

COS 425 - MOBILE DEVELOPMENT

MOBILE DEVICES

Stephen Houser <houser@maine.edu>

MOBILE DEVICES

- ▶ History & Evolution
- ▶ Design Philosophy
- ▶ Mobile Architecture



MOBILE DEVICE EVOLUTION

Evolution of the Mobile Phone



MOBILE DEVICE EVOLUTION



iPhone 12

5.4" OLED Super f
HD
Aluminum bo
Dual lens w/o
A14 Chip
4GB RAM
5G capability

Price

128GB - \$649
256GB - \$749

vs



iPhone 12 Pro

6.1" OLED Super Retina
XDR
Stainless steel
Triple lens + LiDAR
120Hz ProMotion
A14 Chip
6GB RAM
5G capability

Price

128GB - \$999
256GB - \$1099
512GB - \$1299

GOOGLE PIXEL SERIES

4,500mAh -
30W Fast Charge

Android 12 -

128/256GB ROM -
6/8GB RAM

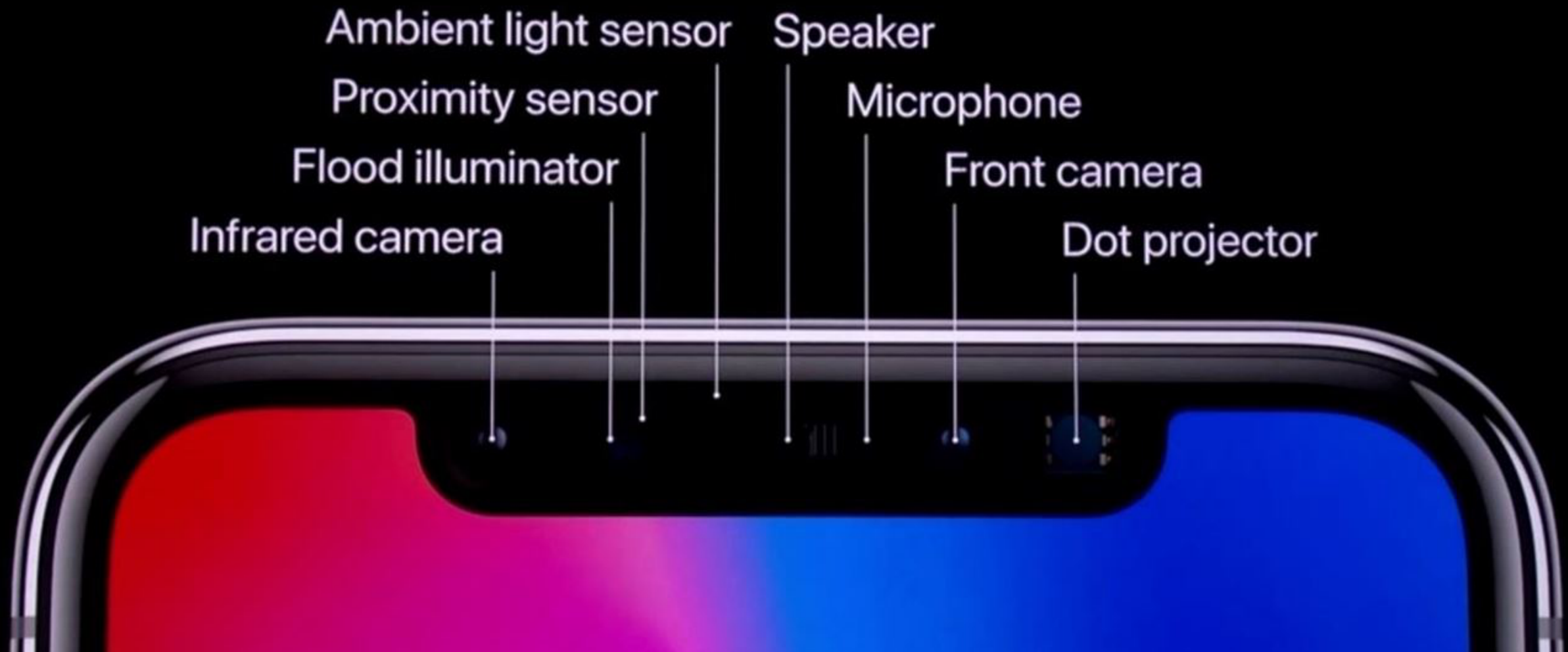


- 6.40" FHD+ AMOLED
120Hz Refresh rate

- Whitechapel Chip

- 50+8+8MP (Back)
16MP (Front)

