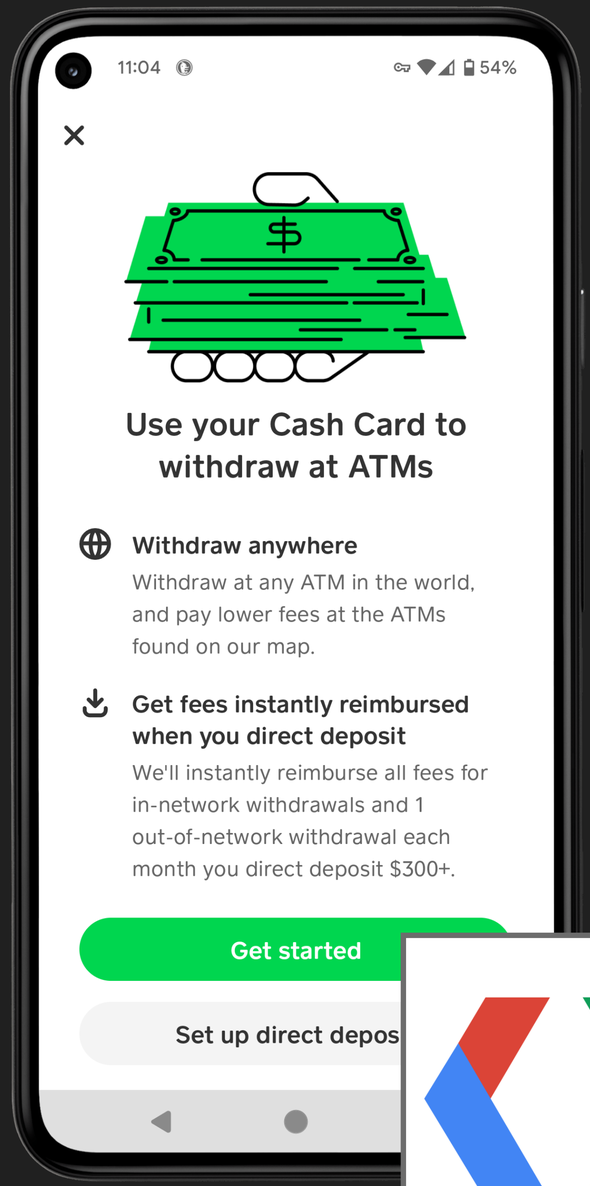

10:33
58%
Your Account
Linked Banks
Family
Limits
Support
Security & Privacy
Notifications
Documents
Sign Out
Cash App's Privacy Notice, Terms of
Service, and Open Source S
4.12.0


LaunchDarkly


11:04
54%
Use your Cash Card to
withdraw at ATMs
Withdraw anywhere
Withdraw at any ATM in the world,
and pay lower fees at the ATMs
found on our map.
Get fees instantly reimbursed
when you direct deposit
We'll instantly reimburse all fees for
in-network withdrawals and 1
out-of-network withdrawal each
month you direct deposit $300+.
Get started
Set up direct depos

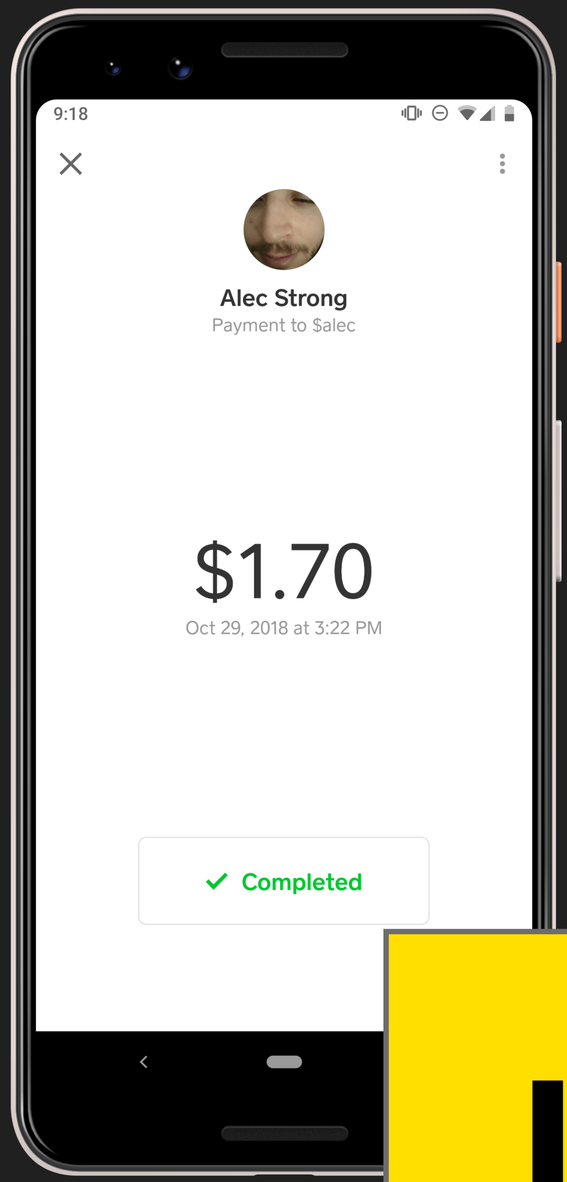

9:18
Alec Strong
Payment to $alec
$1.70
Oct 29, 2018 at 3:22 PM
Completed


10:33
58%
Your Account
Linked Banks
Family
Limits
Support
Security & Privacy
Notifications
Documents
Sign Out
Cash App's Privacy Notice, Terms of
Service, and Open Source S
4.12.0

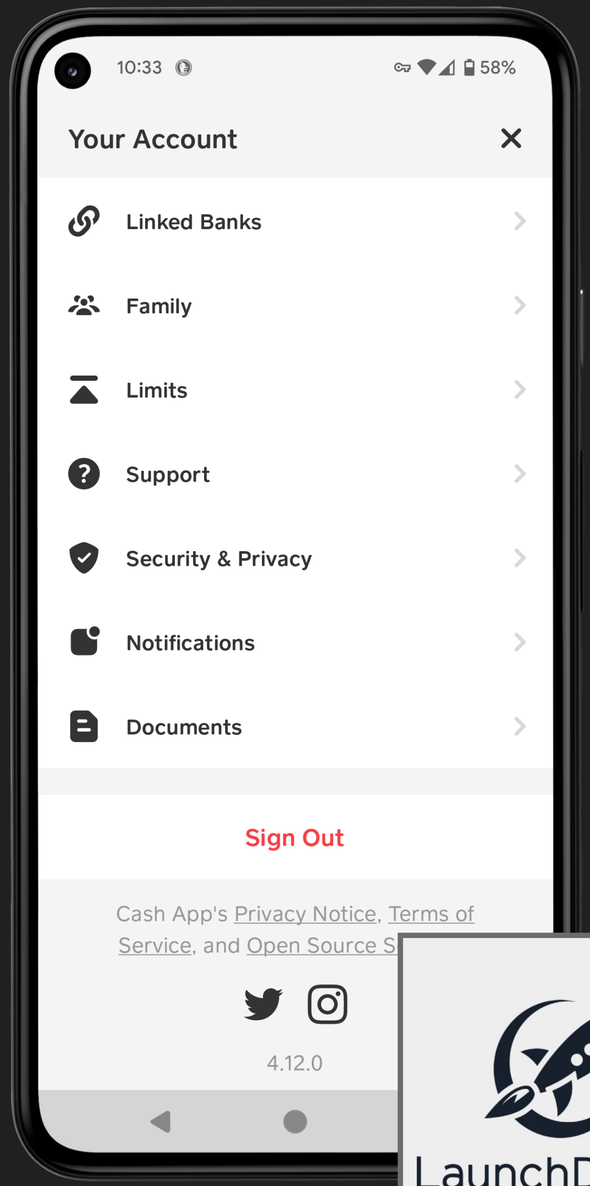

11:04
54%
Use your Cash Card to withdraw at ATMs
Withdraw anywhere
Withdraw at any ATM in the world, and pay lower fees at the ATMs found on our map.
Get fees instantly reimbursed when you direct deposit
We'll instantly reimburse all fees for in-network withdrawals and 1 out-of-network withdrawal each month you direct deposit $300+.
Get started
Set up direct depos


JS

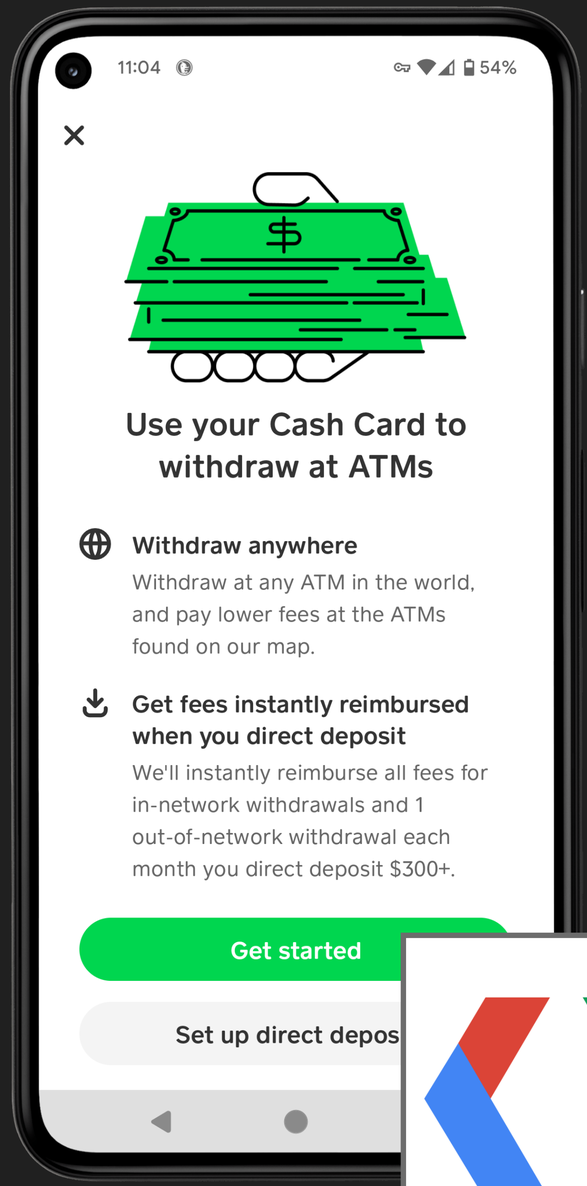

LaunchDarkly


9:18
Alec Strong
Payment to $alec
$1.70
Oct 29, 2018 at 3:22 PM
Completed


JS


10:33
58%
Your Account
Linked Banks
Family
Limits
Support
Security & Privacy
Notifications
Documents
Sign Out
Cash App's Privacy Notice, Terms of
Service, and Open Source S
4.12.0


LaunchDarkly


11:04
54%
Use your Cash Card to
withdraw at ATMs
Withdraw anywhere
Withdraw at any ATM in the world,
and pay lower fees at the ATMs
found on our map.
Get fees instantly reimbursed
when you direct deposit
We'll instantly reimburse all fees for
in-network withdrawals and 1
out-of-network withdrawal each
month you direct deposit $300+.
Get started
Set up direct depos

Unification Goals

Unification Goals

- Logic to be updated outside of app store releases

Unification Goals

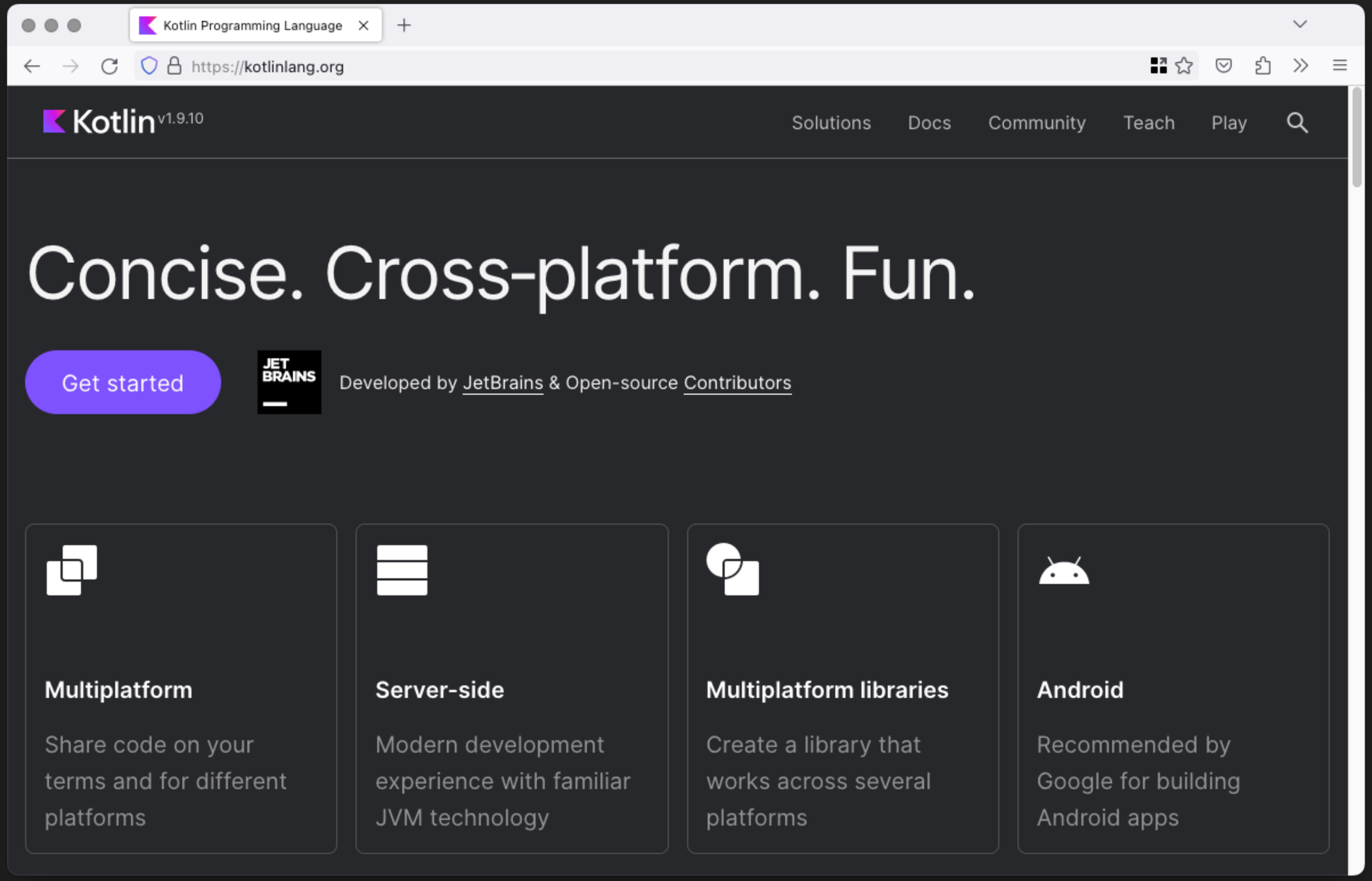
- Logic to be updated outside of app store releases
- Render screens using existing native UI elements

