Type Driven Development

ABC

Arundev SinthanaiSirrpi

Senior iOS Engineer

@arunsirrpi

Outline

- Background
- Domain dictionary
- App

Background

- Australian Broadcasting Corporation - ABC
- Technology Division
- 12 developers
- Building native apps
- Swift

http://about.abc.net.au/abc-history/

Background

Before we code

- Overall architecture
- Sequence diagrams
- Domain dictionary

Domain dictionary

Defines

- Terminology
- Requirements
- Functionality

Background

Audience research

TV

Radio

News

Technology

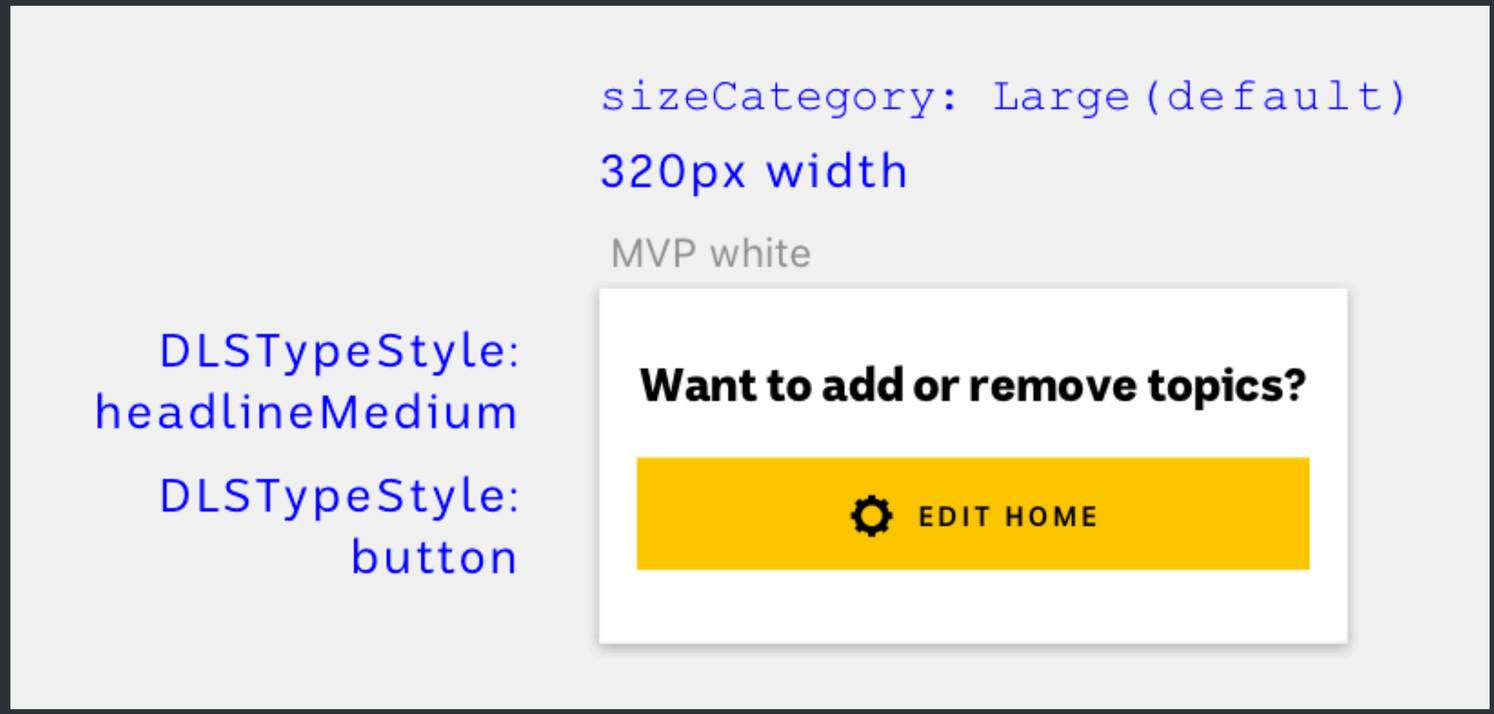
How Domain dictionary helps

- Communication / Single source of truth
- Create Types

Domain Dictionary

	xSmall		Small		Medium		Large (Default)	
Style & Weight	Point size	line-height	Point size	line-height	Point size	line-height	Point size	line-height
Headline X-Large (Black)	29	37	30	38	31	39	32	40
Headline Large (Black)	21	29	22	30	23	31	24	32
Headline Medium (Black)	17	21	18	22	19	23	20	24
Headline Small (Black)	15	21	16	22	17	23	18	24
Headline X-Small (Bold)	15	21	16	22	17	23	18	24
Lead (Regular)	21	29	22	30	23	31	24	32
Body (Regular)	13	21	14	22	15	23	16	24
Body-card (Regular)	11	21	12	22	13	23	14	24
By Line (Bold)	11	21	12	22	13	23	14	24
Quote (Regular)	21	29	22	30	23	31	24	32
Toolbar (Regular)	17	21	18	22	19	23	20	24
Utility + Time (Regular)	11	21	11	22	11	23	12	24
Topic + Button (Bold)	11	21	11	22	11	23	12	24

App


sizeCategory: Large (default)