MOBILE DEVICE EVOLUTION



MOBILE DEVICE EVOLUTION

LG G5 specs overview



4-core, 2.2 GHz

CPU Qualcomm Snapdragon 820

GPU Adreno 530

RAM 4 GB RAM

STORAGE 32 GB



2800 mAh



LTE

HSPA, HSUPA, UMTS,
EDGE, GPRS
NFC, Wi-Fi 802.11, Bluetooth 4.2



16 megapixels

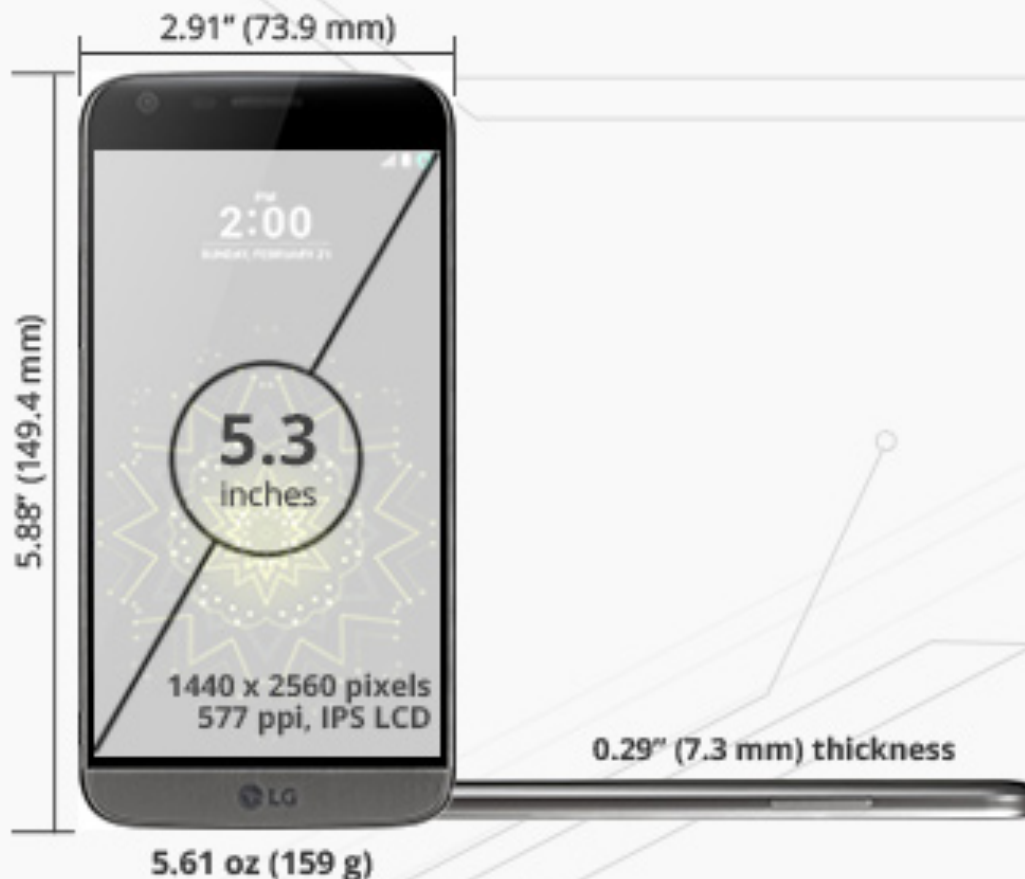
REAR F1.8 aperture, sensor 1/2.6"

Front: 8 Megapixels

Video capture: 3840x2160 (4K)



Android (6.0)



MOBILE DEVICE EVOLUTION

A large teal square with a textured, paper-like appearance. In the center, the text "LG G5" is written in a white, bold, sans-serif font. The "G" is stylized with a thick, curved top bar.

