

Android development tips

Writing clean code, improving productivity and keeping up-to-date



Lyubomir Ganev

About



2010
Saarland
University

2012
1st Position

2013
Chip
online

2014
Spontacts

2014
Jochen
Schweizer



Private
projects

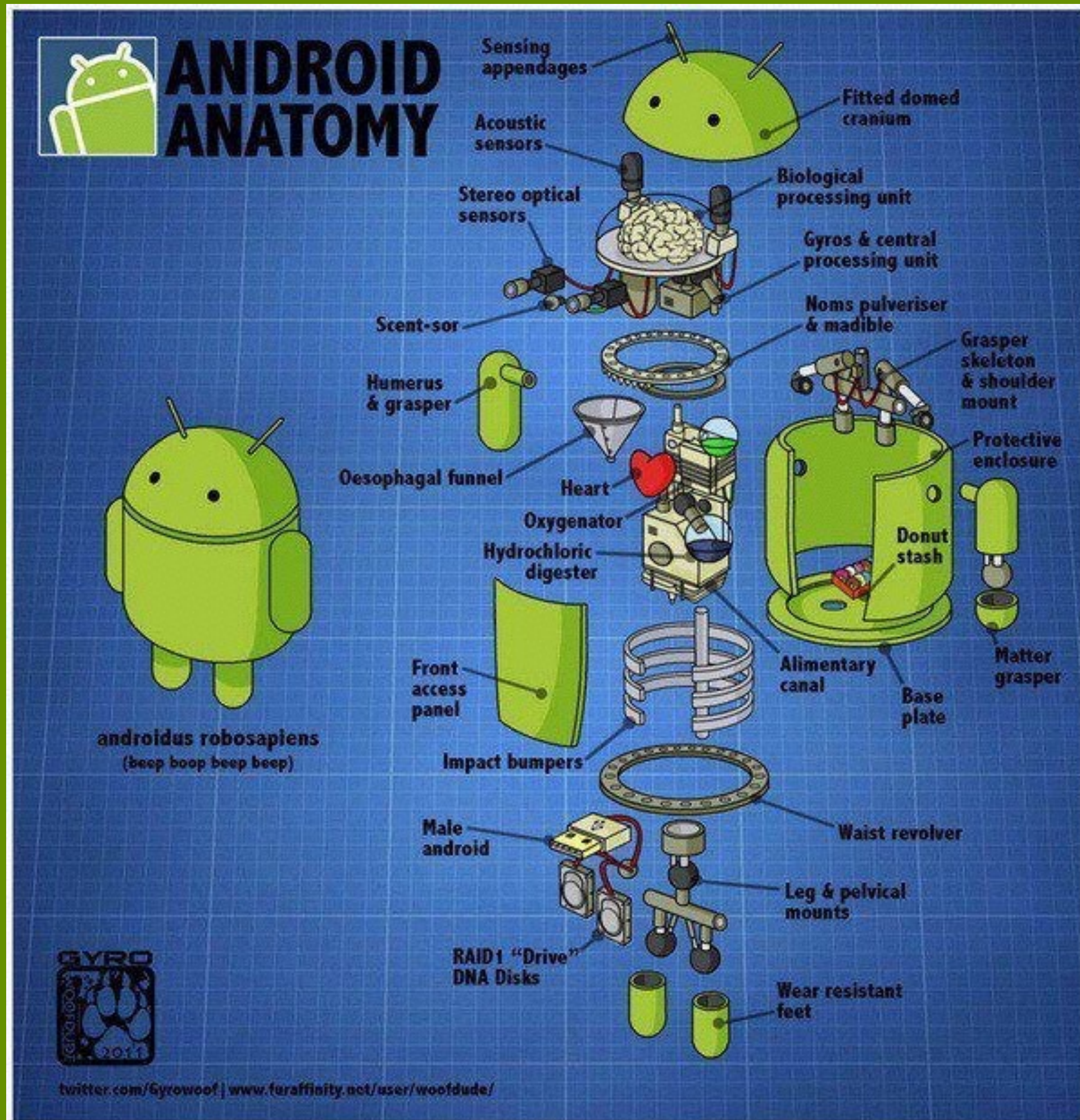


Agenda

- write clean code
 - complexity, S.O.L.I.D. architectures, testing
- improve productivity
 - open-source, IDE tips and tricks
- keep up-to-date

Android apps are complex

The SDK is not exactly simple



- Activities, Services, Broadcast Receivers, Content Providers, Bundles, Intents
- Fragments, Views, Notifications, Resources
- Databases, IPC, Threads, Storage
- Thousands of APIs related to services and specific hardware features of devices

Where to put our code?

