

# COEN 268: Mobile App Development

Instructor: Farokh Eskafi

Email: [feskafi@scu.edu](mailto:feskafi@scu.edu)



Santa Clara University

Center for Science, Technology, and Society

# Course Description

- This course introduces mobile application development. Focused on Android – for practical reasons!
- Students will learn how to think about mobile app and how to develop and deploy Android applications
- Particular emphasis on software engineering topics including software architecture, software process, usability and deployment.
- Topics such as Activities, Views, Intents, Broadcast Receivers, Content Providers, Location-based services, Sensors
- You will learn to develop your own apps



# Class Background

- Background?
- How many years of Java?
- Any app development experience?
- What mobile app do you think is the most impactful/ innovative?



# Course Progress

Week 1:	Material design for mobile applications. Student reading and discussions
Week 2	Introduction to mobile application development. Activity
Week 3:	Android Application Basics: Logging, Manifest. Lab1: Activity
Week 4:	Android Application Basics: resources, Lab 2: Layout
Week 5:	Android Application Basics: resources, dialogs, service, Quiz 1
Week 6:	Service continued, Lab 3: service
Week 7:	Fragments, maps, SQLite, Quiz 2
Week 8:	Content provider, threads and asynchronous processing, Lab 4: SQLite
Week 9:	Final Project discussions and preparations, Quiz 3
Week 10:	Project presentations

# Grading

- Quizzes 30 pts
- Class presentation / assignments 20 pts
- Projects 40 pts
- Attendance and lab sessions 10 pts
- Grading is based on class average
  - A (93+), A- (89+,93)
  - B+ (85-89), B (81,85), B- (75-81)
  - C's (62 to 75)

# Student Presentations

- Specific assignments
- A student team will present and lead the topic
- If possible, whole class will participate
- Demo app



# Overview

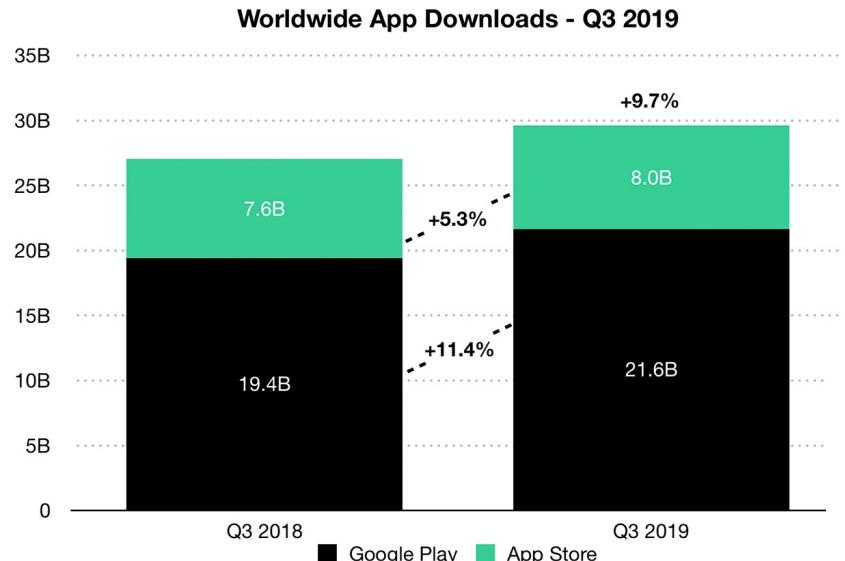
## Mobile Applications Market and Penetration