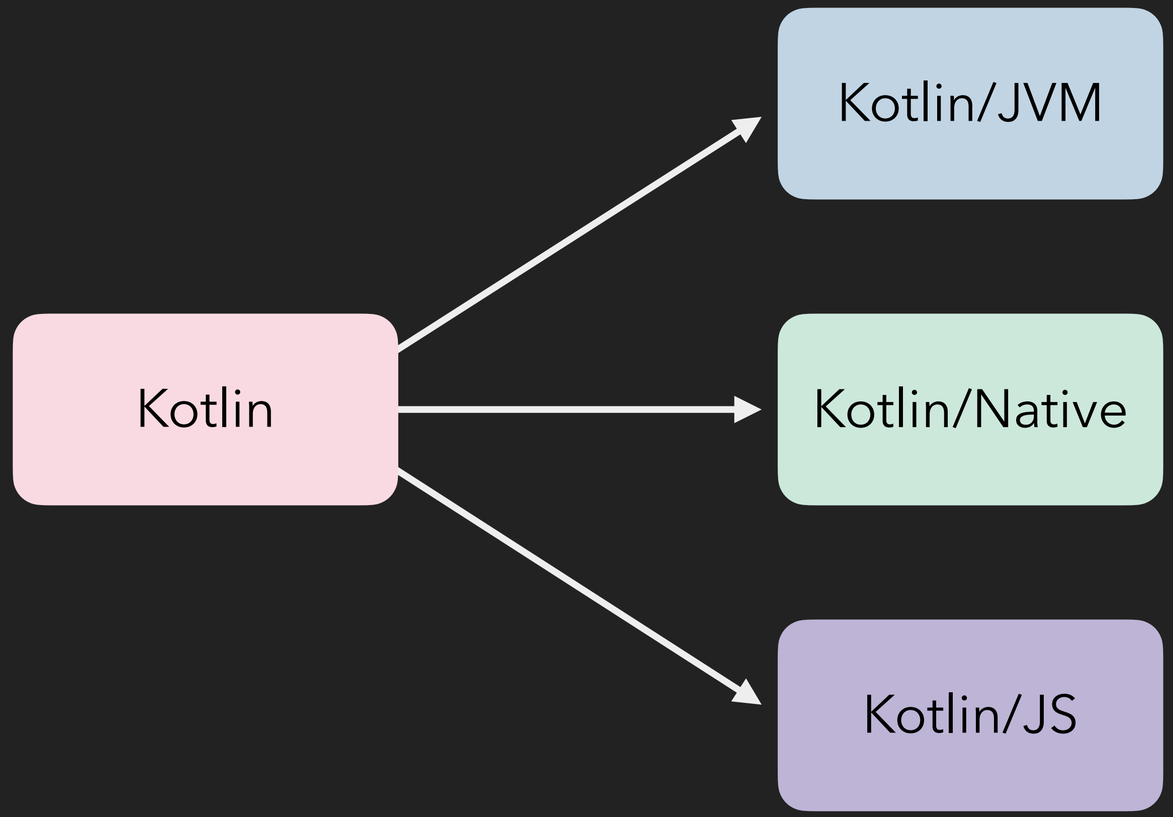
Unification Goals

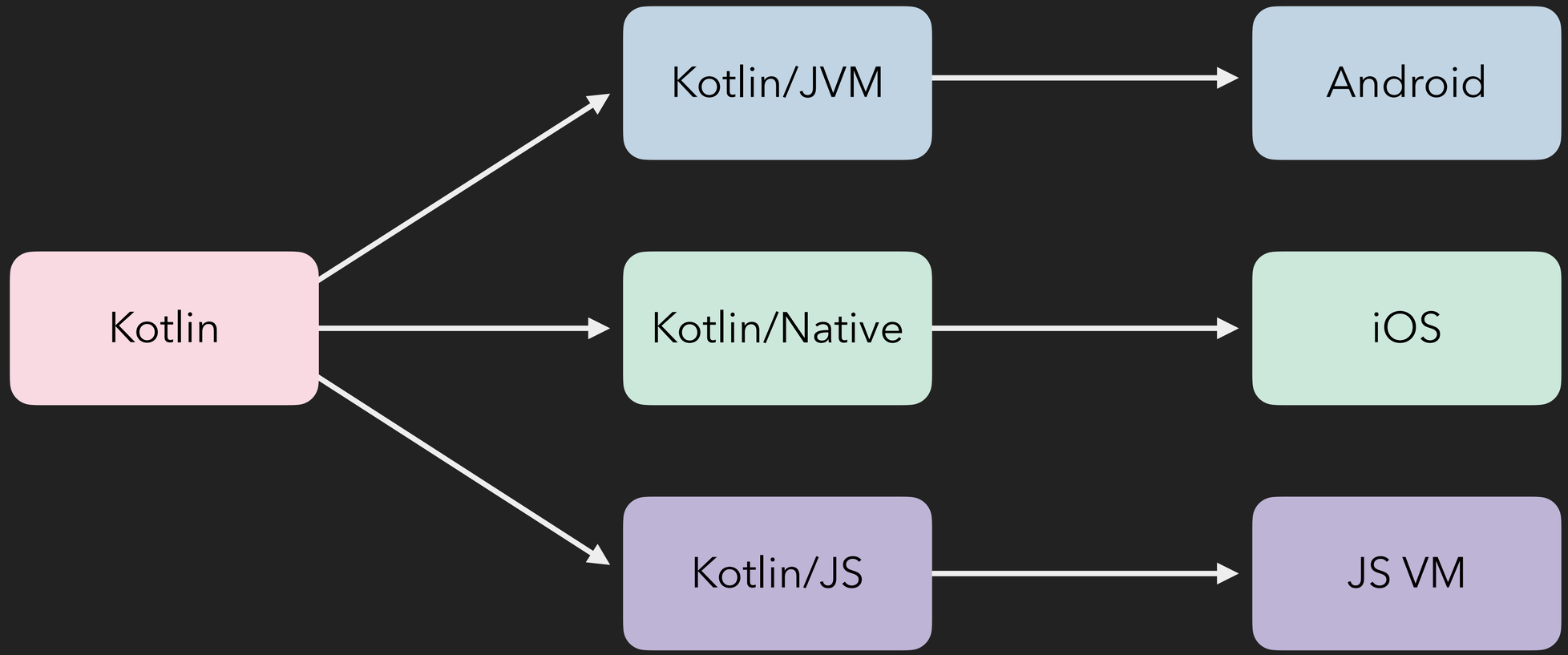
- Logic to be updated outside of app store releases
- Render screens using existing native UI elements
- Enable creation of new screens without prior knowledge

Unification Goals

- Logic to be updated outside of app store releases
- Render screens using existing native UI elements
- Enable creation of new screens without prior knowledge
- Not be a regression on native screen development


Kotlin Programming Language
https://kotlinlang.org
Kotlin v1.9.10
Solutions
Docs
Community
Teach
Play
Concise. Cross-platform. Fun.
Get started
JET BRAINS
Developed by JetBrains & Open-source Contributors
Multiplatform
Share code on your terms and for different platforms
Server-side
Modern development experience with familiar JVM technology
Multiplatform libraries
Create a library that works across several platforms
Android
Recommended by Google for building Android apps


Kotlin
Kotlin/JVM
Kotlin/Native
Kotlin/JS


Kotlin
Kotlin/JVM
Kotlin/Native
Kotlin/JS
Android
iOS
JS VM

Zipline


JS


JS
WA

```
class PaymentRenderer {
  fun render(payment: Payment): String {
    …
  }
}
```

```
class RealPaymentRenderer : PaymentRenderer {
  override fun render(payment: Payment): String {
    …
  }
}

interface PaymentRenderer : ZiplineService {
  fun render(payment: Payment): String
}
```

```
class RealPaymentRenderer : PaymentRenderer {
  override fun render(payment: Payment): String {
    …
  }
}
```

```
class RealPaymentRenderer : PaymentRenderer {
  override fun render(payment: Payment): String {
    …
  }
}

zipline.bind<PaymentRenderer>(RealPaymentRenderer())
```

```
class RealPaymentRenderer : PaymentRenderer {
  override fun render(payment: Payment): String {
    …
  }
}

zipline.bind<PaymentRenderer>(RealPaymentRenderer())
```

JS

```
val renderer = zipline.take<PaymentRenderer>()

println(renderer.render(Payment(…)))
```

```
class RealPaymentRenderer : PaymentRenderer {
  override fun render(payment: Payment): String {
    /* Fancy new impl */
  }
}

zipline.bind<PaymentRenderer>(RealPaymentRenderer())
```

JS

```
val renderer = zipline.take<PaymentRenderer>()

println(renderer.render(Payment(…)))
```

```
class PaymentPresenter : Presenter {
  override fun render() {
    /* Display UI on Android + iOS somehow… */
  }
}
```

```
class PaymentPresenter : Presenter {
  override fun render() {
    /* Display UI on Android + iOS somehow… */
  }
}
```

???

Redwood

Design System Schema

```
data class Column(
  val children: () -> Unit,
)

data class TextInput(
  val hint: String,
  val text: String,
  val onTextChanged: (String) -> Unit,
)
```

Design System Schema

```
data class Column(
  val children: () -> Unit,
)

data class TextInput(
  val hint: String,
  val text: String,
  val onTextChanged: (String) -> Unit,
)
```

Design System Schema

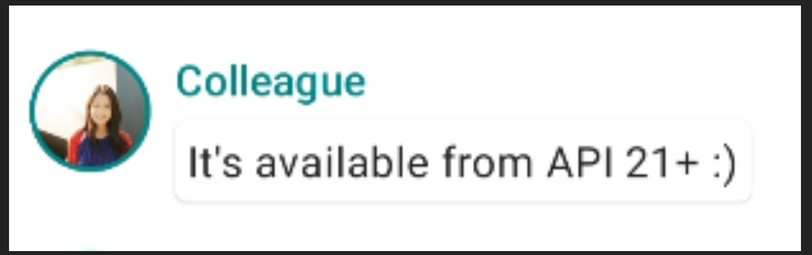
```
data class Column(
  val children: () -> Unit,
)

data class TextInput(
  val hint: String,
  val text: String,
  val onTextChanged: (String) -> Unit,
)
```

Design System Schema

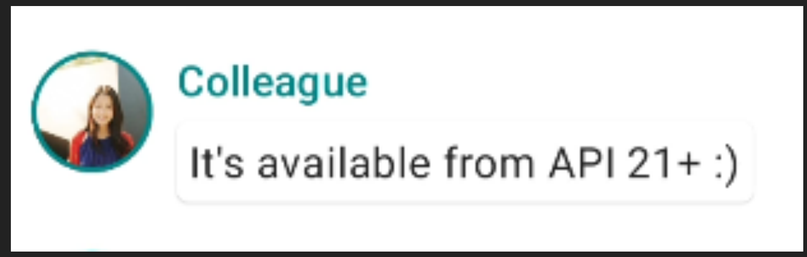
```
data class Column(
  val children: () -> Unit,
)

data class TextInput(
  val hint: String,
  val text: String,
  val onTextChanged: (String) -> Unit,
)
```

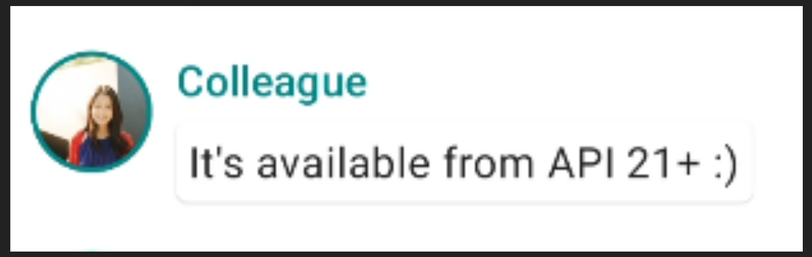
Design System Schema

```
data class Column(
  val children: () -> Unit,
)

data class TextInput(
  val hint: String,
  val text: String,
  val onTextChanged: (String) -> Unit,
)
```


Colleague
It's available from API 21+ :)

```
data class Row(…)
data class Column(…)

data class Image(
  val url: HttpUrl,
  val size: ImageSize,
  val borderStyle: BorderStyle,
)

data class Text(
  val text: String,
  val font: FontFamily,
  val style: FontStyle,
)
```






```
data class ContactItem(
  val name: String,
  val image: HttpUrl,
  val content: () -> Unit,
)

data class Text(
  val text: String,
  val font: FontFamily,
  val style: FontStyle,
)
```

```
data class Row(…)
data class Column(…)

data class Image(
  val url: HttpUrl,
  val size: ImageSize,
  val borderStyle: BorderStyle,
)

data class Text(
  val text: String,
  val font: FontFamily,
  val style: FontStyle,
)
```

```
data class ContactItem(
  val name: String,
  val image: HttpUrl,
  val content: () -> Unit,
)

data class Text(
  val text: String,
  val font: FontFamily,
  val style: FontStyle,
)
```

Compose

```
data class Column(
  val children: () -> Unit,
)

data class TextInput(
  val hint: String,
  val text: String,
  val onTextChanged: (String) -> Unit,
)
```

Compose

```
data class Column(
  val children: () -> Unit,
)

data class TextInput(
  val hint: String,
  val text: String,
  val onTextChanged: (String) -> Unit,
)
```

```
@Composable fun Column(
  children: @Composable () -> Unit,
) { … }

@Composable fun TextInput(
  hint: String,
  text: String,
  onTextChanged: (String) -> Unit,
) { … }
```

Compose

```
@Composable fun Column(
  children: @Composable () -> Unit,
) { … }

@Composable fun TextInput(
  hint: String,
  text: String,
  onTextChanged: (String) -> Unit,
) { … }
```

Compose

```
Column {
  var query by remember { mutableStateOf("") }
  TextInput(
    hint = "Search",
    text = query,
    onTextChanged = { query = it },
  )
  val images = LoadImages(query)
  ScrollableColumn {
    for (image in images) {
      Image(url = image.url)
    }
  }
}
```

Widget Bindings

```
data class Column(
  val children: () -> Unit,
)

data class TextInput(
  val hint: String,
  val text: String,
  val onTextChanged: (String) -> Unit,
)
```

Widget Bindings

```
data class Column(
  val children: () -> Unit,
)

data class TextInput(
  val hint: String,
  val text: String,
  val onTextChanged: (String) -> Unit,
)
```

```
interface Column<T : Any> : Widget<T> {
  val children: Widget.Children<T>
}

interface TextInput<T : Any> : Widget<T> {
  fun hint(hint: String)
  fun text(text: String)
  fun onTextChanged(onTextChanged: ((String) -> Unit)?)
}
```

Widget Bindings

```
interface Column<T : Any> : Widget<T> {
  val children: Widget.Children<T>
}

interface TextInput<T : Any> : Widget<T> {
  fun hint(hint: String)
  fun text(text: String)
  fun onTextChanged(onTextChanged: ((String) -> Unit)?)
}
```

Widget Bindings

```
interface Widget<T : Any> {
  val value: T
}

interface Column<T : Any> : Widget<T> {
  val children: Widget.Children<T>
}

interface TextInput<T : Any> : Widget<T> {
  fun hint(hint: String)
  fun text(text: String)
  fun onTextChanged(onTextChanged: ((String) -> Unit)?)
}
```

Widget Bindings

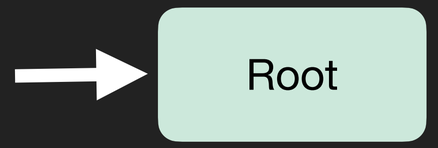
```
interface Column<T : Any> : Widget<T> {
  val children: Widget.Children<T>
}
```

Widget Bindings

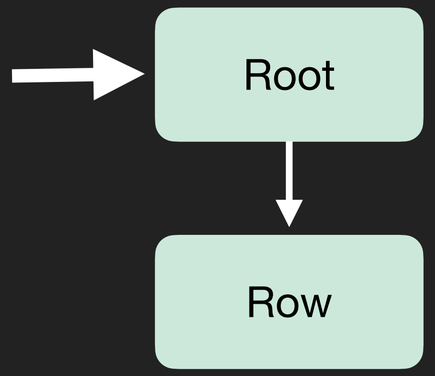
```
class ViewColumn(
  override val value: LinearLayout,
) : Column<View> {
  override val children = ViewGroupChildren(value)
}

interface Column<T : Any> : Widget<T> {
  val children: Widget.Children<T>
}
```

```
@Composable
fun MessageCard(…) {
  Row(…) {
    Image(…)
    Spacer(…)

    Column {
      Text(…)

      Spacer(…)

      Surface(…) {
        Text(…)
      }
    }
  }
}
```

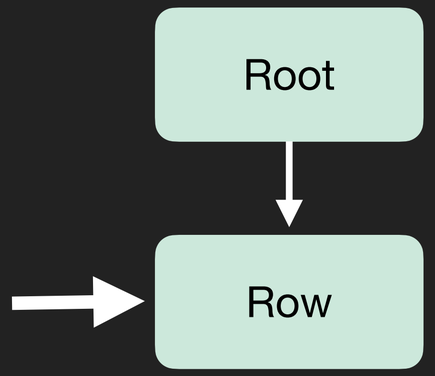
```
@Composable
fun MessageCard(…) {
  Row(…) {
    Image(…)
    Spacer(…)

    Column {
      Text(…)

      Spacer(…)

      Surface(…) {
        Text(…)
      }
    }
  }
}
```



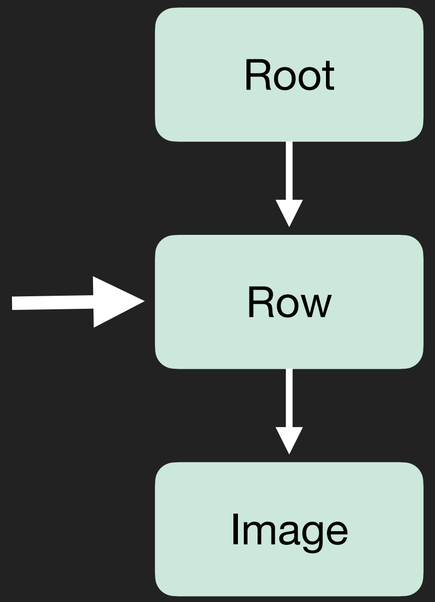
```
@Composable
fun MessageCard(…) {
  Row(…) {
    Image(…)
    Spacer(…)

    Column {
      Text(…)

      Spacer(…)

      Surface(…) {
        Text(…)
      }
    }
  }
}
```



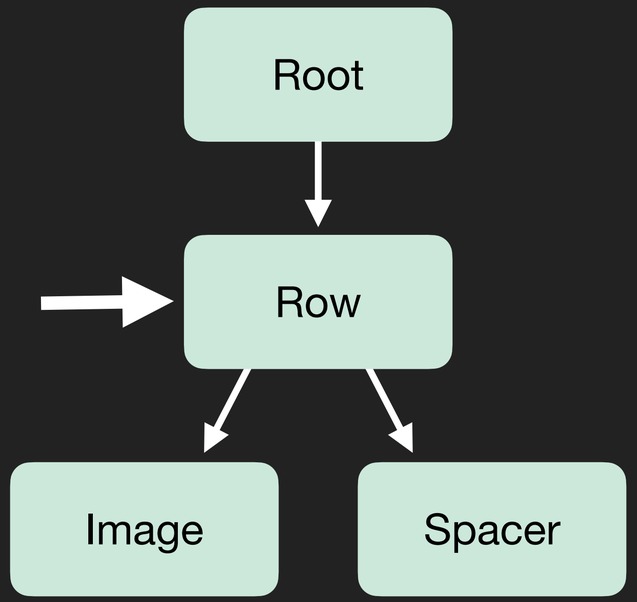
```
@Composable
fun MessageCard(…) {
  Row(…) {
    Image(…)
    Spacer(…)

    Column {
      Text(…)

      Spacer(…)

      Surface(…) {
        Text(…)
      }
    }
  }
}
```