LG G5

MOBILE DEVICE EVOLUTION

- ▶ Better batteries - longer lasting
- ▶ Smaller components, displays
- ▶ More powerful processors and graphics
- ▶ Cameras!

APPLICATIONS

- ▶ Who's going to write all these applications?
 - ▶ Wireless Application Protocol (WAP)
 - ▶ Effectively stripped down HTML
 - ▶ Java Applications as well
- ▶ Smart (programmable) Phones
 - ▶ Microsoft, Apple, Google

PLATFORMS

- ▶ Apple iOS
 - ▶ iPod + Phone + Web Browser
 - ▶ First widely used “app store”
- ▶ Google Android
 - ▶ Operating System, not hardware
 - ▶ Open Source, no developer fees, alternate “stores”
- ▶ Microsoft
 - ▶ “Windows Everywhere”, never really caught on.

APPLE IOS DEVELOPMENT

- ▶ iPhones and iPads, rumors of inclusion in macOS
- ▶ Objective-C or Swift for native development
- ▶ Requires macOS to bundle for distribution*
- ▶ Tight security controls on app distribution
- ▶ Subject to Apple's approval process
- ▶ Here lies the most money ([App Annie](#) 2015)

ANDROID DEVELOPMENT

- ▶ Thousands of devices from multiple manufacturers
 - ▶ varying degrees of “upgradeability”
- ▶ Java using Android Studio or native ARM (C/C++)
- ▶ Open Source
- ▶ Google Play store is \$25 one-time fee
- ▶ Alternate “stores” and alternate distribution channels

HYBRID DEVELOPMENT

- ▶ Apache Cordova (open source)
 - ▶ HTML, CSS, & JS
- ▶ Xamarin
 - ▶ C#
 - ▶ Part of Visual Studio (Microsoft)
- ▶ React Native (open source, facebook)
 - ▶ HTML, JS, & Native UI elements