- Activity does a lot
 - lifecycle, state save and restore (Bundles, Parcelables)
 - multiple entry points (Intents, Share, Notifications deep links)
 - non-trivial navigation (back, up, backstack, processes, Intent flags)
 - Views and Fragments inside and their communication (interfaces, callbacks, fragment transactions, inflation)
 - transitions, shared hero elements, alternative layouts and resources for devices, orientation etc.
 - Loaders, Dynamic permissions, Activity results, and more

App architecture

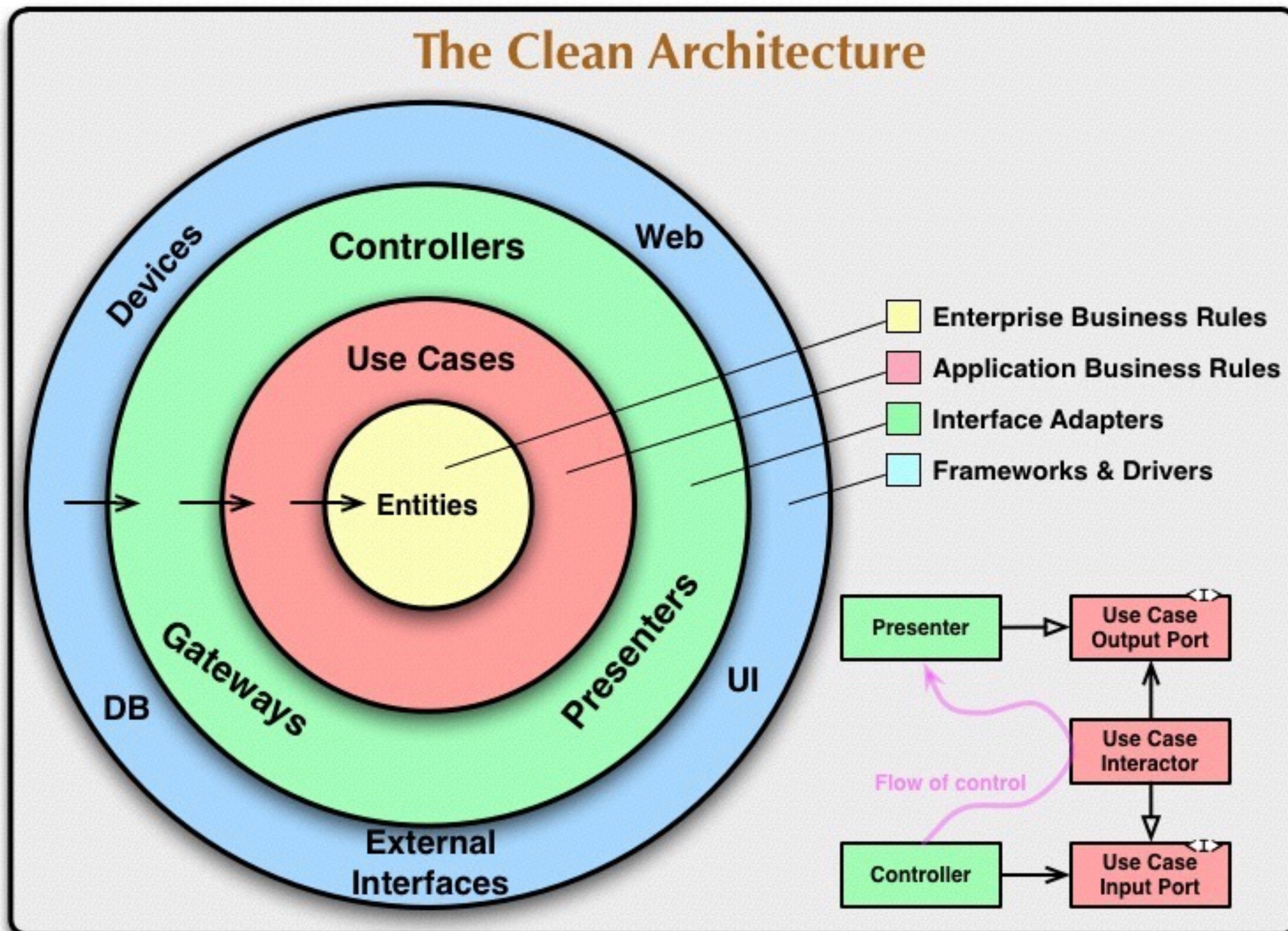
Android app architecture

- Goals
 - modular, flexible, extensible, testable
 - make or life easier, improve team work
- Current state
 - a hot topic now, so there are “n” different solutions
 - no silver bullet for all cases

Android MVP/MVC

- Model-View-Presenter/Controller
- View is the UI - can be Activity, Fragment or View
- Models contain the business logic and handle data
- Presenters/controllers do the magic between View and Model
- Blogpost on MVP in Android
<http://goo.gl/84NmoJ>