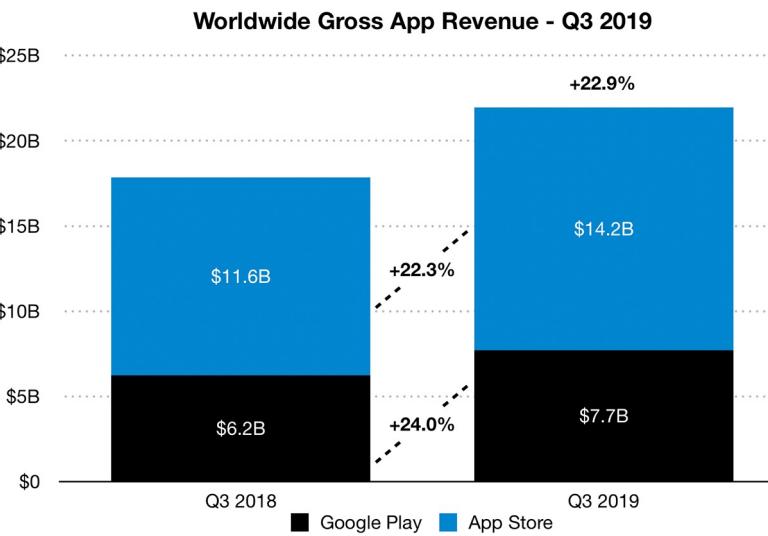
**Santa Clara University**

Center for Science, Technology, and Society

# Mobile Apps Annual Gross Revenue



SensorTower



SensorTower



Santa Clara University

Center for Science, Technology, and Society



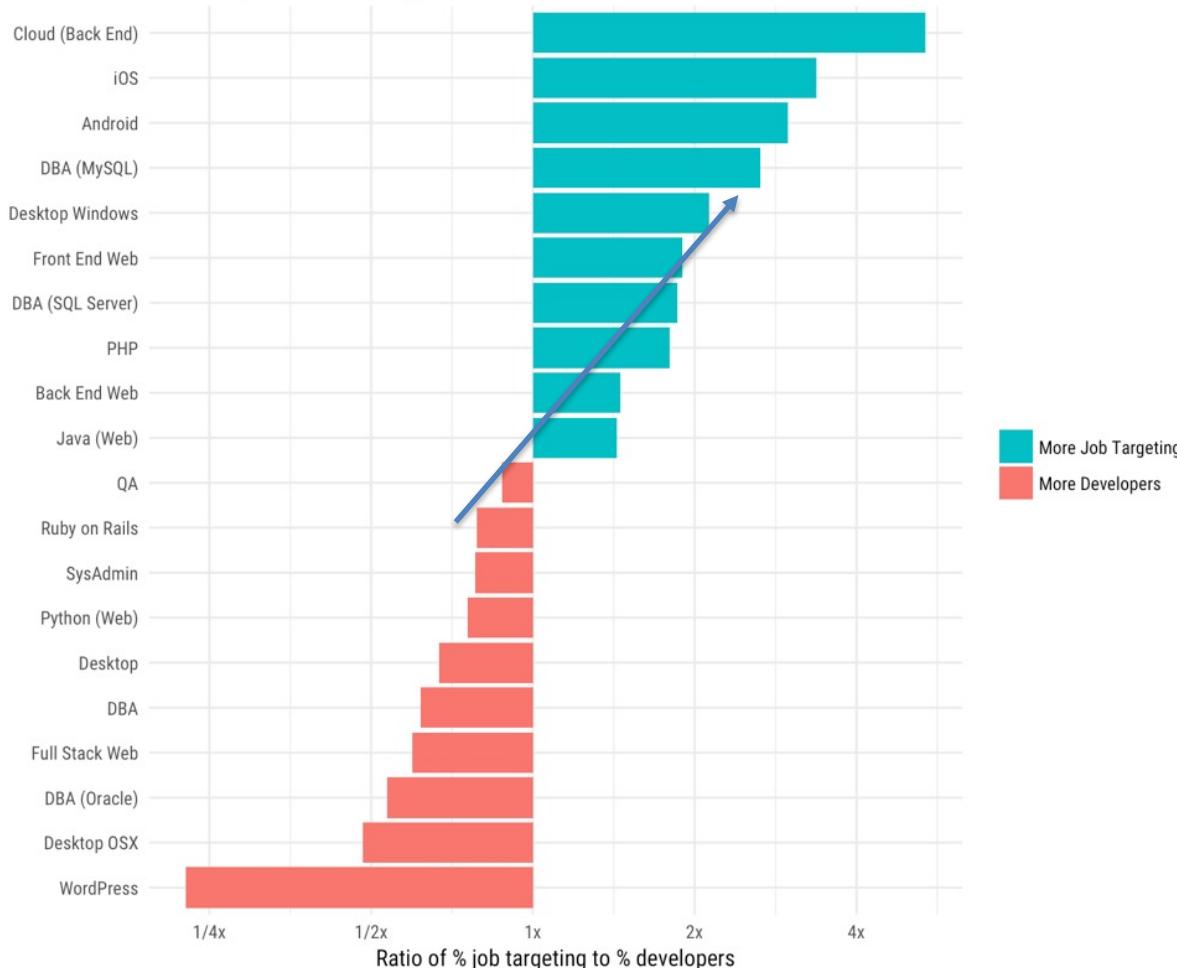
SensorTower Data That Drives App Growth

[sensortower.com](http://sensortower.com)

# Job Opportunities

## What job targeting is in high demand?

There is high demand for developers who work in backend web/cloud, iOS, Android, and DBA/SQL



# Mobile App Dev Process

Why? What? How?



Santa Clara University

Center for Science, Technology, and Society

# Why?

- More than a third of world population are on smart phones!
- Most use phones as the only means of communications and entertainment (and “fake” news)
- Billions \$ been made and to be made
- Preferred method of organizing and protesting



# Why Mobile App

- Changing business process