DESIGN PHILOSOPHY

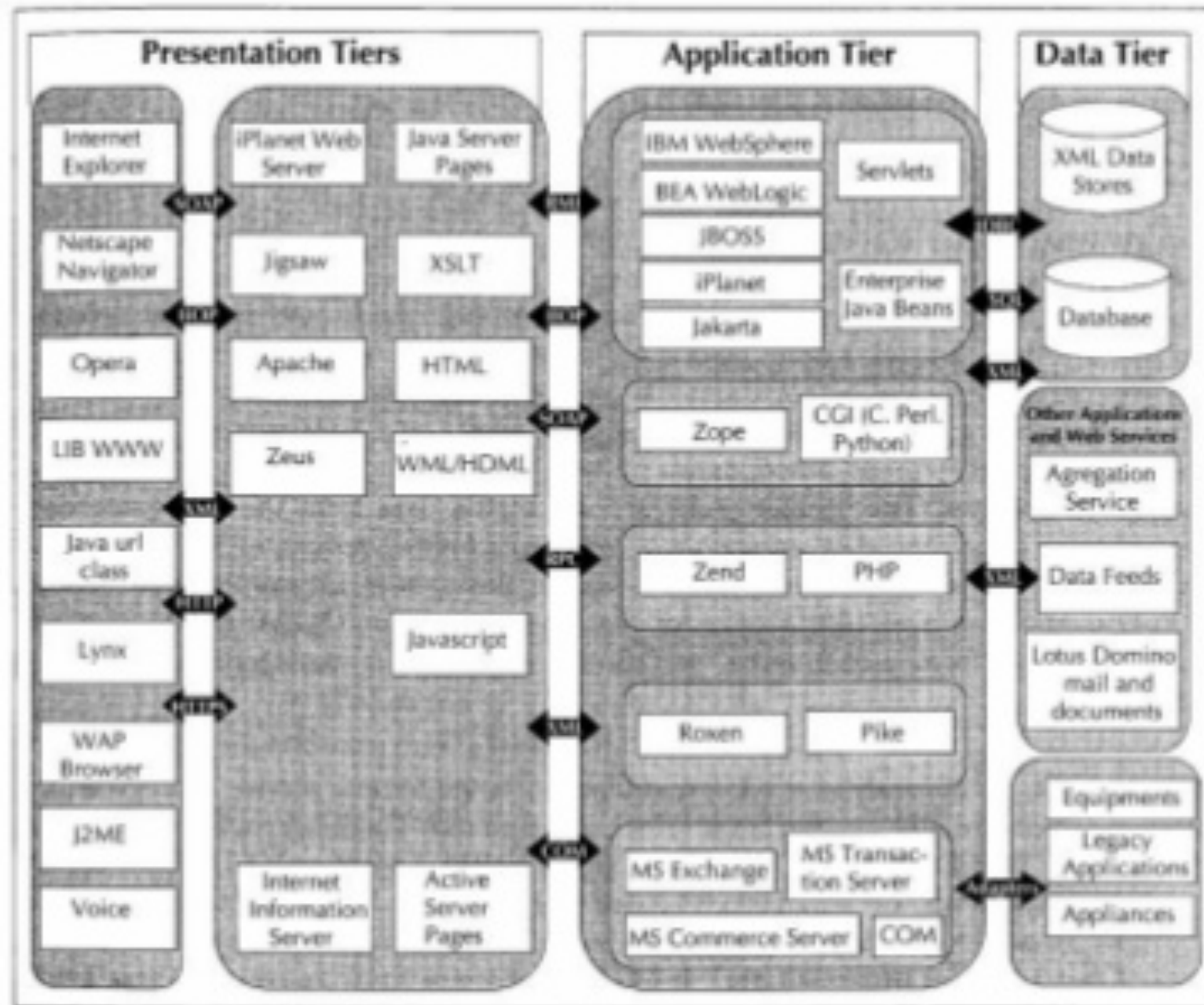
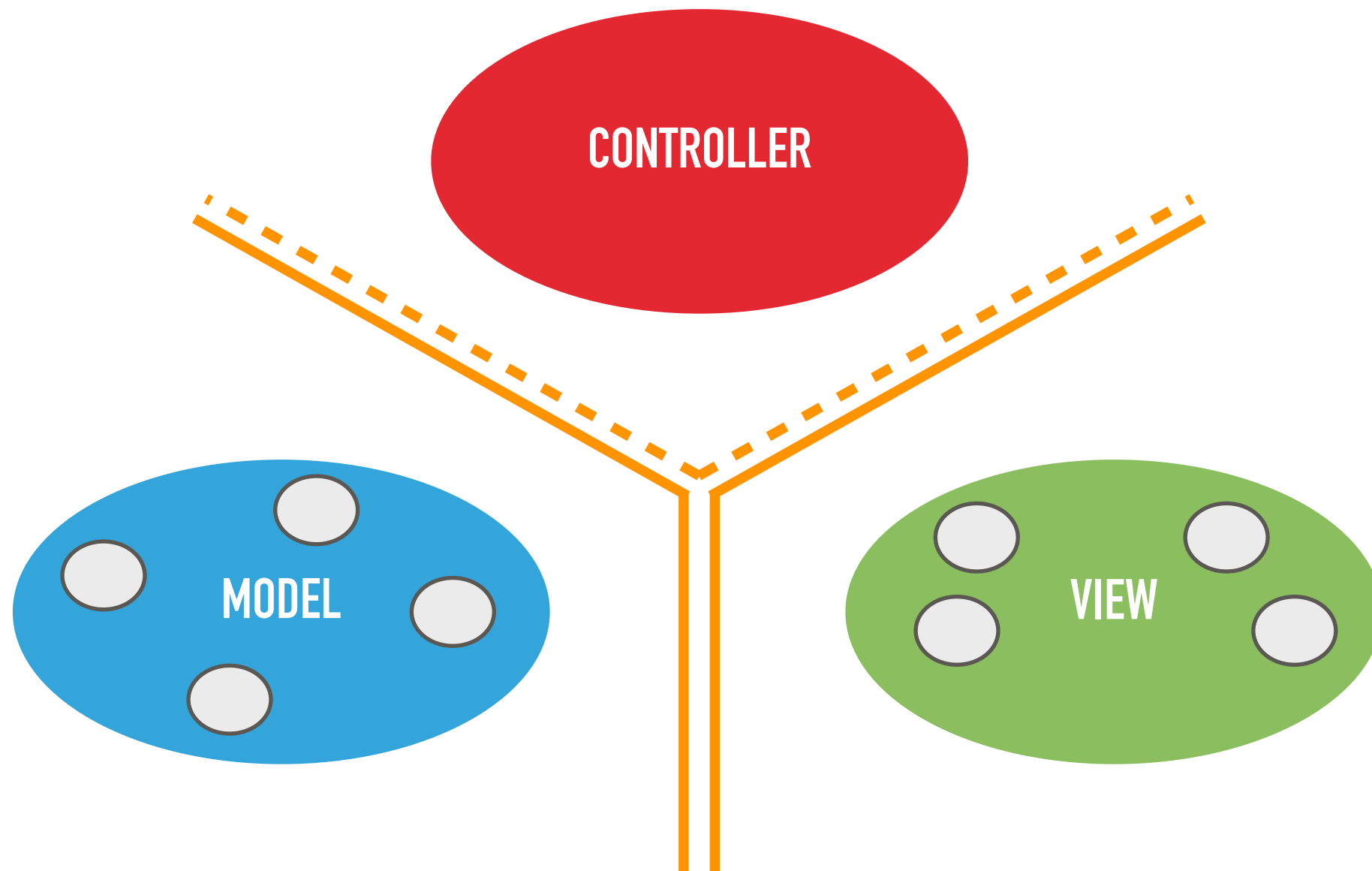