Uncle Bob's CLEAN architecture



- Multi-layer architecture

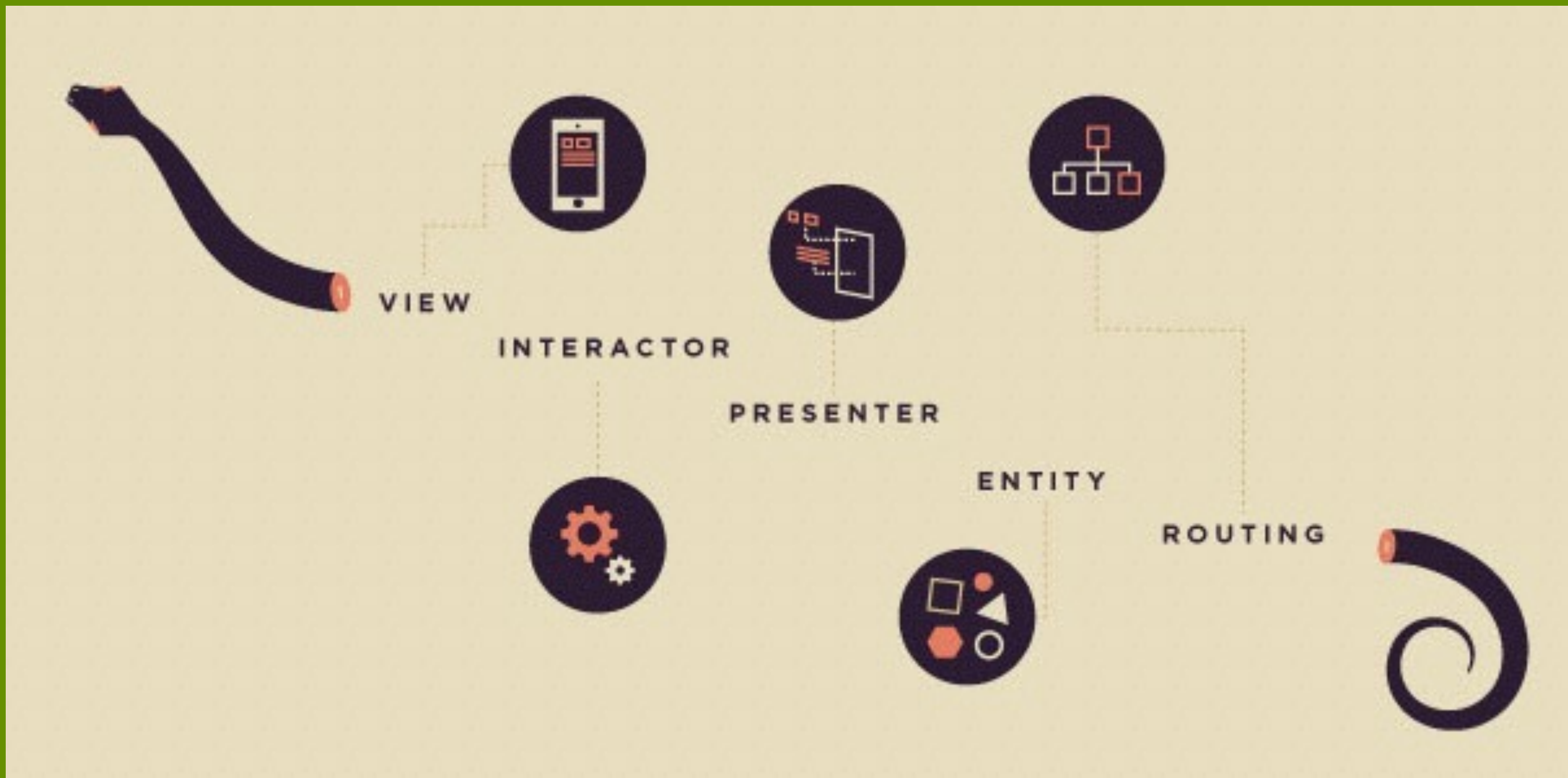
Link: <http://goo.gl/8TvyXR>

- The Dependency Rule - Nothing in an inner circle can know anything at all about something in an outer circle.

- Android implementation
Fernando Cejas blogpost

Link: goo.gl/dL582y

VIPER architecture



- Introduction
Link: goo.gl/Y8v0o3
- In detail article (iOS)
Link: goo.gl/xaxrNC
- Android clean architecture blogposts
Link: <http://goo.gl/AU1ccK>

S.O.L.I.D. principles in OOP

- Wikipedia: <https://goo.gl/A7aSN8>
- **Single responsibility** - A class should have only one reason to change.
- **Open/closed** - Allow a class' behaviour to be extended without modifying its source code.
- **Liskov substitution** - Classes should be replaceable with instances of their subtypes without altering the correctness the program.
- **Interface segregation** - Many client-specific interfaces are better than one general-purpose interface. No client should be forced to depend on methods it does not use.
- **Dependency inversion** - One should depend upon abstractions, not upon concretions.

Testing

Testing on Android is now easy

- Just do it
- TDD when you can
- Unit tests and Instrumented tests
- Android Studio and Gradle have great support for testing
- UI test recorder coming soon to Android Studio

Some testing resources

- Android testing codelab - setup your testing environment
<https://goo.gl/MsSiov>
- Espresso - easy to use UI testing framework
<https://goo.gl/OvsJ0l>
- Android testing patterns - YouTube videos