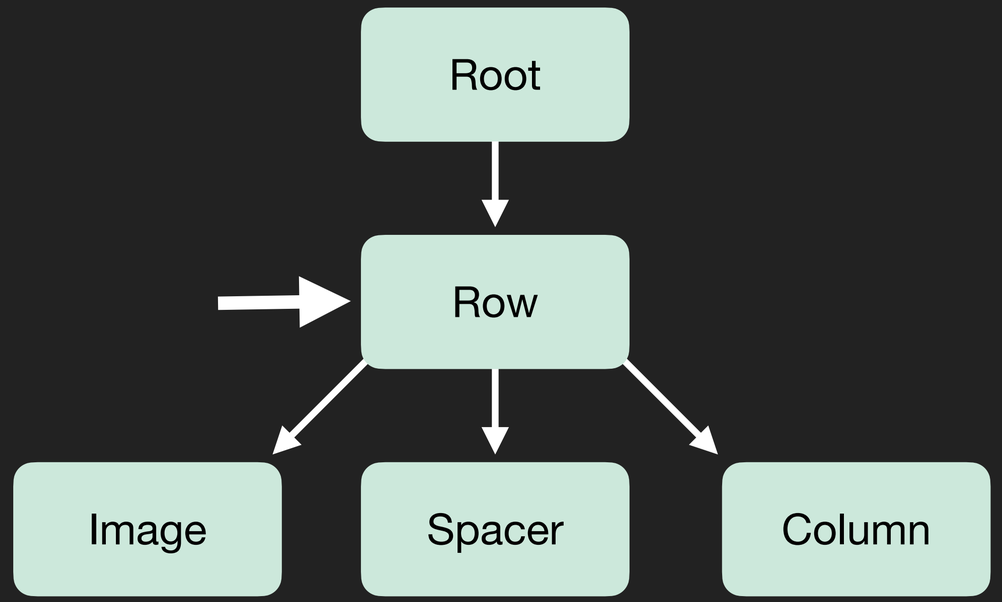
```
@Composable
fun MessageCard(…) {
  Row(…) {
    Image(…)
    Spacer(…)

    Column {
      Text(…)

      Spacer(…)

      Surface(…) {
        Text(…)
      }
    }
  }
}
```



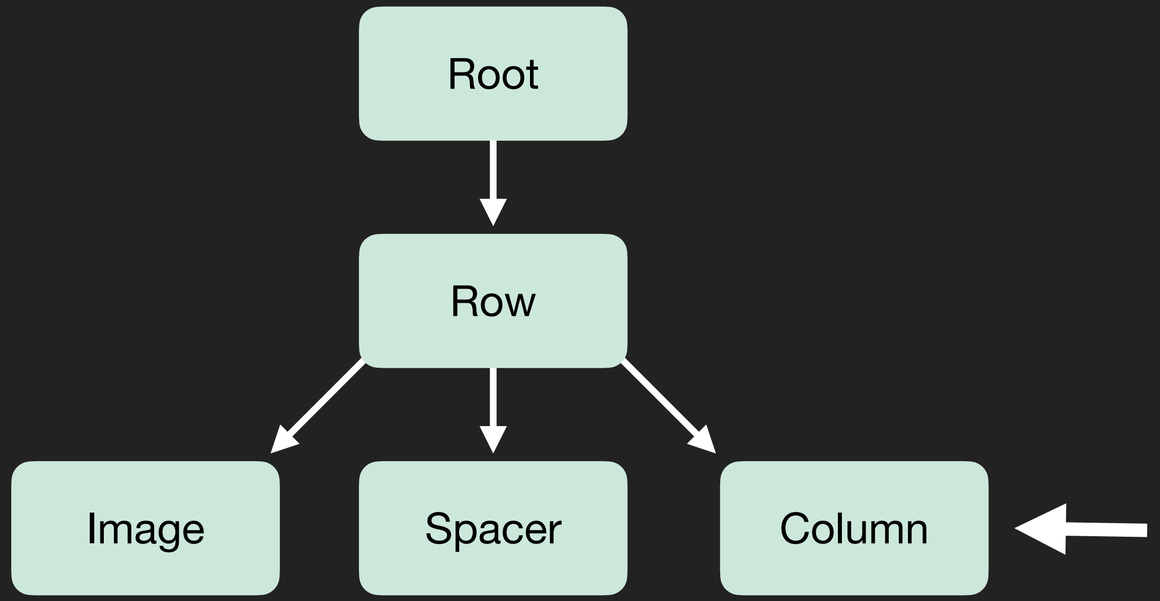
```
@Composable
fun MessageCard(…) {
  Row(…) {
    Image(…)
    Spacer(…)

    Column {
      Text(…)

      Spacer(…)

      Surface(…) {
        Text(…)
      }
    }
  }
}
```



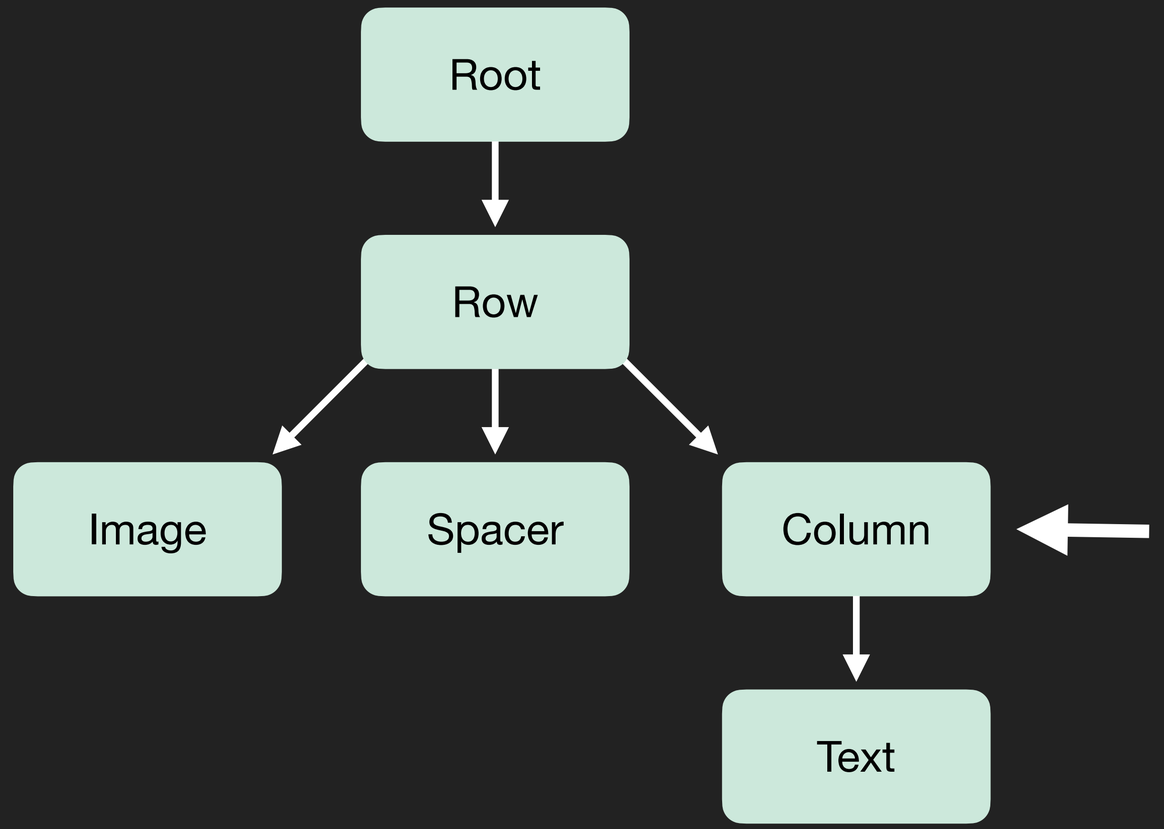
```
@Composable
fun MessageCard(…) {
  Row(…) {
    Image(…)
    Spacer(…)

    Column {
      Text(…)

      Spacer(…)

      Surface(…) {
        Text(…)
      }
    }
  }
}
```



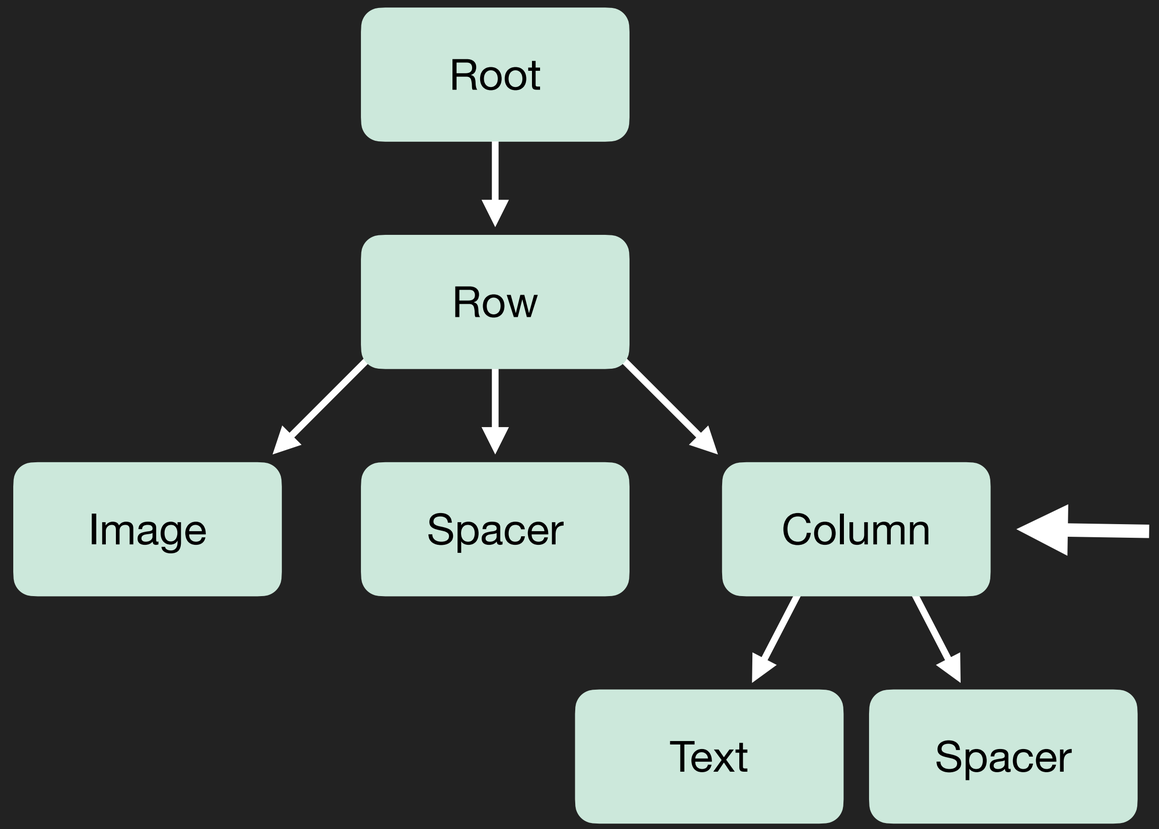
```
@Composable
fun MessageCard(…) {
  Row(…) {
    Image(…)
    Spacer(…)

    Column {
      Text(…)

      Spacer(…)

      Surface(…) {
        Text(…)
      }
    }
  }
}
```



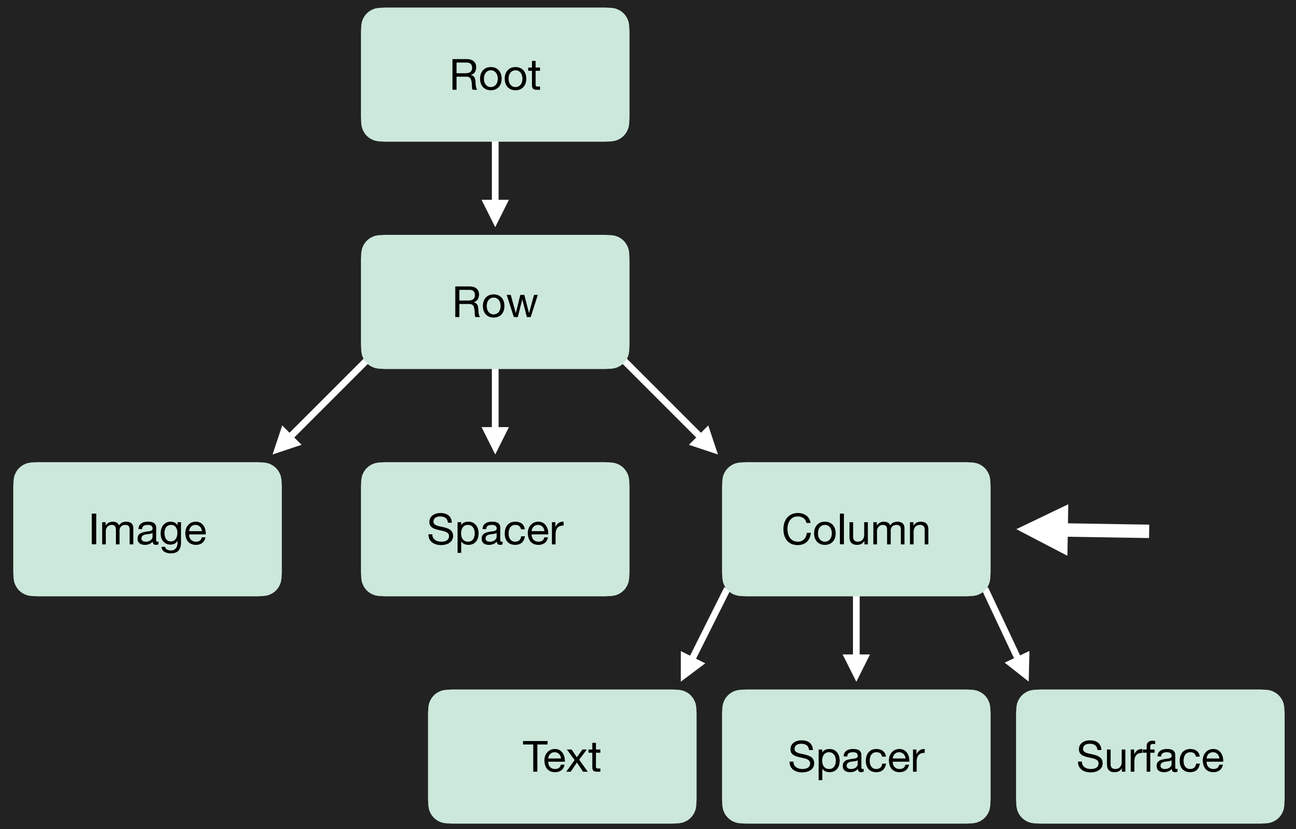
```
@Composable
fun MessageCard(…) {
  Row(…) {
    Image(…)
    Spacer(…)

    Column {
      Text(…)

      Spacer(…)

      Surface(…) {
        Text(…)
      }
    }
  }
}
```