320px width
MVP white
DLSTypeStyle: headlineMedium
DLSTypeStyle: button
Want to add or remove topics?
EDIT HOME

```
let headlineXLarge = "headlineXLarge"
let headlineLarge  = "headlineLarge"
let headlineMedium = "headlineMedium"
   . . .


func apply(style textStyle: String,
                  to text: String) -> NSAttributedString
```

Background Domain Dictionary App SDK

```
func headline(_ text: String)-> NSAttributedString { …
func body(_ text: String) -> NSAttributedString { …
func lead(_ text: String) -> NSAttributedString { …
func byline(_ text: String) -> NSAttributedString { …
```

Background Domain Dictionary App SDK

Product Type

```
public struct TextStyleAttributes {

    public let pointSize: Int8
    public let lineHeight: Int8

}
```

	xSmall	
Style & Weight	Point size	line-height
Headline X-Large (Black)	29	37
Headline Large (Black)	21	29
Headline Medium (Black)	17	21
Headline Small (Black)	15	21

Sum type

```
public enum TextStyle: String {

    case headlineXLarge = "headlineXLarge"
    case headlineLarge  = "headlineLarge"
    case headlineMedium = "headlineMedium"
    case headlineSmall  = "headlineSmall"
    case headlineXSmall = "headlineXSmall"
    . . .

    var attributes: TextStyleAttributes
```

	Style & Weight	Point size	line-height
5	Headline X-Large (Black)	29	37
6	Headline Large (Black)	21	29
7	Headline Medium (Black)	17	21
8	Headline Small (Black)	15	21
9	Headline X-Small (Bold)	15	21