  - automation vs reengineering
  - when you can be there and control the situation using an app!
- use device capability to innovate
  - location services



# Why?

- what is so special about mobile devices:
  - capture the environment (location, direction, speed, temp, ...)
  - communicate
  - computer in the hand
- accessibility
- branding at the micro level
- changing business processes
  - automating existing processes (questionable)
  - information at your fingertip!



Santa Clara University

Center for Science, Technology, and Society

# What?

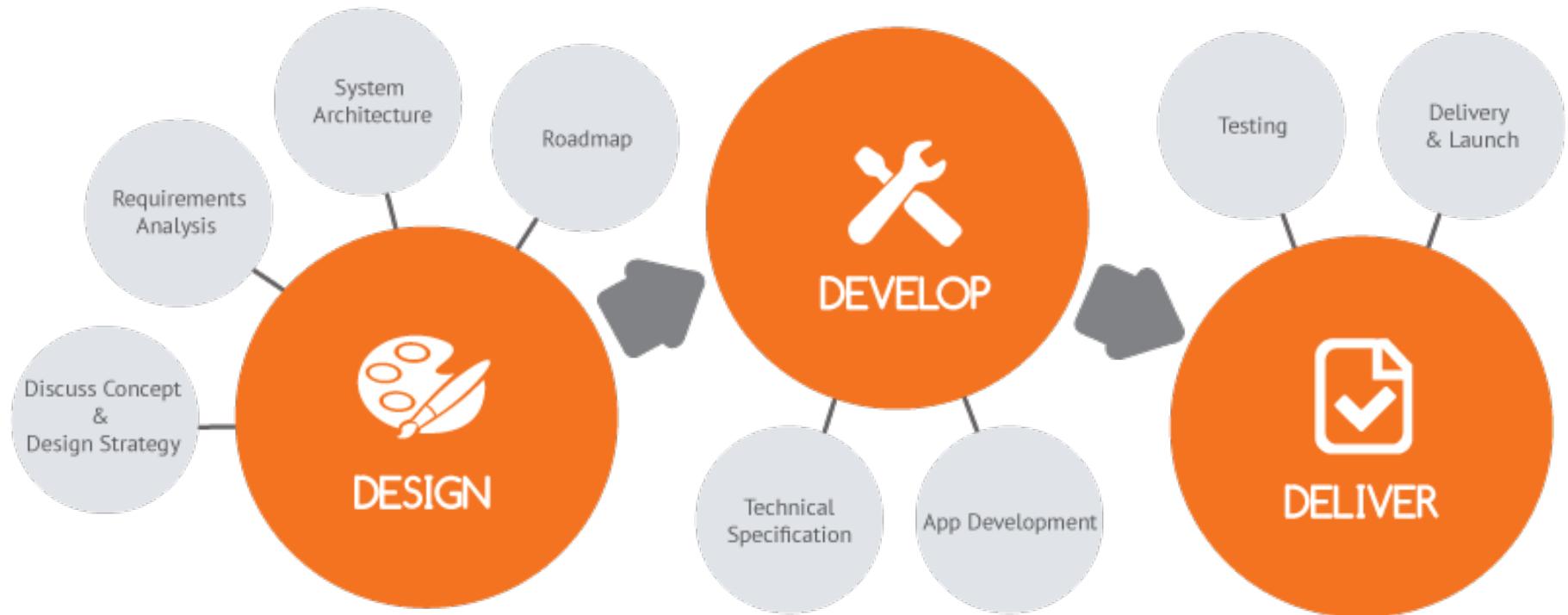
- Education
- Social
- Games
- Streaming -> entertainment
- Entertainment
- Health
- fitness
- Fintech
- E-commerce, B2B, B2C, C2C
- Ride sharing
- Maps, services
- Dating
- Messaging app
- Browser (?)
- Recommendations
- Personalization
- Sports
- Home automation
- Alarm
- Productivity
- News
- Enterprise -> ERP, CRM , B2B