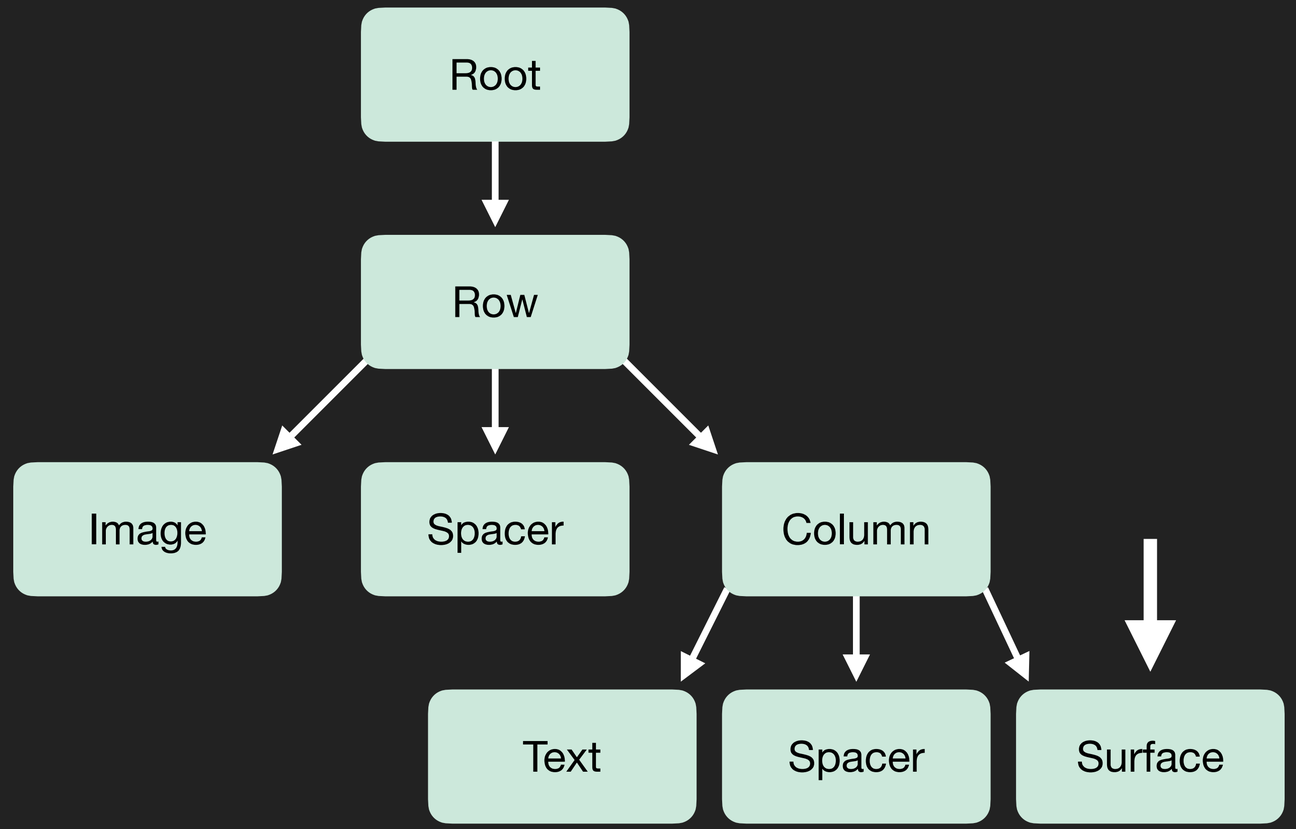
```
@Composable
fun MessageCard(…) {
  Row(…) {
    Image(…)
    Spacer(…)

    Column {
      Text(…)

      Spacer(…)

      Surface(…) {
        Text(…)
      }
    }
  }
}
```



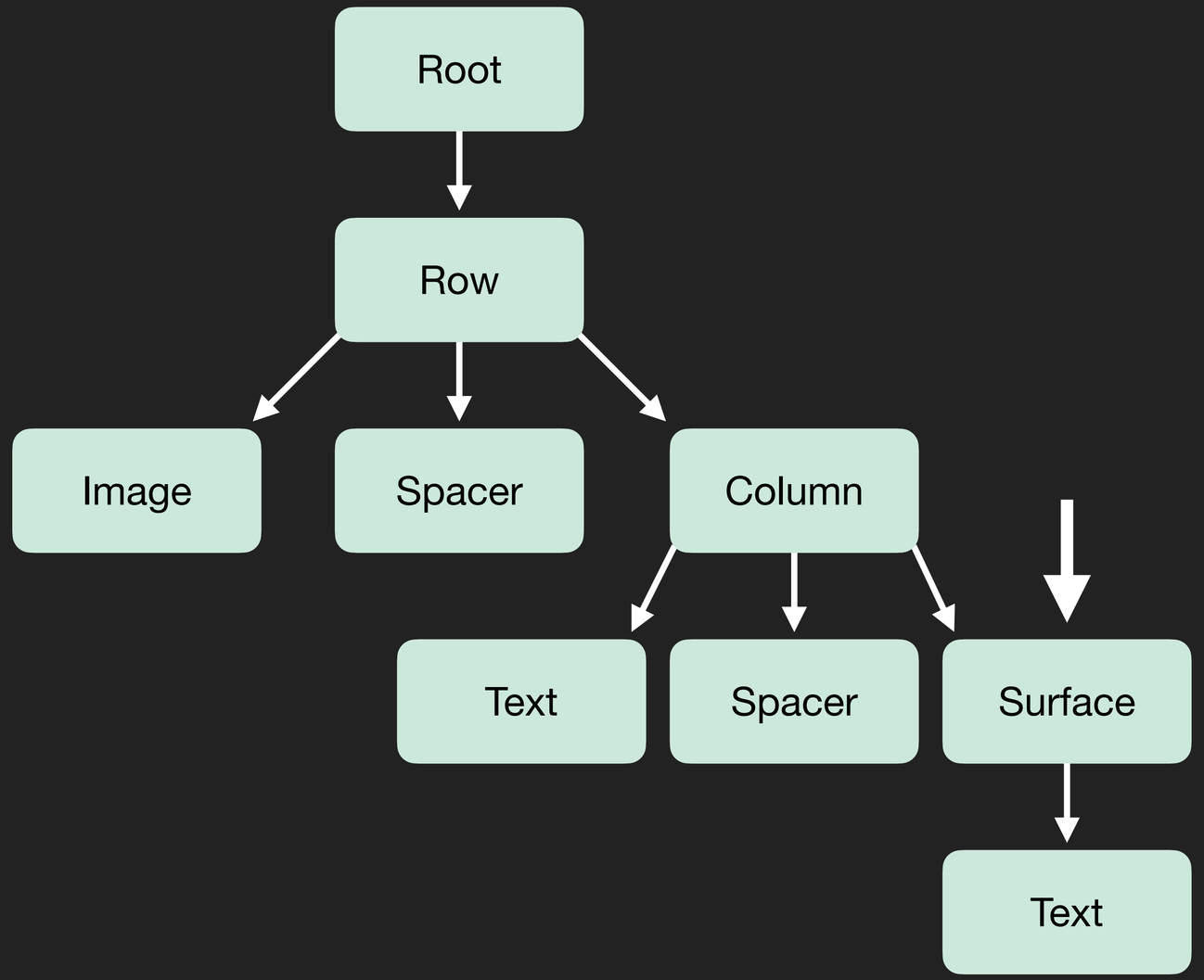
```
@Composable
fun MessageCard(…) {
  Row(…) {
    Image(…)
    Spacer(…)

    Column {
      Text(…)

      Spacer(…)

      Surface(…) {
        Text(…)
      }
    }
  }
}
```



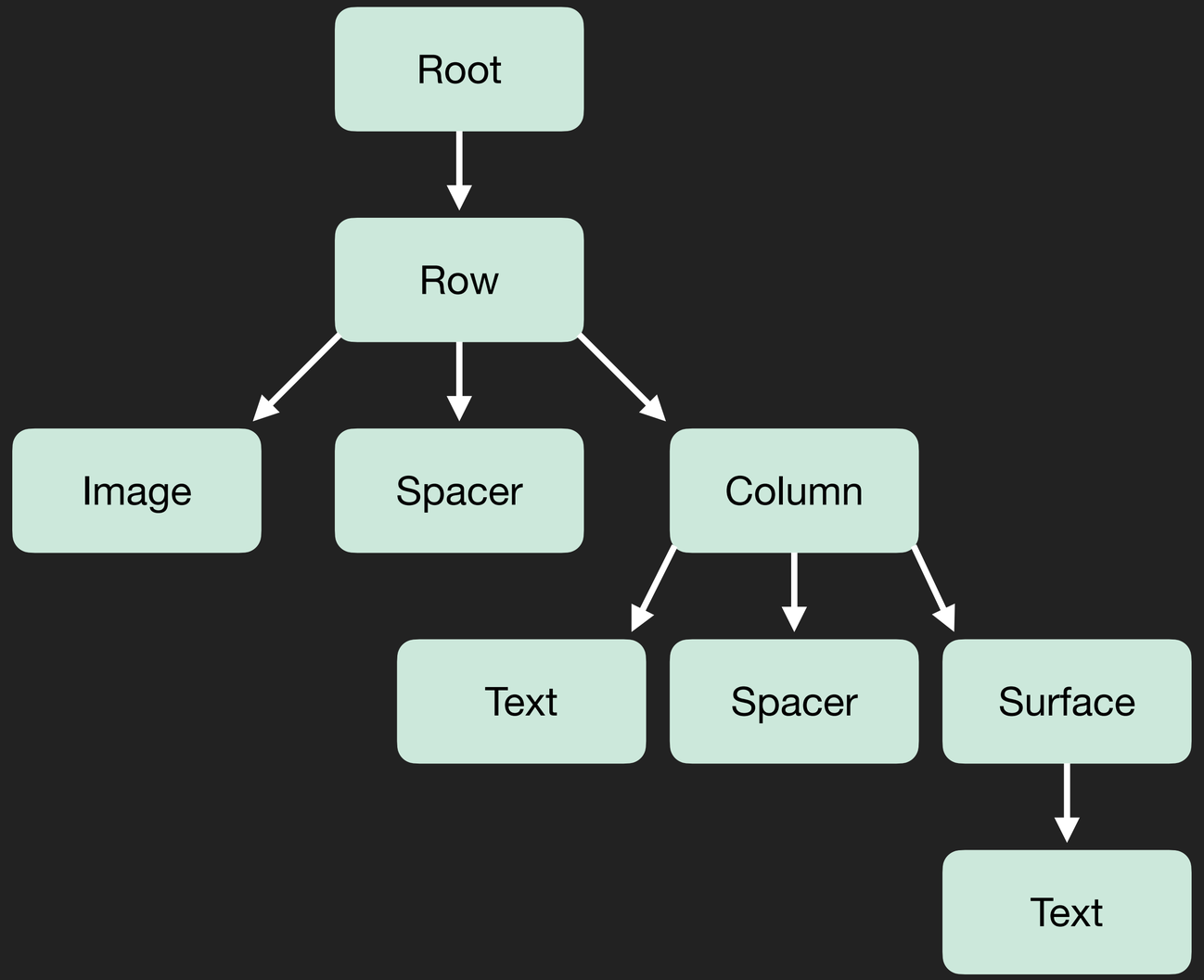
```
@Composable
fun MessageCard(…) {
  Row(…) {
    Image(…)
    Spacer(…)

    Column {
      Text(…)

      Spacer(…)

      Surface(…) {
        Text(…)
      }
    }
  }
}
```



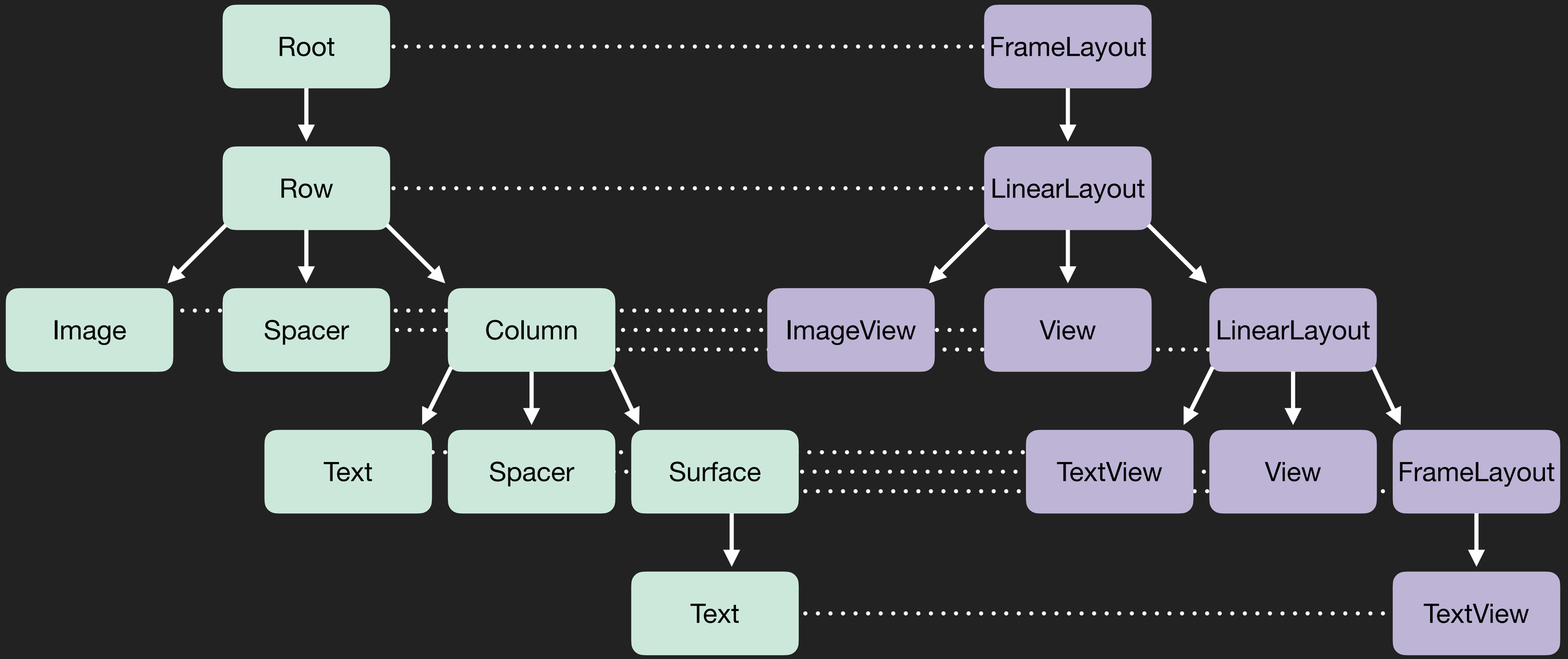
```
@Composable
fun MessageCard(…) {
  Row(…) {
    Image(…)
    Spacer(…)

    Column {
      Text(…)

      Spacer(…)

      Surface(…) {
        Text(…)
      }
    }
  }
}
```