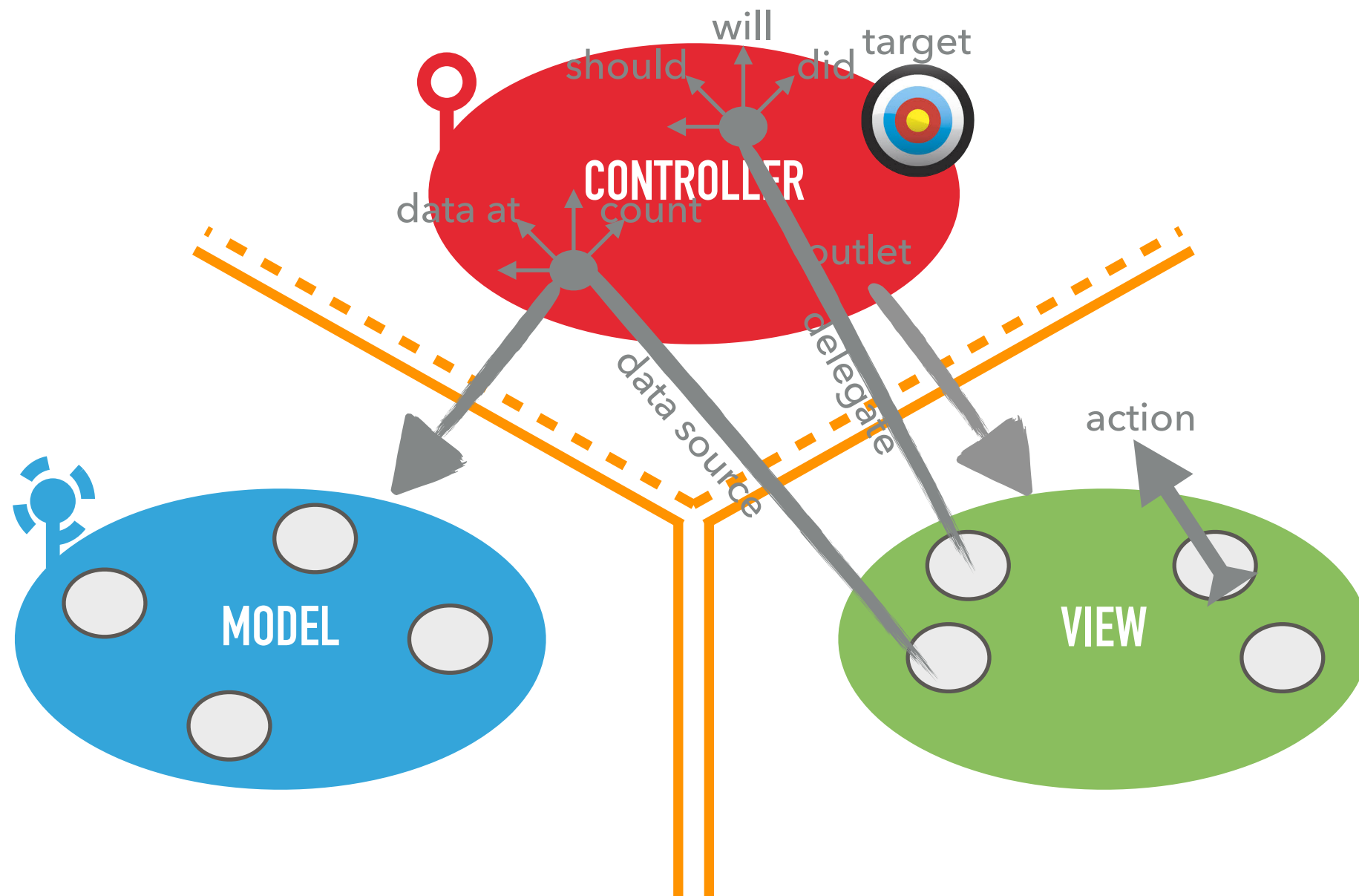
Figure 2.2 The mobile computing architecture