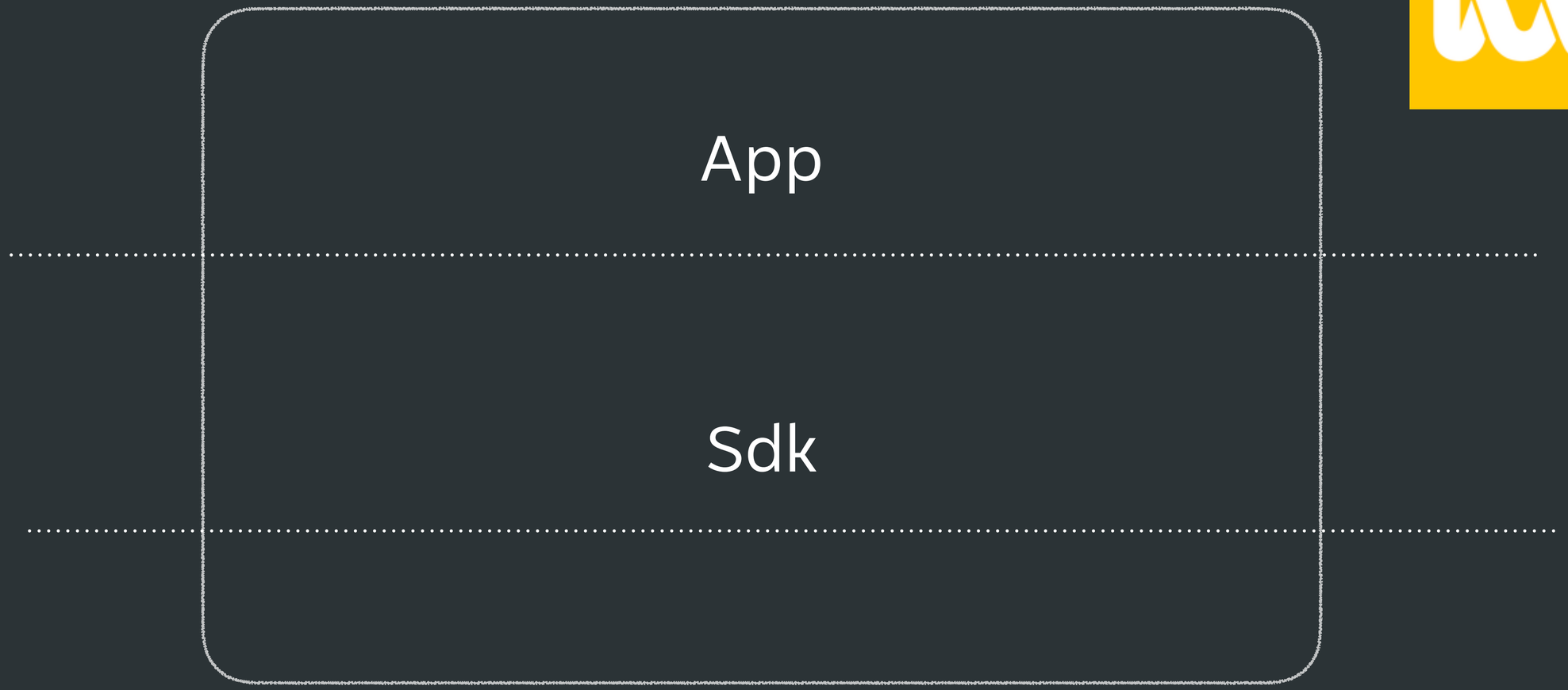
```
func apply(style textStyle: TextStyle,
           to text: String) -> NSAttributedString
```

Background Domain Dictionary App SDK

```
func apply(style textStyle: TextStyle,
           to text: String) -> NSAttributedString
```

```
func apply(style textStyle: String,
           to text: String) -> NSAttributedString
```