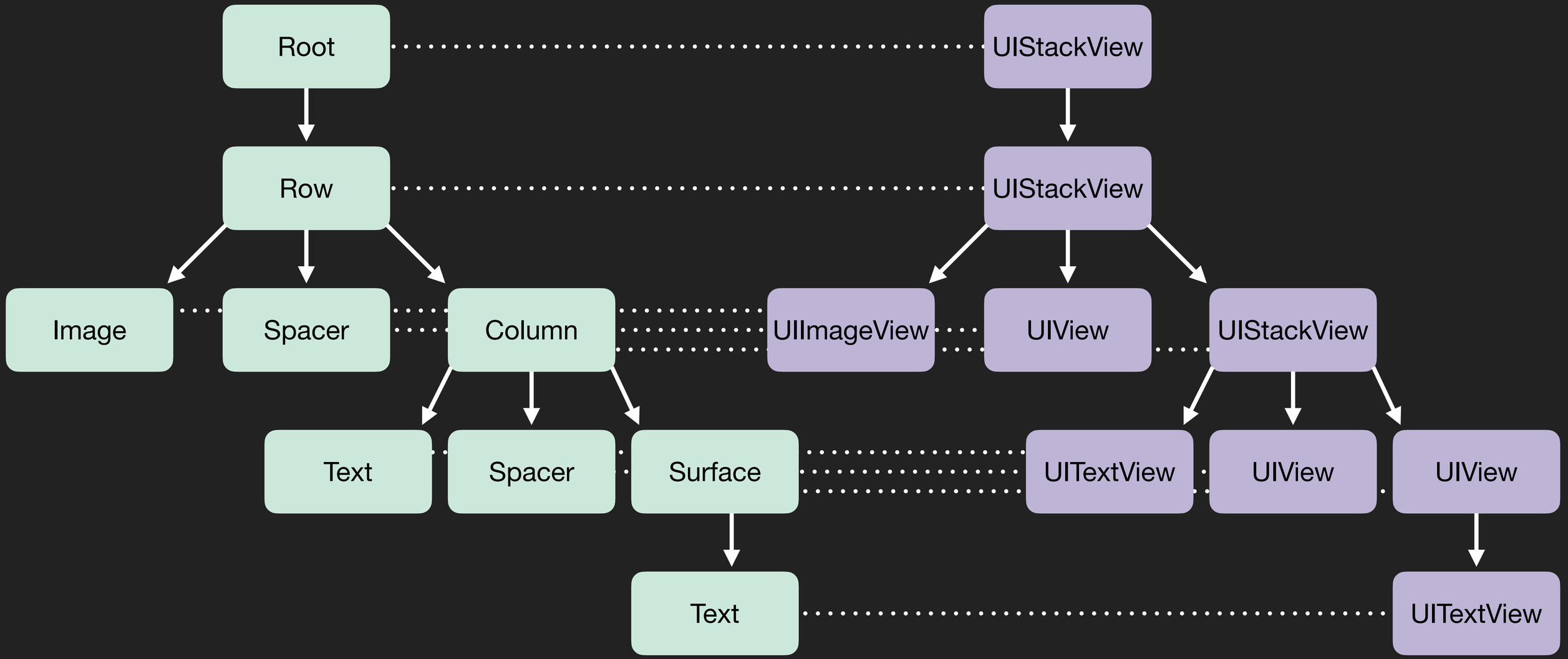


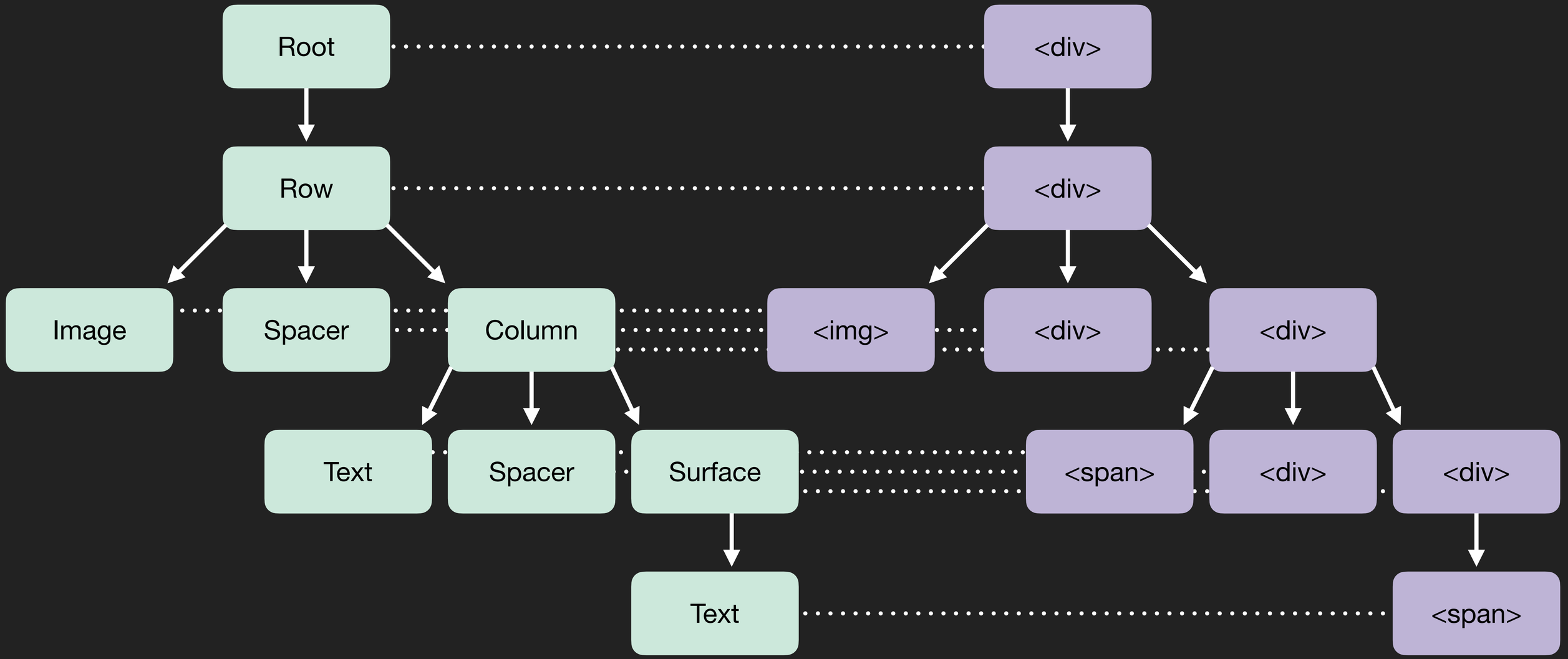
```
@Composable
fun MessageCard(…) {
  Row(…) {
    Image(…)
    Spacer(…)

    Column {
      Text(…)

      Spacer(…)

      Surface(…) {
        Text(…)
      }
    }
  }
}
```




Root
FrameLayout


Root
FrameLayout
Row
LinearLayout
Image
Spacer
Column
ImageView
View
LinearLayout
Text
Spacer
Surface
TextView
View
FrameLayout
Text
TextView


Root
Row
Image
Spacer
Column
Text
Spacer
Surface
Text
UIStackView
UIStackView
UIImageView
UIView
UIStackView
UITextView
UIView
UIView
UITextView


Root
Row
Image
Spacer
Column
Text
Spacer
Surface
Text

Redwood Counter Sample

```
data class Text(
  val text: String?,
)

data class Button(
  val text: String?,
  val enabled: Boolean = true,
  val onClick: (() -> Unit)? = null,
)
```

Redwood Counter Sample

```
@Composable
fun Counter(value: Int = 0) {
  var count by remember { mutableStateOf(value) }

  Column {
    Button("-1", onClick = { count-- })
    Text("Count: $count")
    Button("+1", onClick = { count++ })
  }
}
```

Redwood Counter Sample

```
@Composable
fun Counter(value: Int = 0) {
  var count by remember { mutableStateOf(value) }

  Column {
    Button("-1", onClick = { count-- })
    Text("Count: $count")
    Button("+1", onClick = { count++ })
  }
}
```

Redwood Counter Sample

```
@Composable
fun Counter(value: Int = 0) {
  var count by remember { mutableStateOf(value) }

  Column {
    Button("-1", onClick = { count-- })
    Text("Count: $count")
    Button("+1", onClick = { count++ })
  }
}
```

Redwood Counter Sample

```
@Composable
fun Counter(value: Int = 0) {
  var count by remember { mutableStateOf(value) }

  Column {
    Button("-1", onClick = { count-- })
    Text("Count: $count")
    Button("+1", onClick = { count++ })
  }
}
```

Redwood Counter Sample

```
@Composable
fun Counter(value: Int = 0) {
  var count by remember { mutableStateOf(value) }

  Column {
    Button("-1", onClick = { count-- })
    Text("Count: $count")
    Button("+1", onClick = { count++ })
  }
}
```

Redwood Counter Sample

```
class AndroidText(
  override val value: TextView,
) : Text<View> {
  override fun text(text: String?) {
    value.text = text
  }
}
```

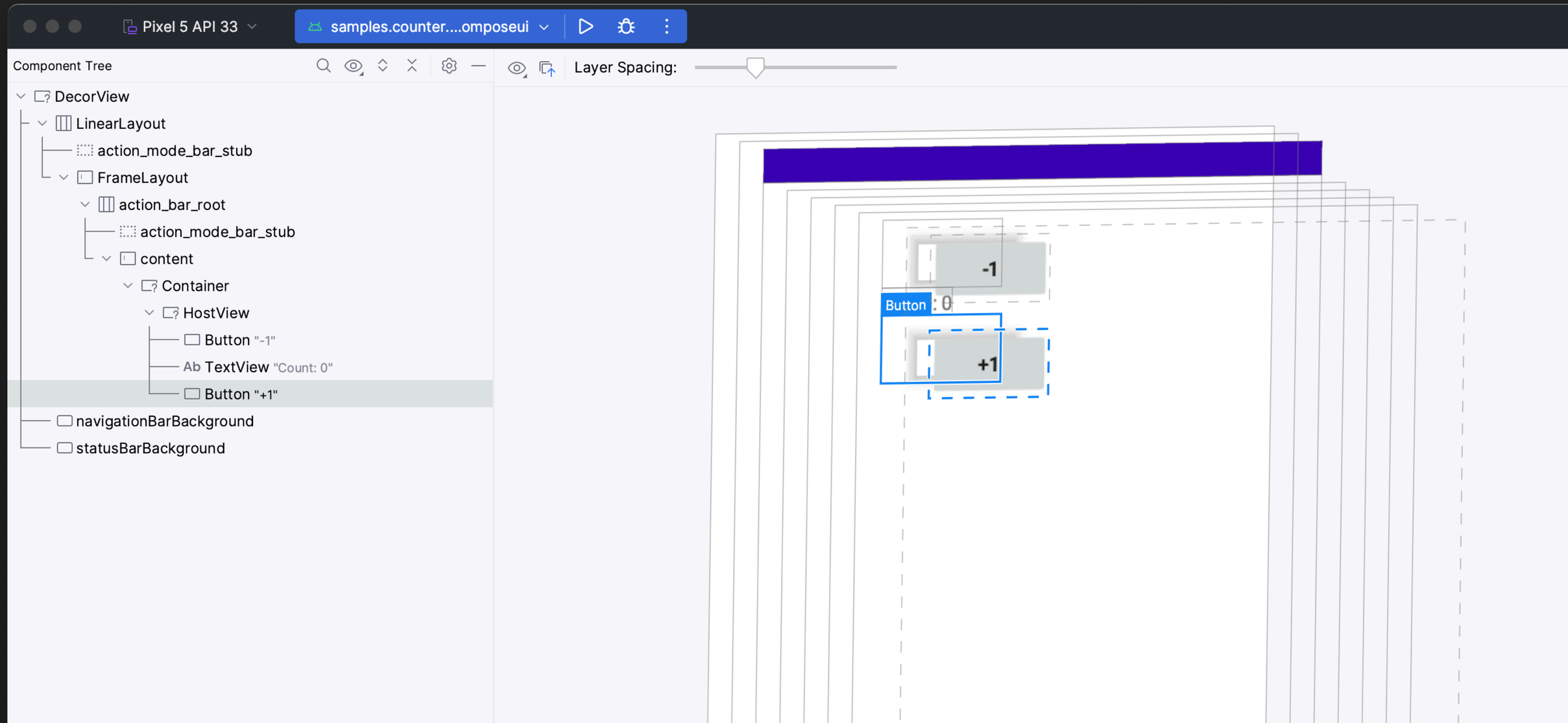
Redwood Counter Sample

```
val redwoodLayout = RedwoodLayout(this)
val composition = RedwoodComposition(
  scope = mainScope,
  view = redwoodLayout,
  provider = SchemaWidgetFactories(
    Counter = AndroidCounterWidgetFactory(this),
    RedwoodLayout = ViewRedwoodLayoutWidgetFactory(this),
  ),
)
```

Redwood Counter Sample

```
val redwoodLayout = RedwoodLayout(this)
val composition = RedwoodComposition(
  scope = mainScope,
  view = redwoodLayout,
  provider = SchemaWidgetFactories(
    Counter = AndroidCounterWidgetFactory(this),
    RedwoodLayout = ViewRedwoodLayoutWidgetFactory(this),
  ),
)

composition.setContent {
  Counter()
}
```

Redwood Counter Sample



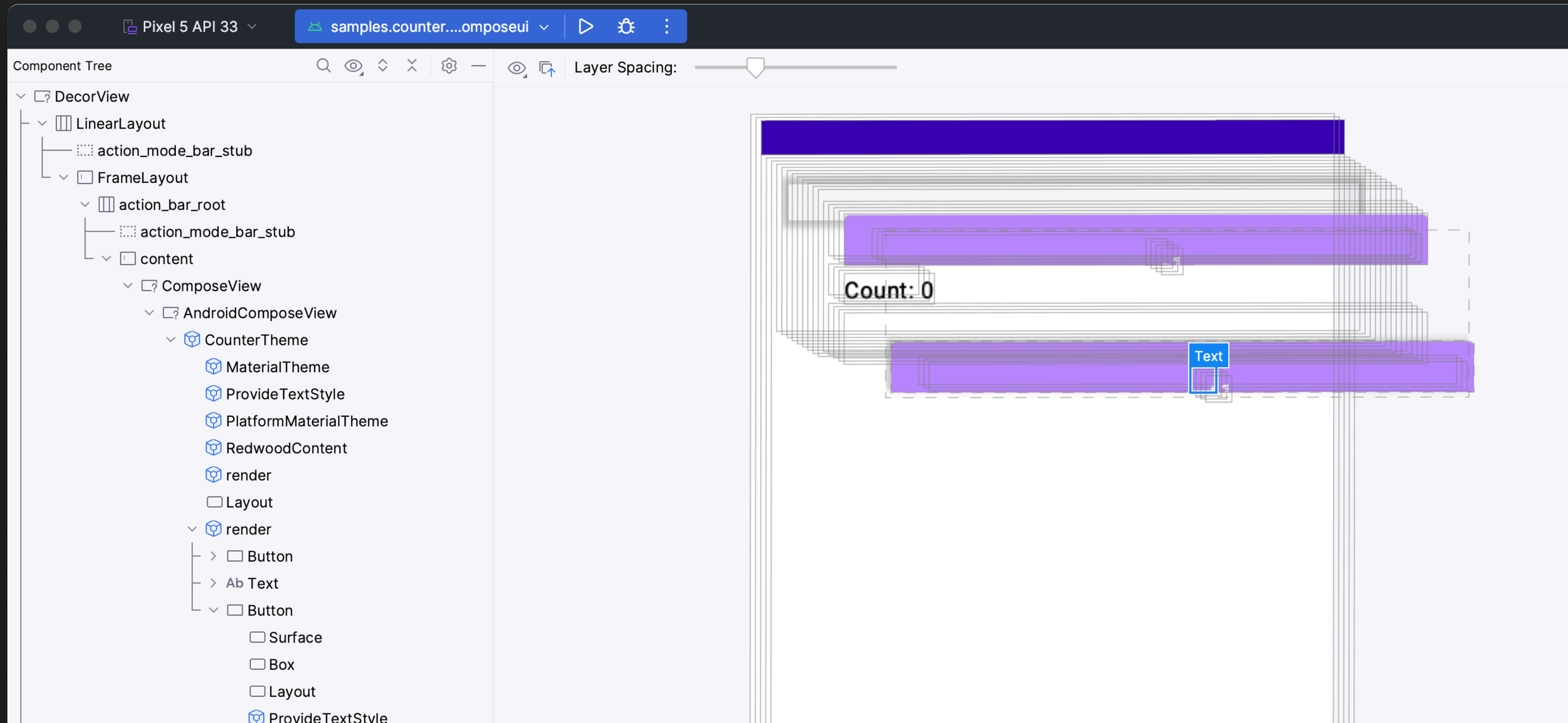
Redwood Counter Sample

```
class ComposeUiText : Text<@Composable () -> Unit> {
  private var text by mutableStateOf("")

  override val value = @Composable {
    Text(text = text)
  }

  override fun text(text: String?) {
    this.text = text ?: ""
  }
}
```

Redwood Counter Sample

```
val factories = SchemaWidgetFactories(
  Counter = ComposeUiCounterWidgetFactory,
  RedwoodLayout = ComposeUiRedwoodLayoutWidgetFactory(),
)

setContent {
  CounterTheme {
    RedwoodContent(factories) {
      Counter()
    }
  }
}
```

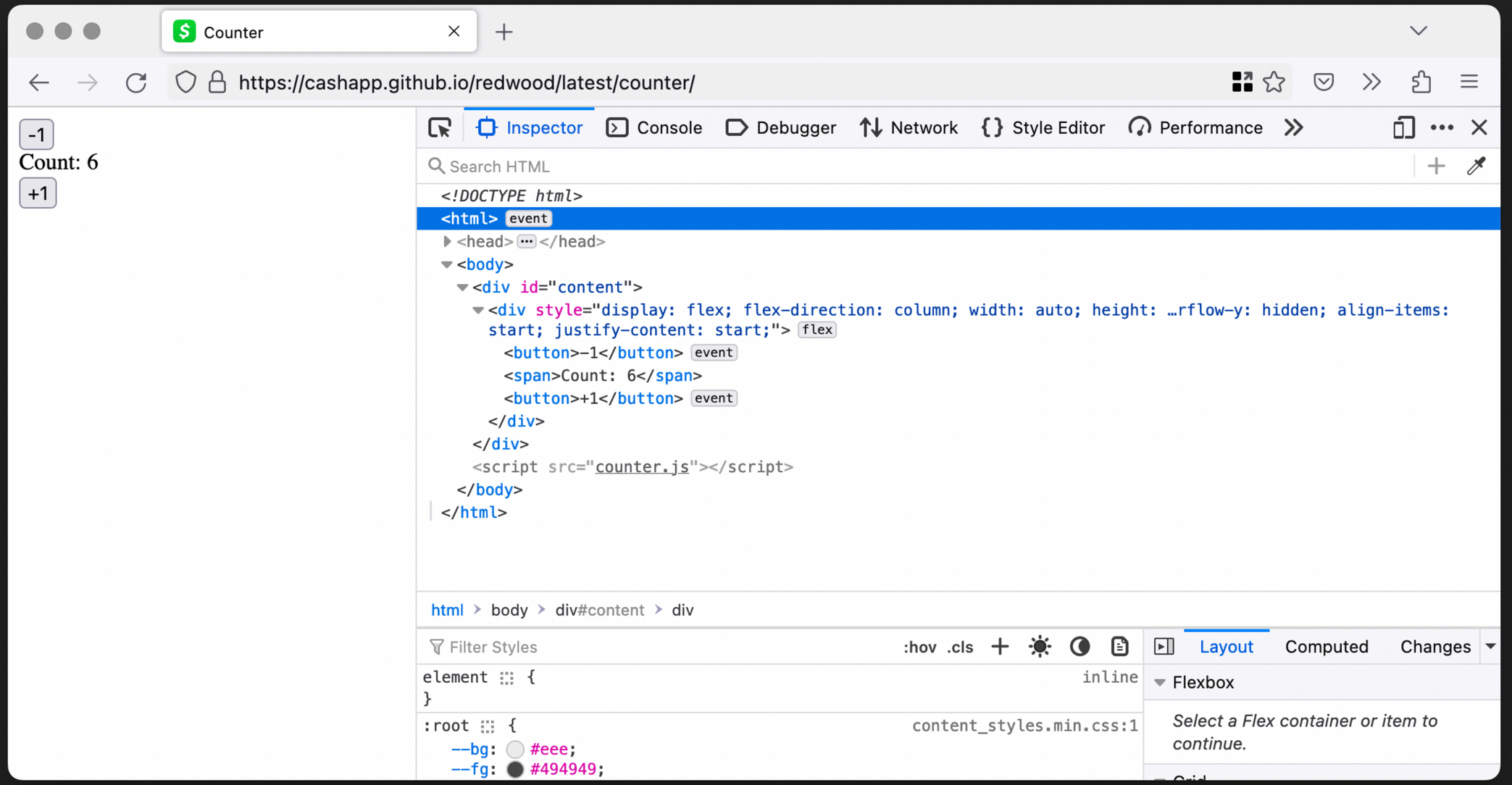
Redwood Counter Sample



Redwood Counter Sample

```
class HtmlText(
  override val value: HTMLSpanElement,
) : Text<HTMLElement> {
  override fun text(text: String?) {
    value.textContent = text
  }
}
```