# How?

- Requirements
- Design
- Development
- Test
- Support



# Mobile Application Development



**Santa Clara University**

Center for Science, Technology, and Society

# Mobile App Design (UI)

- User interface design
  - Storyboard: everything including the images to be shown, icons appearing, theme, color, background, etc. is a part of storyboard
    - Process Flow: the Process Flow chart provides a visual representation of the steps in a process. Flow charts are also referred to as *ProcessMapping* or Flow Diagrams.
  - Wireframe: Elements visible on each of the screen of the mobile application are discussed and noted down as wireframes



# Mobile App Design (Software)

- Architecture design
  - Planning and development of application architecture
  - Coding structure
- Test planning
  - Manual
  - Automated



Santa Clara University

Center for Science, Technology, and Society

# Development (1)

- Prototyping
  - proof-of-concept phase
  - core functionality, or specific parts of the application are working
  - Major bugs are present
- Examples



# Development (2)

- Alpha
  - Core functionality is generally code-complete (built, but not fully tested)
  - Major bugs are still present
  - Outlying functionality may still not be present
- Beta
  - Most functionality is now complete
  - System testing and bug fixing
  - Major issues cannot be present
- Release
  - Release to the wild!

# Mobile Dev Testing

- Feature testing
- App testing
- Testing on individual devices!
  - How many devices are there?
- Testing on different OS versions!
  - Android
    - We are on Android 10
  - iOS
    - We are on iOS 13.4

# Support and Maintenance

- Customer support
  - Installation
  - Training and help
  - Bug reporting
- Backward compatibility
  - Decide and plan for backward compatibility



# Mobile OS



**Santa Clara University**

Center for Science, Technology, and Society

# iOS

- Software platform for all Apple's mobile hardware
- Built as a version of Macintosh OS (shrunk version), first version came out in 2007
- it is a proprietary software, tied to the hardware
- Check here for iOS versions:  
<https://www.gkgigs.com/list-apple-ios-version-history/>



Santa Clara University

Center for Science, Technology, and Society

# Android

- A Software Platform and Operating System for Mobiles
- It's based on Linux Kernel
- Android was found in 2003
- Google took over Android in 2005 for \$50m
- It's an open, complete, and free mobile platform
- Software stack of Android is open sourced and licensed under Apache 2.0
- check  
<https://www.computerworld.com/article/3235946/android-versions-a-living-history-from-1-0-to-today.html>