Background Domain Dictionary App SDK

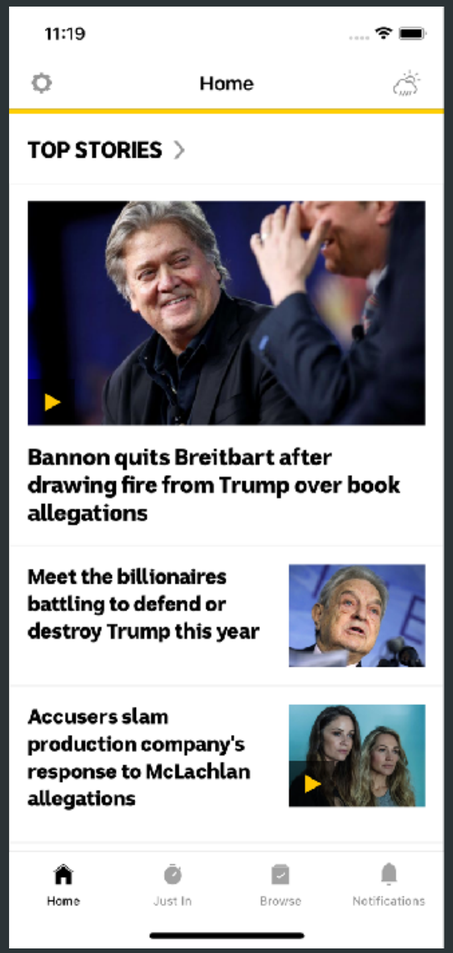

App
Sdk

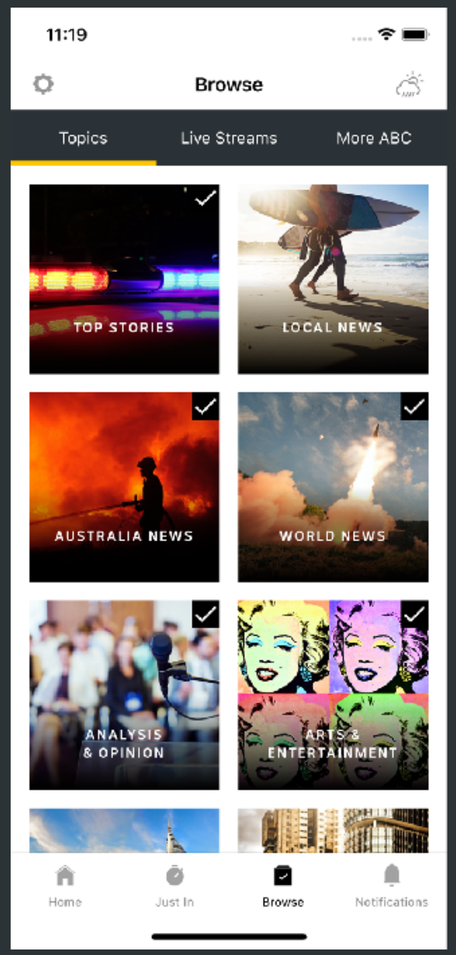
Concrete

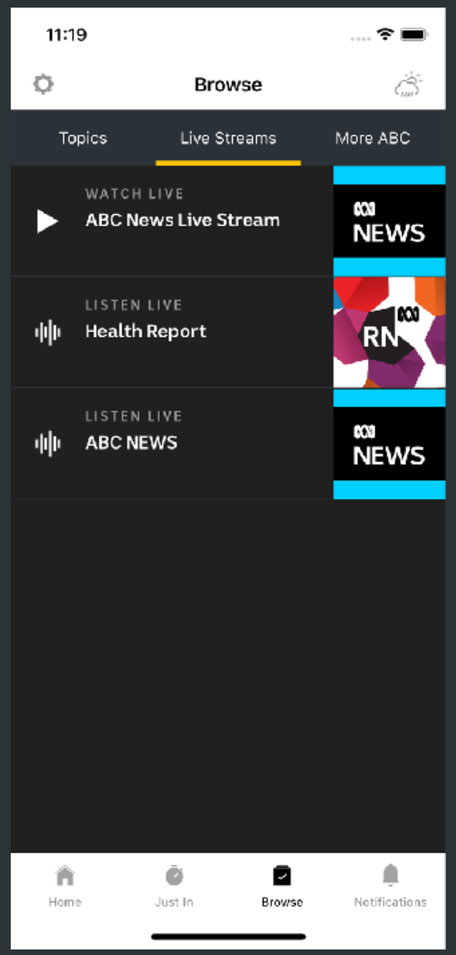
```
struct SDK {
    let apiKey: String

    func getTopics() -> [Topics]
{ …


struct Topic {
    let id: String
    …
    func getArticles(with sdk: SDK) -> [Article] { …
```







Background

Domain Dictionary

App

SDK

Load Requirement

```
protocol Loadable {
    associatedtype T
    func load(_ : @escaping callback<T> )
}
```

Load Requirement

```
struct TopicsLoader: Loadable {
    typealias  T = [Topic]

    func load(_ callback: @escaping callback<[Topic]>) {
        …
```

Concrete

```
struct SDK {
    let apiKey: String

    func getTopics() -> [Topic] {
…


struct Topic {
    let id: String
    …
    func getArticles(with sdk: SDK) -> [Article] { …
```

Background Domain Dictionary App SDK

Load Requirement

```
protocol Loadable {
    associatedtype T
    func load(_ : @escaping callback<T> )
}
```

```
struct Model<T: Loadable> {
    let loader: T
. . .
```

```
class ViewController<T: Loadable> {
    let model: Model<T>
. . .
```

Background Domain Dictionary App SDK

```
final class TopicList: ViewController<ShowsLoader> {
    . . .
    override func viewDidLoad() {
        self.model.loader.load() { result in
            switch result {
            case .success(let topics):
                print("\(topics)")
            case .failure:
                print(" failed :( ")
            }
        }
    }
    . . .
```