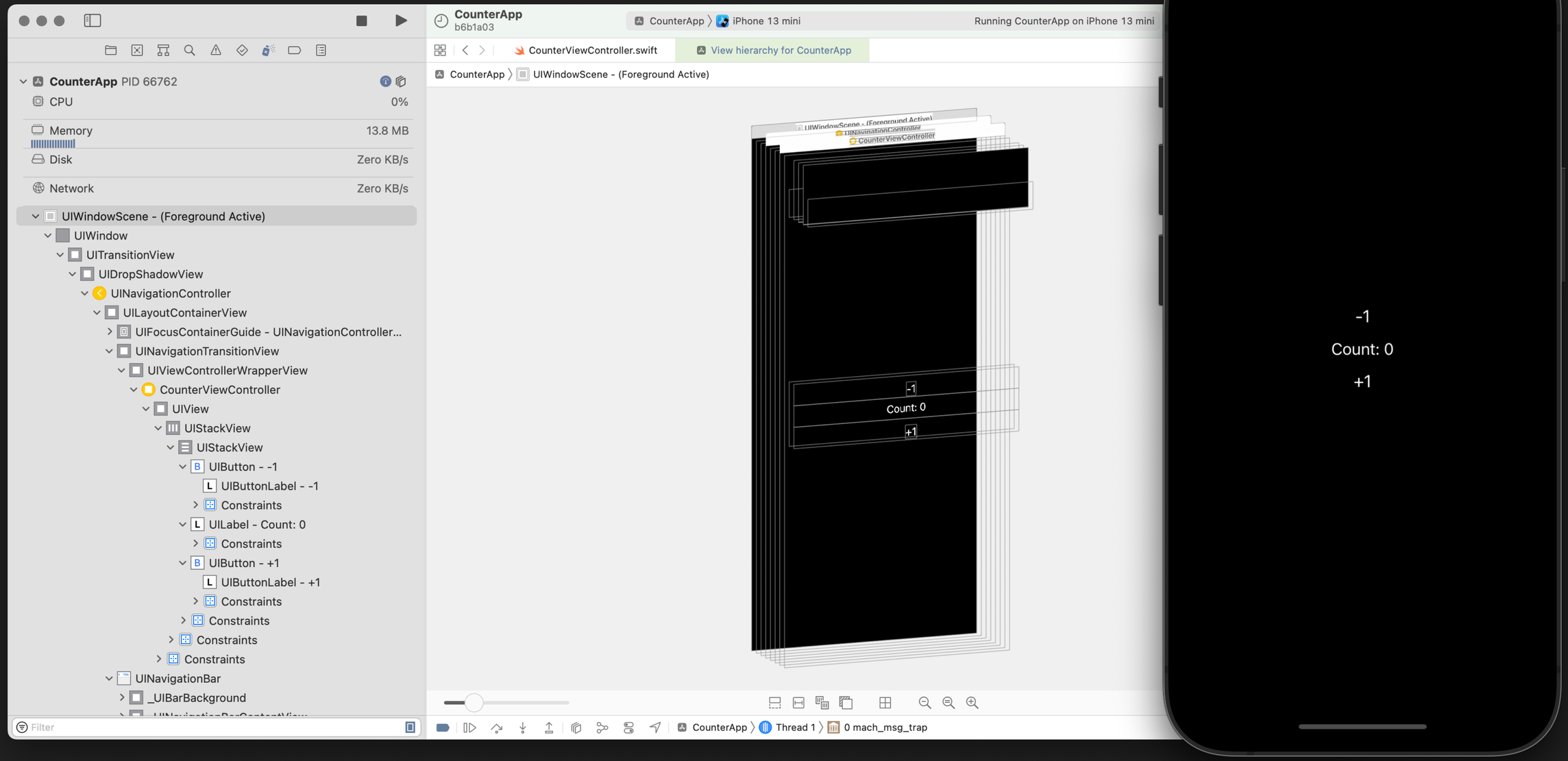
Redwood Counter Sample



Redwood Counter Sample

```
class IosText : Text<UIView> {
  override val value = UILabel().apply {
    textAlignment = NSTextAlignmentCenter
  }

  override fun text(text: String?) {
    value.text = text
  }
}
```

Redwood Counter Sample



Zipline

Redwood


Treehouse
Zipline
Redwood

Treehouse Counter Sample

```
val redwoodLayout = RedwoodLayout(this)
val composition = RedwoodComposition(
  scope = mainScope,
  view = redwoodLayout,
  provider = SchemaWidgetFactories(
    Counter = AndroidCounterWidgetFactory(this),
    RedwoodLayout = ViewRedwoodLayoutWidgetFactory(this),
  ),
)

composition.setContent {
  Counter()
}
```

Treehouse Counter Sample

```
val redwoodLayout = RedwoodLayout(this)
val composition = RedwoodComposition(
  scope = mainScope,
  view = redwoodLayout,
  provider = SchemaWidgetFactories(
    Counter = AndroidCounterWidgetFactory(this),
    RedwoodLayout = ViewRedwoodLayoutWidgetFactory(this),
  ),
)

composition.setContent {
  Counter()
}
```

Treehouse Counter Sample

```
val redwoodLayout = RedwoodLayout(this)
val composition = RedwoodComposition(
  scope = mainScope,
  view = redwoodLayout,
  provider = SchemaWidgetFactories(
    Counter = AndroidCounterWidgetFactory(this),
    RedwoodLayout = ViewRedwoodLayoutWidgetFactory(this),
  ),
)

composition.setContent {
  Counter()
}
```

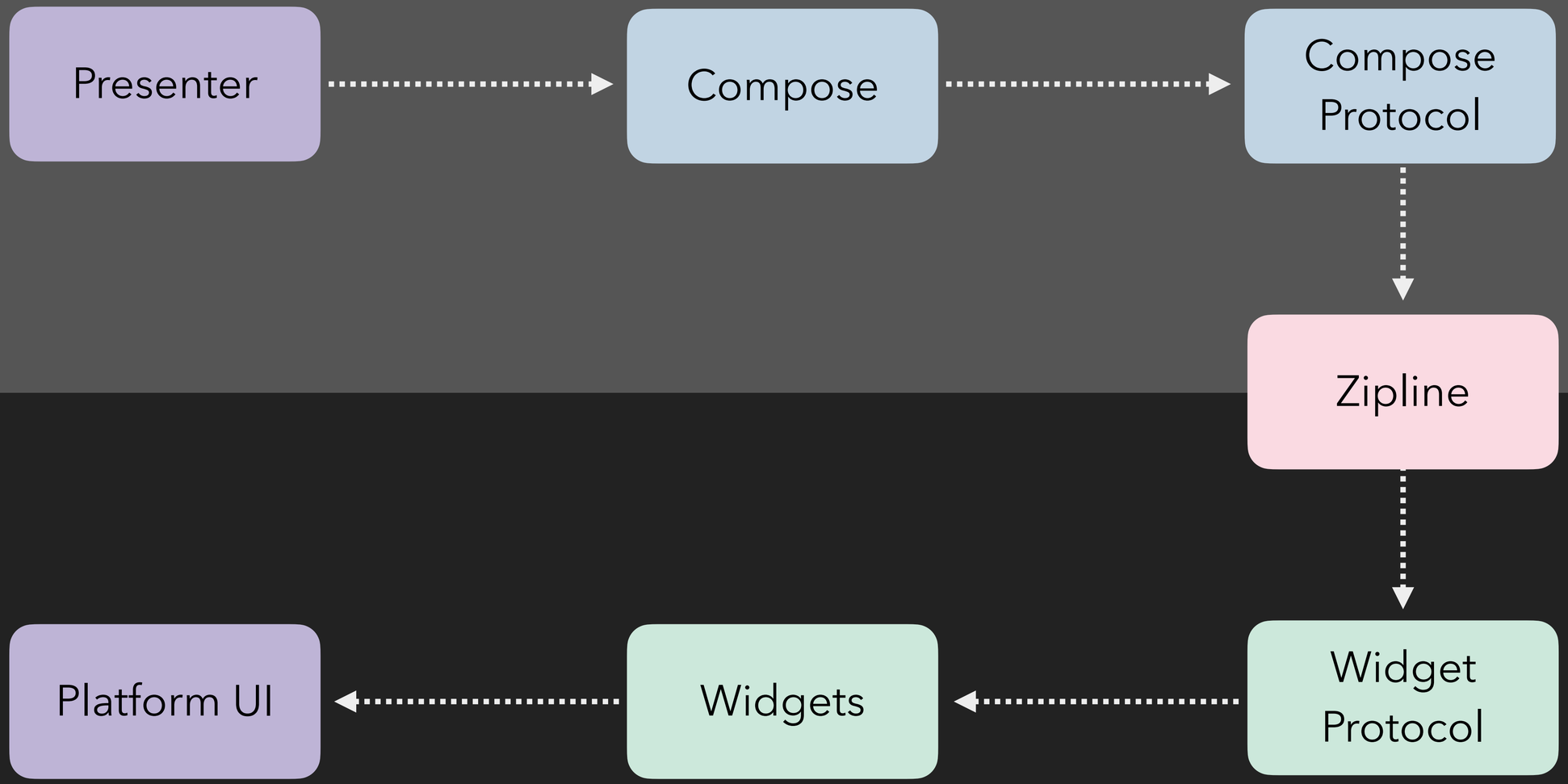
Treehouse Counter Sample

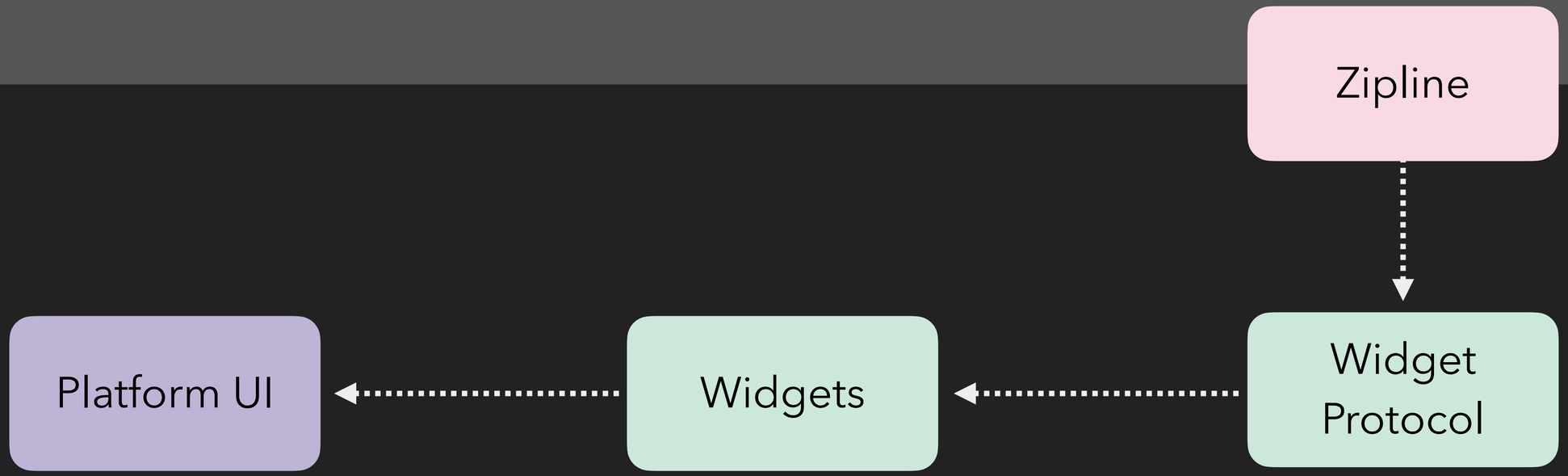
```
val redwoodLayout = RedwoodLayout(this)
val composition = RedwoodComposition(
  scope = mainScope,
  view = redwoodLayout,
  provider = SchemaWidgetFactories(
    Counter = AndroidCounterWidgetFactory(this),
    RedwoodLayout = ViewRedwoodLayoutWidgetFactory(this),
  ),
)

composition.setContent {
  Counter()
}
```

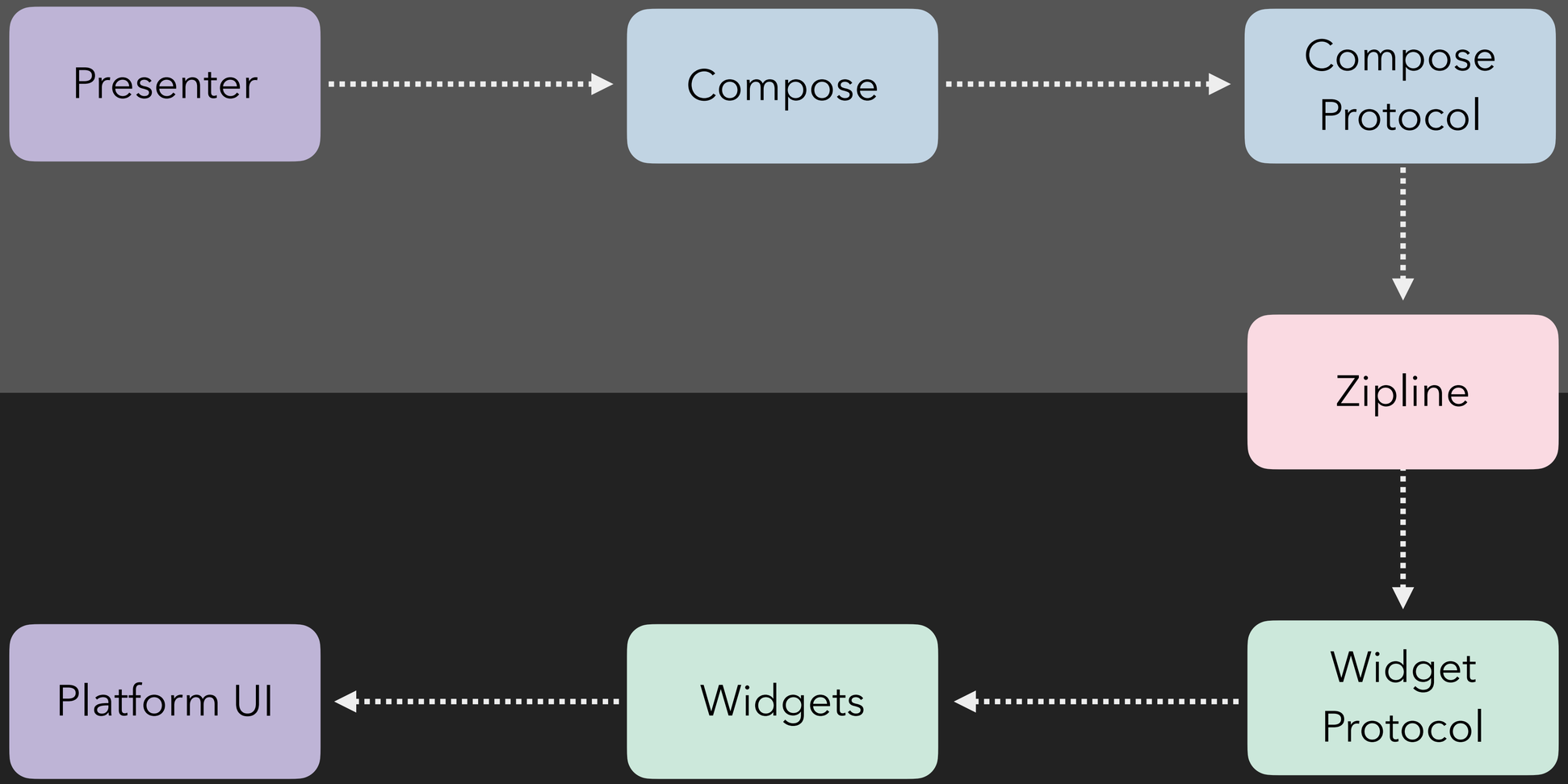
```
val composition = RedwoodComposition(
  scope = mainScope,
)
composition.setContent {
  Counter()
}
```