<https://goo.gl/QQa89m>

Productivity

Open-source libraries

- You do not have to re-invent the wheel
- Many 3rd party libraries are better tested and supported than your custom implementation
- Android open-source rockstars
<https://github.com/square> <https://github.com/JakeWharton>
- Android arsenal - a good collection of open-source libraries
<https://android-arsenal.com/>
- Don't use too many and keep an eye on their size (65K dex limit)
<https://goo.gl/Uxsvv8>

Know your IDE

- Android Studio is extremely powerful (based on IntelliJ)
- Becoming power user is a worthwhile investment
- My favourite Android Studio tips and tricks



Powered by IntelliJ Platform

Even more tips and tricks

- The experts' guide to Android development tools (Google IO 2016)
<https://www.youtube.com/watch?v=hHnTIMjd1Y8>
- Android Studio for Experts (Android Dev Summit 2015)
<https://www.youtube.com/watch?v=Y2GC6P5hPeA>
- (About) 10 things you (probably) didn't know
<https://www.youtube.com/watch?v=eOV2owswDkE>
- What's new in android developer tools Google I/O (2013-2016)
<https://goo.gl/Mkrnw3>

Staying up-to-date

Recommended resources

- Android developer weekly newsletters

<http://androidweekly.net/> <https://www.androiddevdigest.com/>

- Podcasts

<http://fragmentedpodcast.com/> <http://androidbackstage.blogspot.de/>

- Android developers YouTube channel

<https://www.youtube.com/user/androiddevelopers>

- Android developers blog

<http://android-developers.blogspot.de/>

Questions?

twitter.com/ lgancho

google.com/+ LyubomirGanev

website & blog luboganev.github.io