Background Domain Dictionary App SDK

Types

Group the values

Guarantees the flow

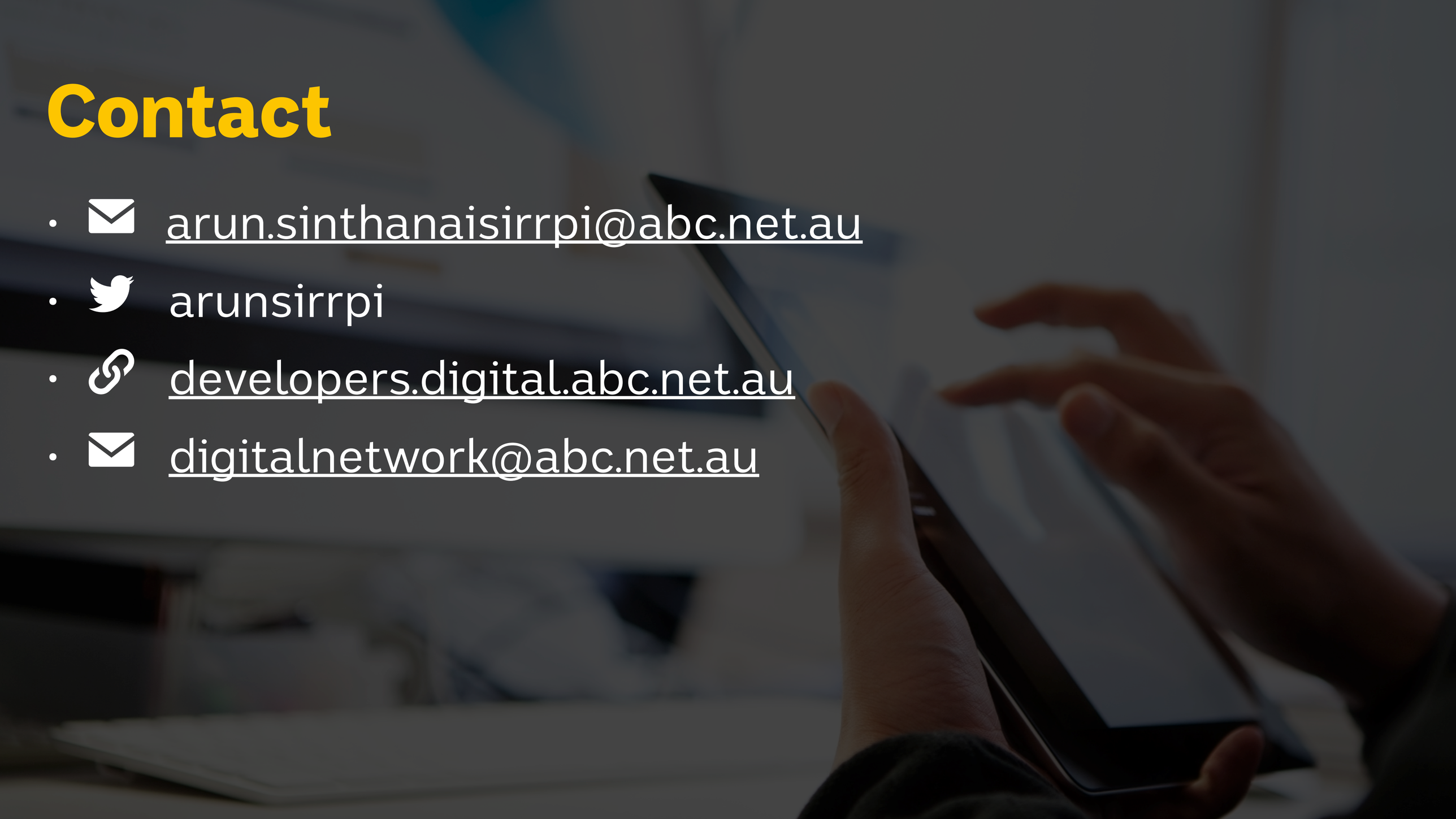
Types

- Teach the compiler our domain logic
- Catch bugs at compile time
- Reduce the number of tests we write
- Produce a living documentation

0.0007%

Summary

- Domain Dictionary
- Types

Contact

- arun.sinthanaisirrpi@abc.net.au
- arunsirrpi
- developers.digital.abc.net.au
- digitalnetwork@abc.net.au