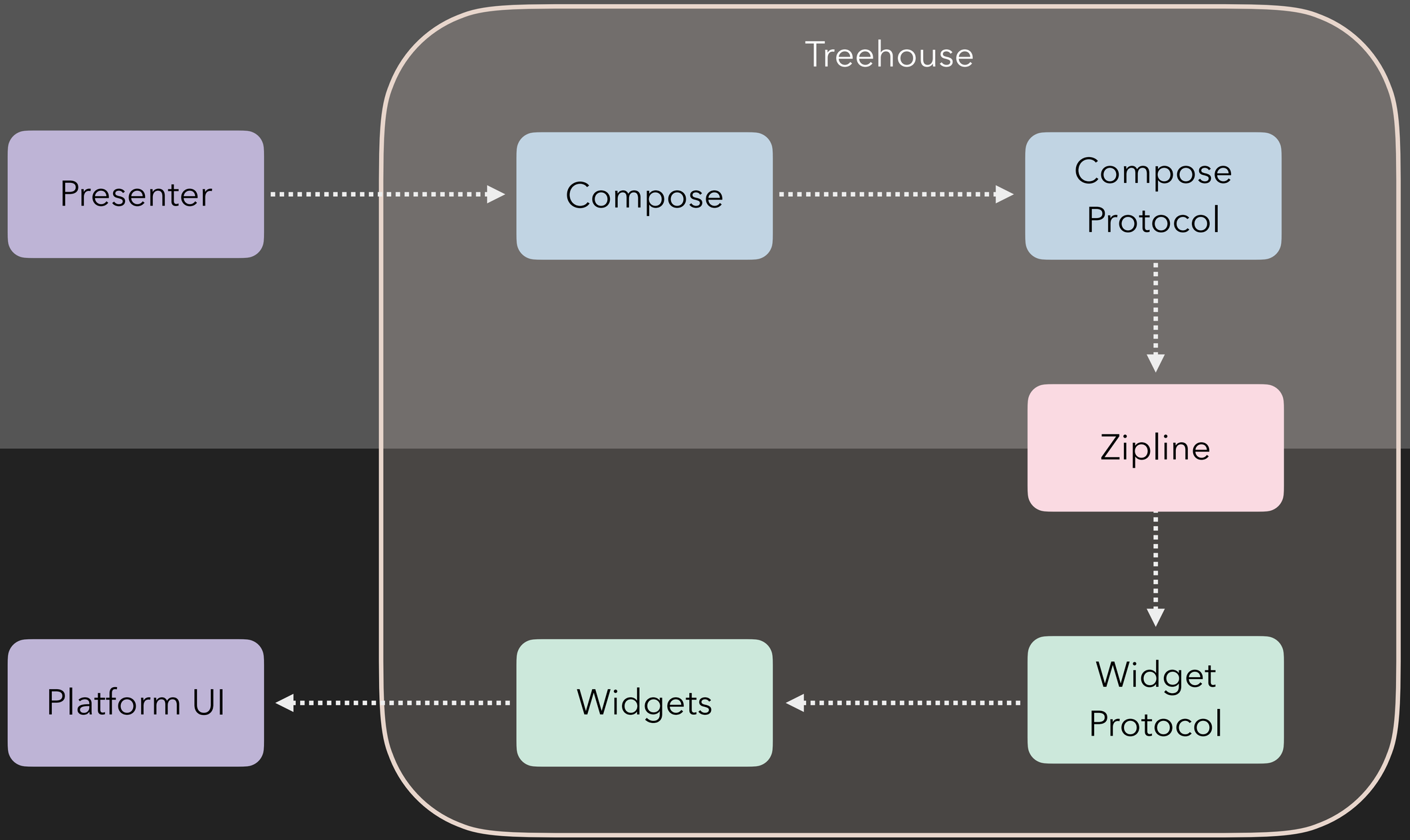
JS

```
val redwoodLayout = RedwoodLayout(this)
val rendering = RedwoodRendering(
  view = redwoodLayout,
  provider = SchemaWidgetFactories(
    Counter = AndroidCounterWidgetFactory(this),
    RedwoodLayout = ViewRedwoodLayoutWidgetFactory(this),
  ),
)
```

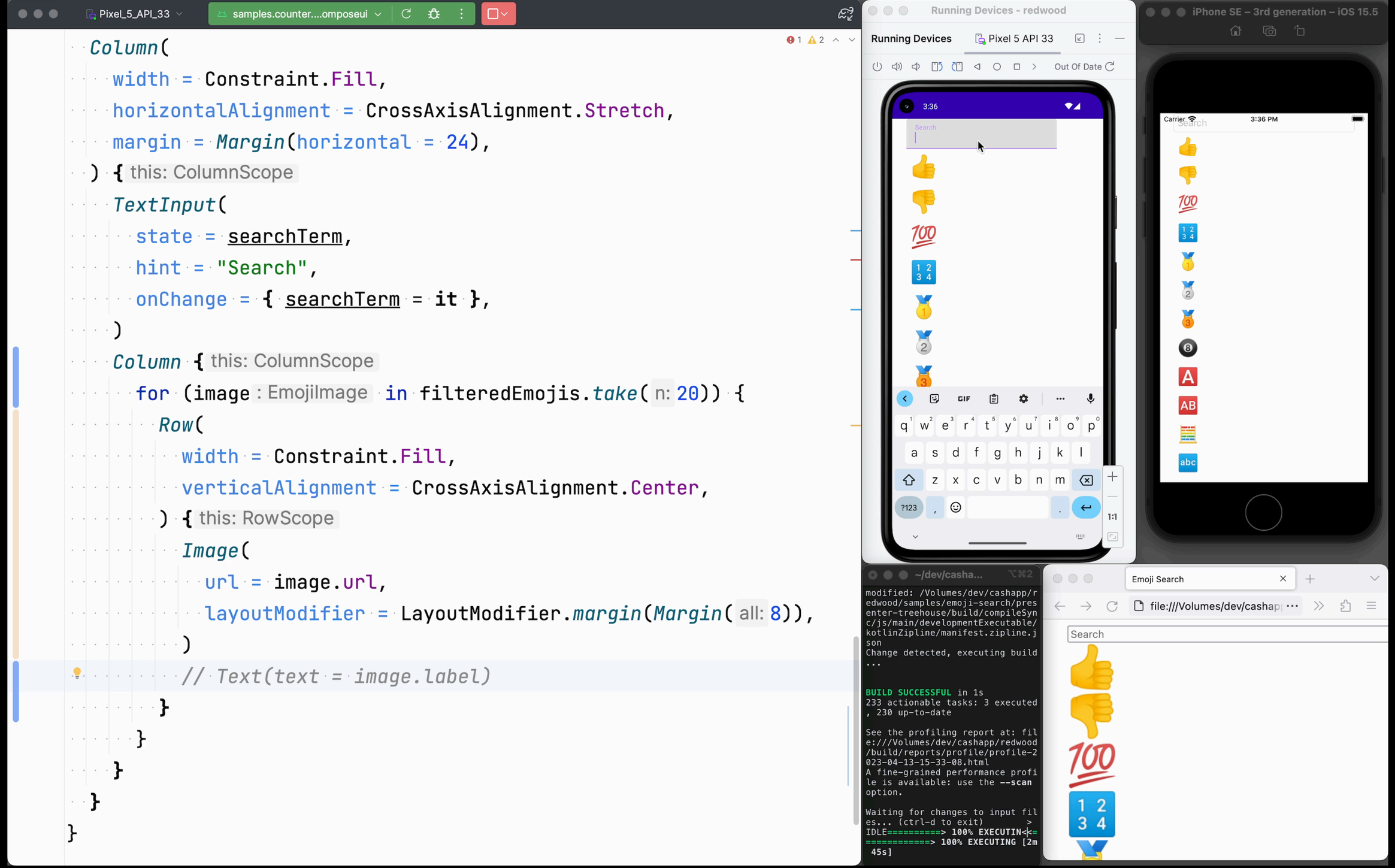

Presenter
Compose
Compose Protocol
Zipline
Widget Protocol
Widgets
Platform UI

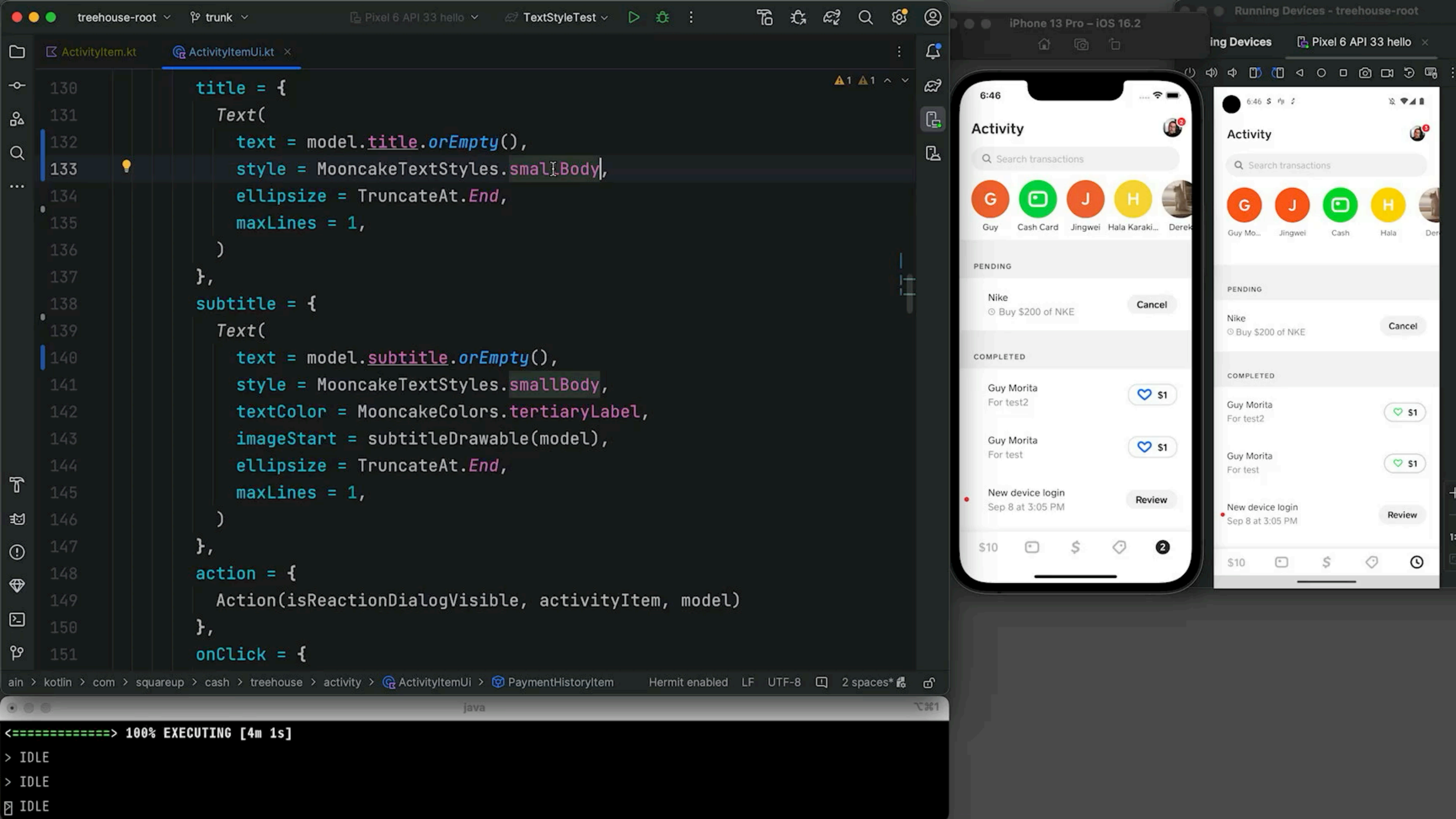

Zipline
Widget
Protocol
Widgets
Platform UI


Presenter
Compose
Compose Protocol
Zipline
Widget Protocol
Widgets
Platform UI

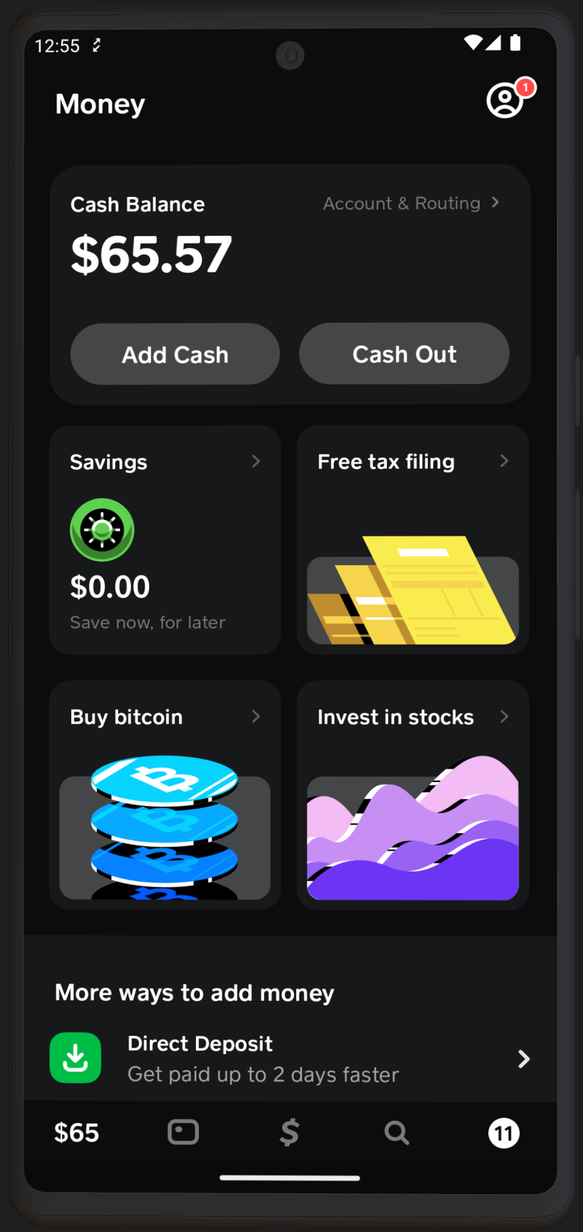

Treehouse
Presenter
Compose
Compose Protocol
Zipline
Widget Protocol
Widgets
Platform UI