# Android Versions

Code name	Version number	Linux kernel version <sup>[1]</sup>	Initial release date	API level	Ref
(No codename)	1.0	?	September 23, 2008	1	<a href="#">[2]</a>
Petit Four	1.1	2.6	February 9, 2009	2	<a href="#">[2]</a>
Cupcake	1.5	2.6.27	April 27, 2009	3	
Donut	1.6	2.6.29	September 15, 2009	4	<a href="#">[3]</a>
Eclair	2.0 – 2.1	2.6.29	October 26, 2009	5 – 7	<a href="#">[4]</a>
Froyo	2.2 – 2.2.3	2.6.32	May 20, 2010	8	<a href="#">[5]</a>
Gingerbread	2.3 – 2.3.7	2.6.35	December 6, 2010	9 – 10	<a href="#">[6]</a>
Honeycomb	3.0 – 3.2.6	2.6.36	February 22, 2011	11 – 13	<a href="#">[7]</a>
Ice Cream Sandwich	4.0 – 4.0.4	3.0.1	October 18, 2011	14 – 15	<a href="#">[8]</a>
Jelly Bean	4.1 – 4.3.1	3.0.31 to 3.4.39	July 9, 2012	16 – 18	<a href="#">[9]</a>
KitKat	4.4 – 4.4.4	3.10	October 31, 2013	19 – 20	<a href="#">[10]</a>
Lollipop	5.0 – 5.1.1	3.16	November 12, 2014	21 – 22	<a href="#">[11]</a>
Marshmallow	6.0 – 6.0.1	3.18	October 5, 2015	23	<a href="#">[12]</a>
Nougat	7.0 – 7.1.2	4.4	August 22, 2016	24 – 25	<a href="#">[13]</a>
Oreo	8.0 – 8.1	4.10	August 21, 2017	26 – 27	<a href="#">[14]</a>
Pie	9.0	4.4.107, 4.9.84, and 4.14.42	August 6, 2018	28	<a href="#">[15]</a>
Android Q	10.0			29	

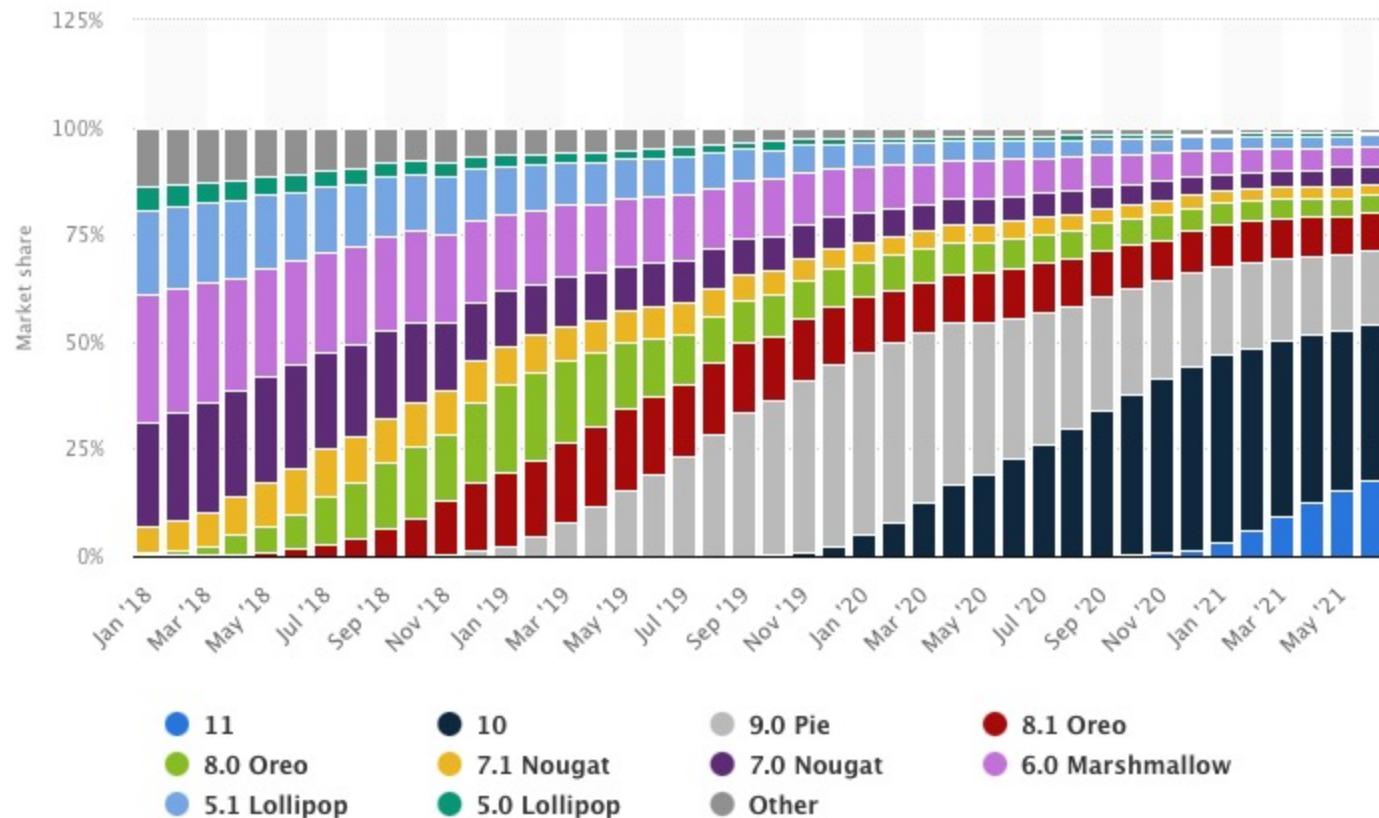
Legend:  Old version  Older version, still supported  Latest version  Latest preview version