- ▶ Both follow 3-tier
- ▶ Presentation (View)
- ▶ Logic/Application (Controller)
- ▶ Data (Model)

DESIGN PHILOSOPHY



DESIGN PHILOSOPHY



3-TIER ARCHITECTURE

- ▶ 3-Tier applies to many design scenarios
 - ▶ Keep presentation lightweight, independent of rest
 - ▶ Keep logic separate and encapsulated, easy to modify
 - ▶ Keep data separate. NEVER build based on current values
- ▶ How does it apply to websites and desktop applications?

3-TIER ARCHITECTURE

- ▶ Controller
 - ▶ Role is a “traffic cop”
 - ▶ Takes requests from user and turns into method calls
 - ▶ Finds the right model to use
 - ▶ Finds the right view to use
 - ▶ Returns all that back to the user

3-TIER ARCHITECTURE

- ▶ Model
 - ▶ Representation of the data
 - ▶ May be linked to a database or remote “server”
 - ▶ Can be much more complicated than that!
 - ▶ Contains relationship rules (1-many, many-many)

3-TIER ARCHITECTURE

- ▶ View

- ▶ Template to be populated with data from the model
- ▶ All UI components go here
- ▶ May have internal state data (non-persistent)

CLIENT SERVER

- ▶ Client/Server Architecture
 - ▶ Connect local application (views) with large backend data and services
 - ▶ How much should be done...
 - ▶ Locally (on device)?
 - ▶ Remote (in cloud)?

CLIENT SERVER

- ▶ Thick Client
 - ▶ business logic and some data on device.
 - ▶ “Off the grid” capable
- ▶ Thin Client
 - ▶ Business logic and data in the cloud
 - ▶ Must be connected
- ▶ Which is better? When?

MOBILE DEVICES

- ▶ Thick Client
 - ▶ business logic and some data on device.
 - ▶ “Off the grid” capable
- ▶ Thin Client
 - ▶ Business logic and data in the cloud
 - ▶ Must be connected
- ▶ Which is better? When?

ARCHITECTURE

- ✓ History & Evolution
- ✓ Design Philosophy
- ✓ Mobile Architecture