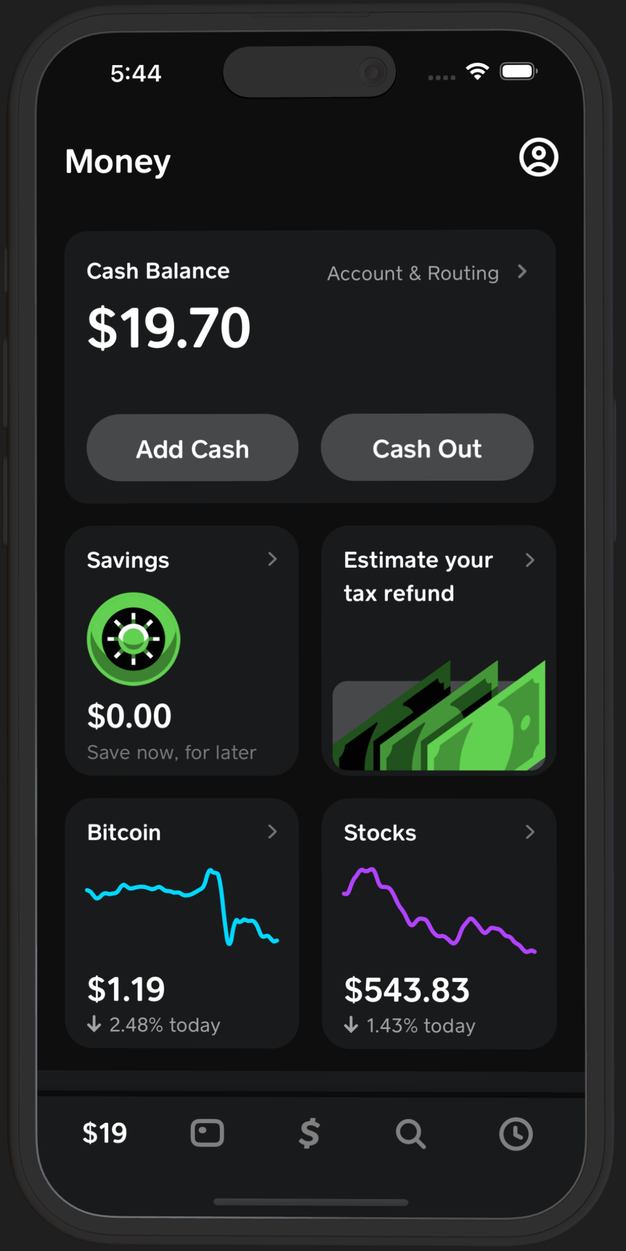
Treehouse Demo

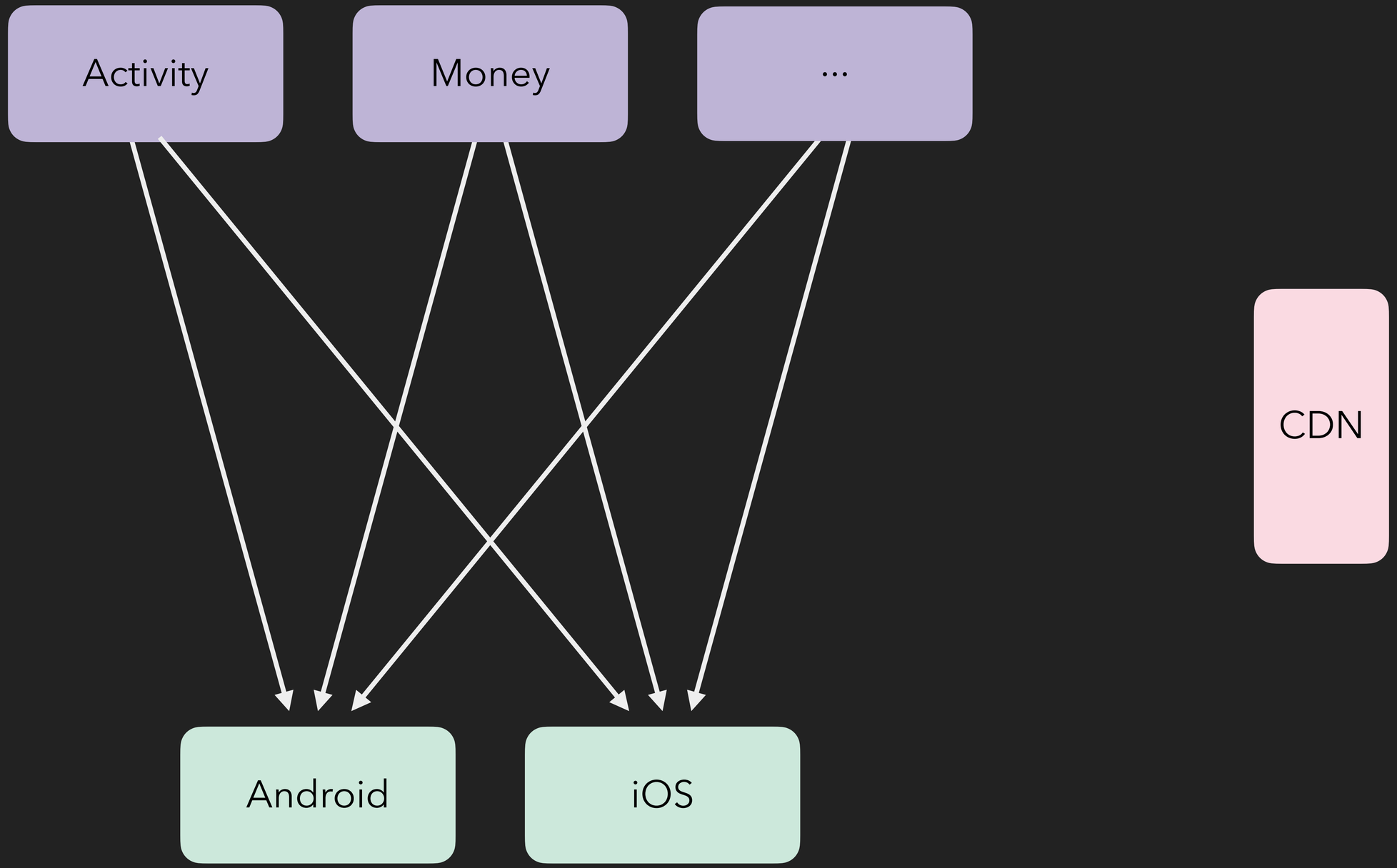

Pixel_5_API_33
samples.counter....omposeui
Column(
width = Constraint.Fill,
horizontalAlignment = CrossAxisAlignment.Stretch,
margin = Margin(horizontal = 24),
) { this: ColumnScope
TextInput(
state = searchTerm,
hint = "Search",
onChange = { searchTerm = it },
)
Column { this: ColumnScope
for (image : EmojiImage in filteredEmojis.take(n: 20)) {
Row(
width = Constraint.Fill,
verticalAlignment = CrossAxisAlignment.Center,
) { this: RowScope
Image(
url = image.url,
layoutModifier = LayoutModifier.margin(Margin(all: 8)),
)
// Text(text = image.label)
}
}
}
}
}
Running Devices - redwood
Running Devices
Pixel 5 API 33
Out Of Date
3:36
Search
iPhone SE – 3rd generation – iOS 15.5
Carrier
3:36 PM
Search
~/dev/casha...
modified: /Volumes/dev/cashapp/redwood/samples/emoji-search/presenter-treehouse/build/compileSync/js/main/developmentExecutable/kotlinZipline/manifest.zipline.json
Change detected, executing build ...
BUILD SUCCESSFUL in 1s
233 actionable tasks: 3 executed, 230 up-to-date
See the profiling report at: file:///Volumes/dev/cashapp/redwood/build/reports/profile/profile-2023-04-13-15-33-08.html
A fine-grained performance profile is available: use the --scan option.
Waiting for changes to input files... (ctrl-d to exit)
IDLE==========> 100% EXECUTING
==========> 100% EXECUTING [2m 45s]
Emoji Search
file:///Volumes/dev/cashap
Search

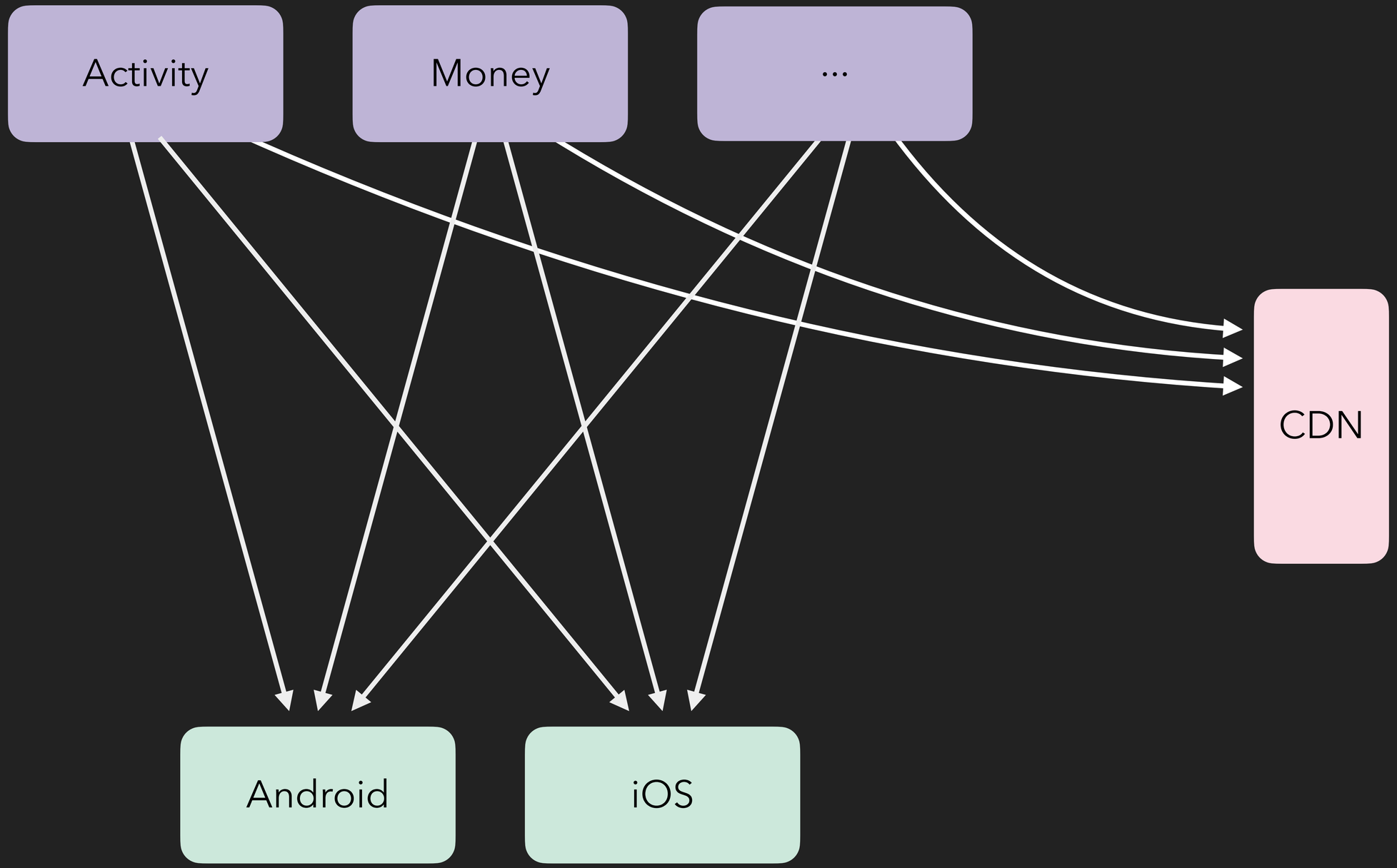

treehouse-root
trunk
Pixel 6 API 33 hello
TextStyleTest
ActivityItem.kt
ActivityItemUi.kt
title = {
Text(
text = model.title.orEmpty(),
style = MooncakeTextStyles.smallBody,
ellipsize = TruncateAt.End,
maxLines = 1,
)
},
subtitle = {
Text(
text = model.subtitle.orEmpty(),
style = MooncakeTextStyles.smallBody,
textColor = MooncakeColors.tertiaryLabel,
imageStart = subtitleDrawable(model),
ellipsize = TruncateAt.End,
maxLines = 1,
)
},
action = {
Action(isReactionDialogVisible, activityItem, model)
},
onClick = {
ain > kotlin > com > squareup > cash > treehouse > activity > ActivityItemUi > PaymentHistoryItem
Hermit enabled
LF
UTF-8
2 spaces*
java
<==========> 100% EXECUTING [4m 1s]
> IDLE
> IDLE
> IDLE
iPhone 13 Pro – iOS 16.2
Running Devices - treehouse-root
Pixel 6 API 33 hello
6:46
Activity
Search transactions
Guy
Cash Card
Jingwei
Hala Karaki...
PENDING
Nike
Buy $200 of NKE
Cancel
COMPLETED
Guy Morita
For test2
$1
Guy Morita
For test
$1
New device login
Sep 8 at 3:05 PM
Review
$10

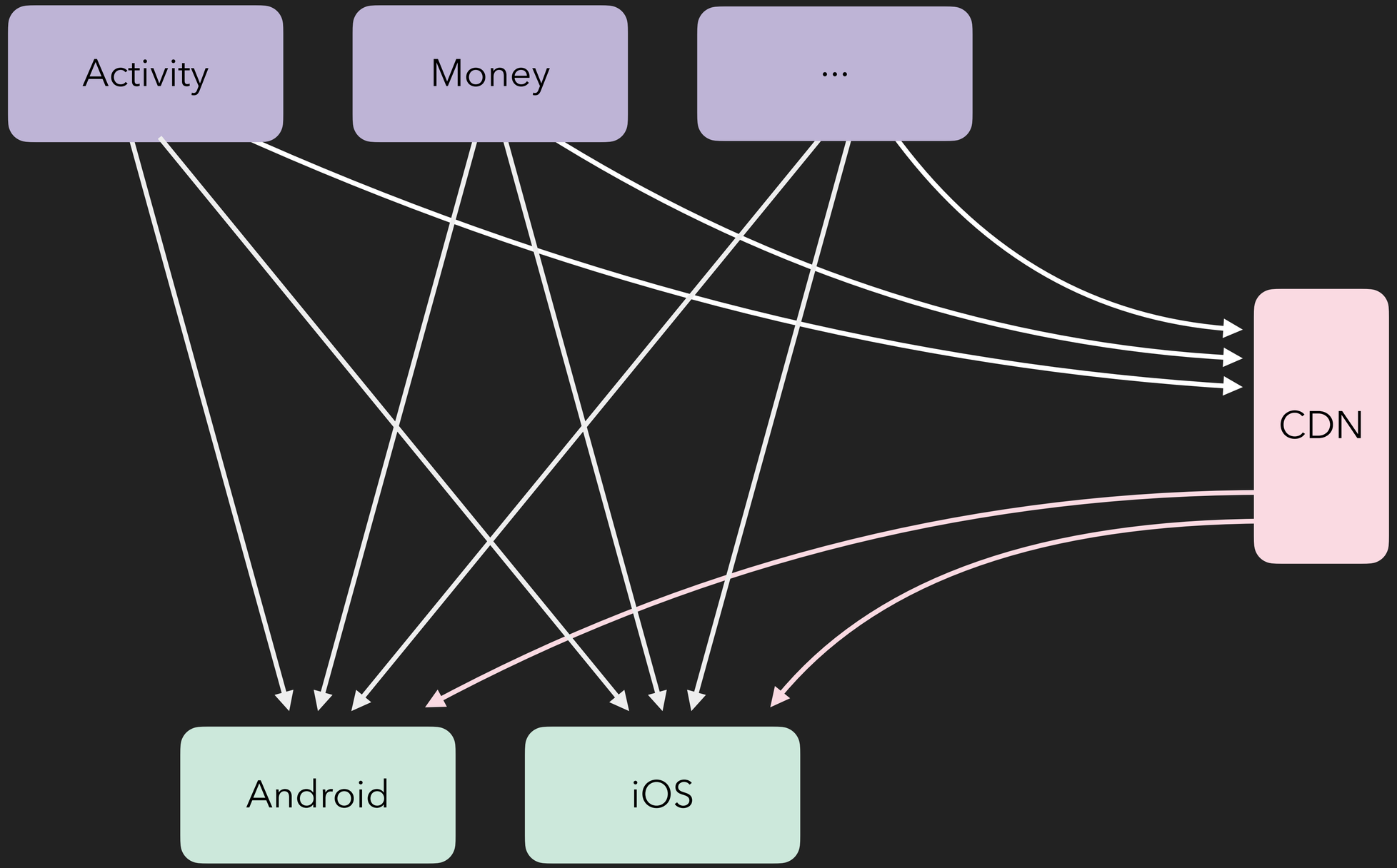
Treehouse Demo

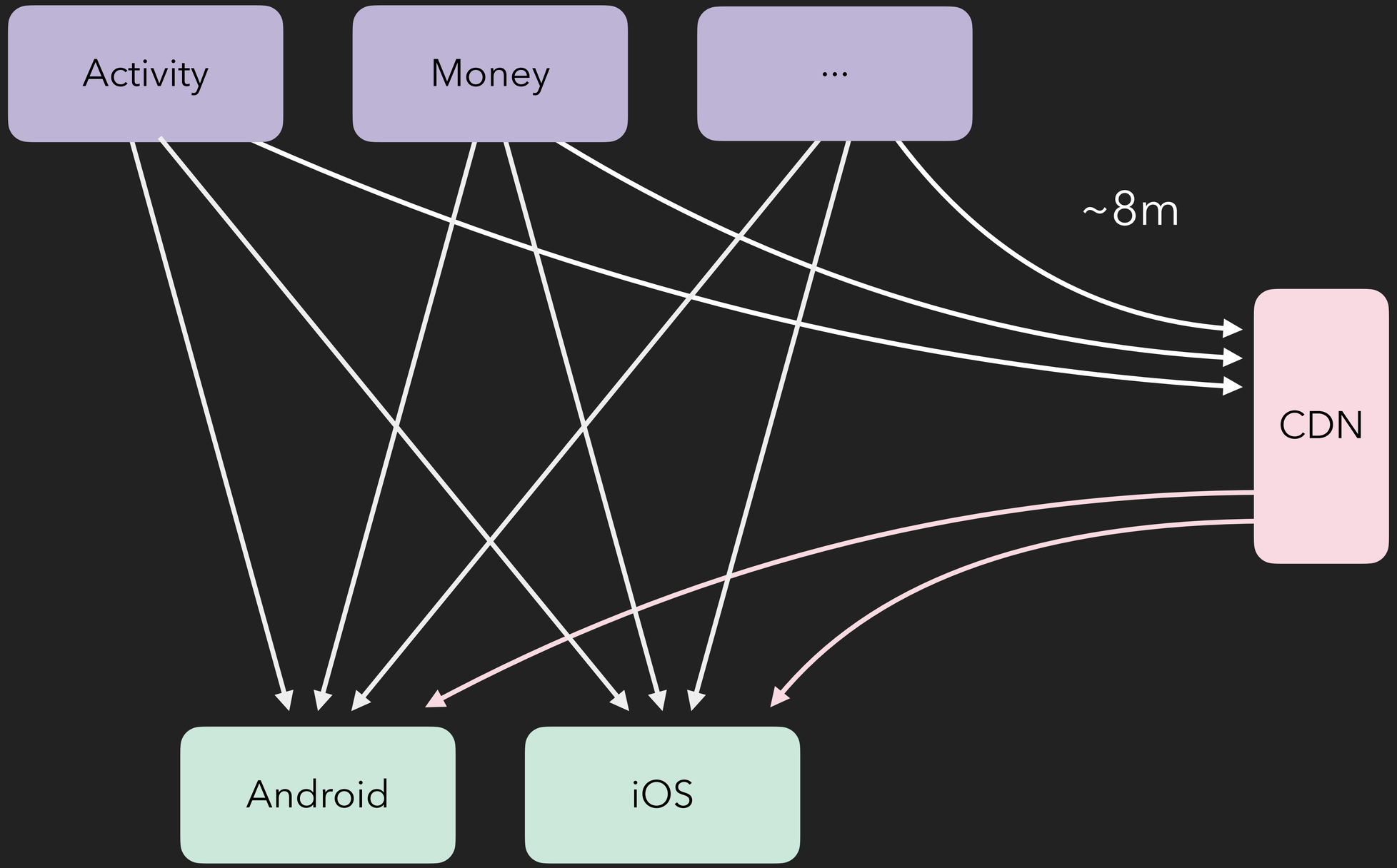





Activity
Money
...
CDN
Android
iOS


Activity
Money
...
CDN
Android
iOS


Activity
Money
...
CDN
Android
iOS


Activity
Money
...
~8m
CDN
Android
iOS

.com/**cashapp/redwood**

.com/**cashapp/zipline**

Releasing faster with Kotlin multiplatform

@jw@jakewharton.com