Santa Clara University

Center for Science, Technology, and Society

# Android Penetration



© Statista 2021

[Additional Information](#)

Show source

# iOS Penetration

- iOS 14: 90%
- iOS 13: 8%
- earlier: 2%



# Mobile Platforms



**Santa Clara University**

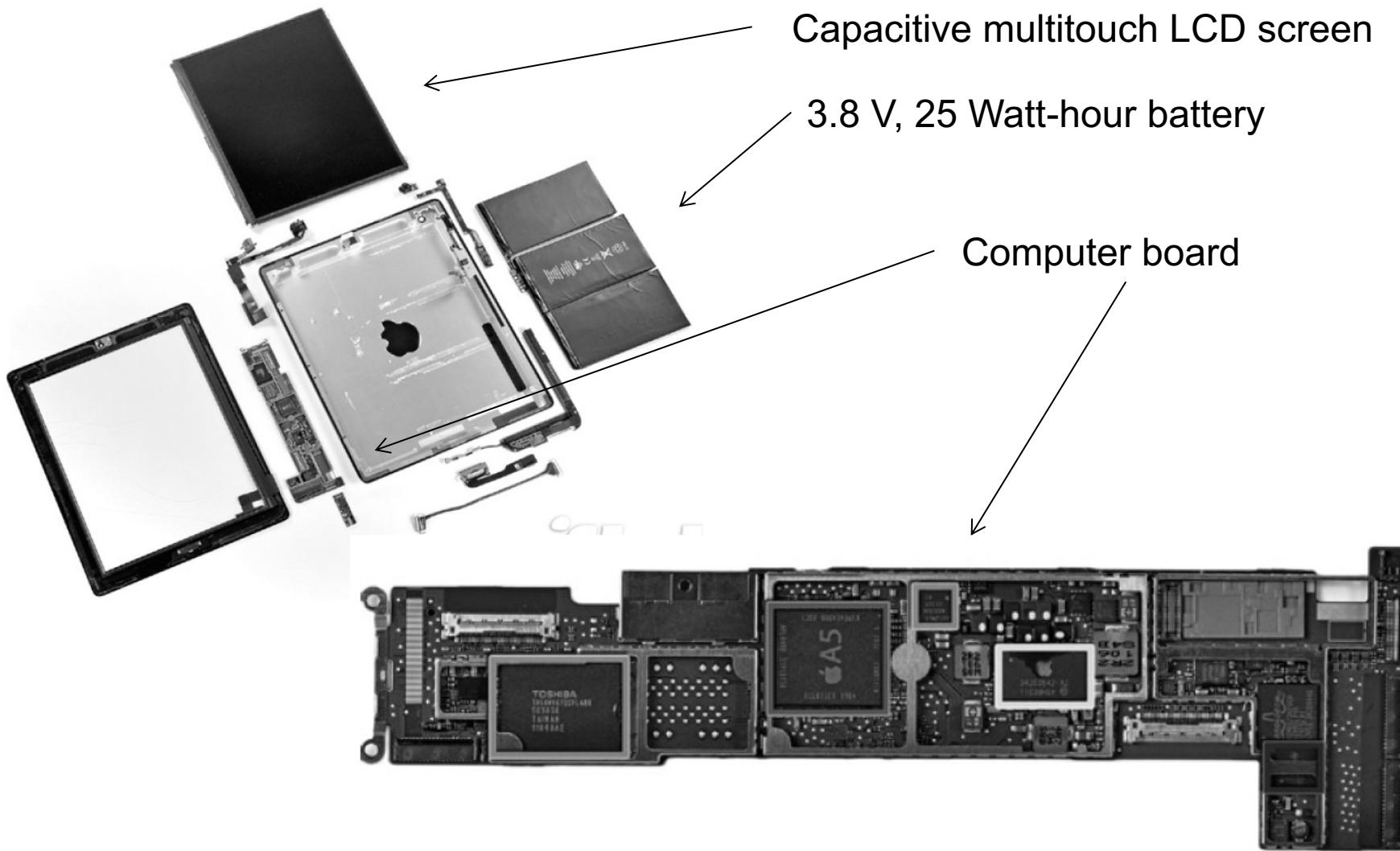
Center for Science, Technology, and Society

# Mobile vs. Laptop/desktop

- processing?
- Memory?
- OS?
- Peripherals?
- I/O?
- Why do we need to distinguish between them?

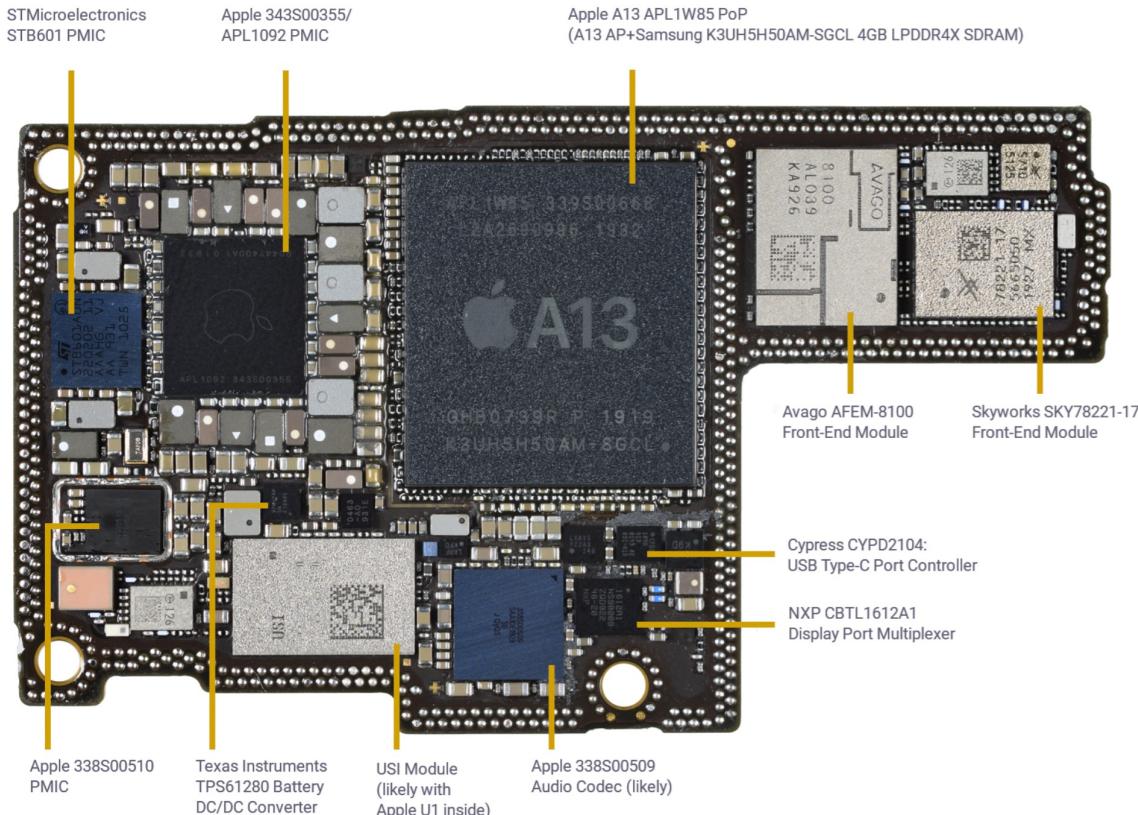


# Opening An iPad Box

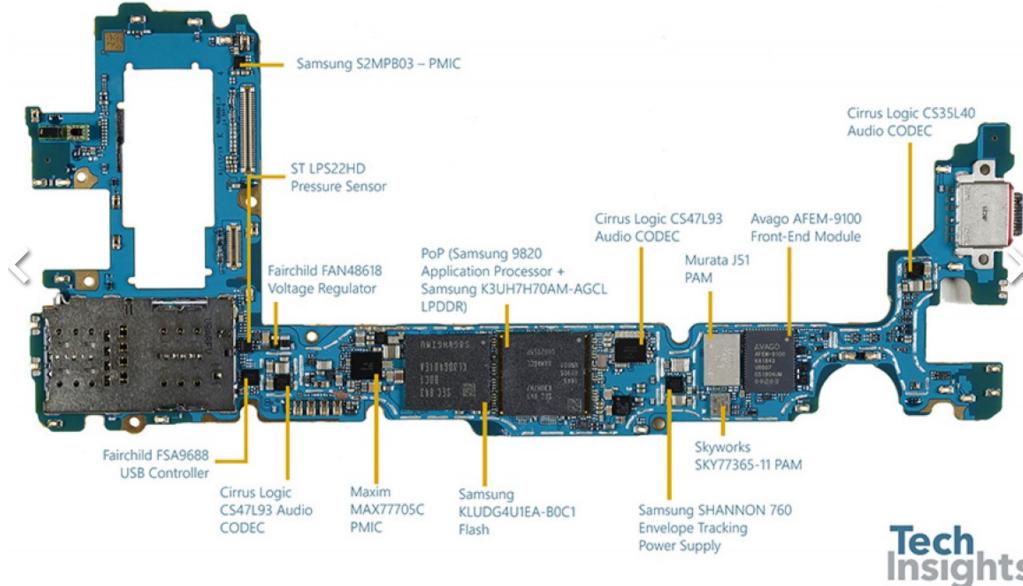


# iPhone 11 pro max

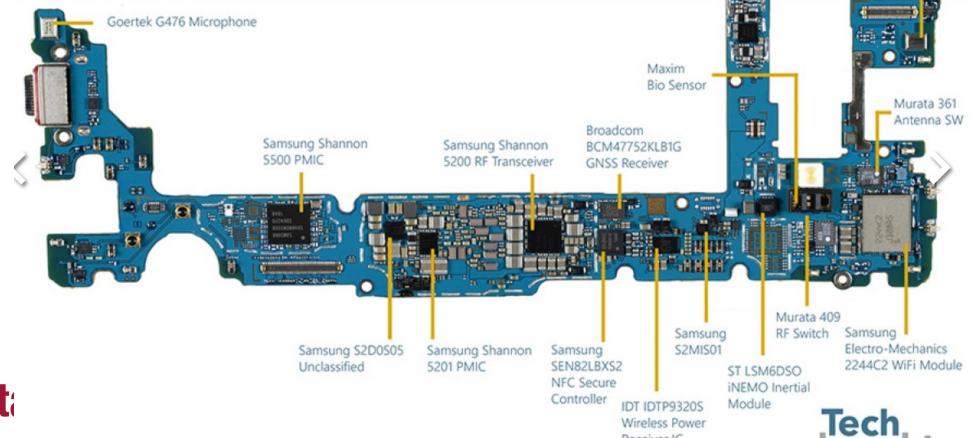
<https://www.techinsights.com/blog/apple-iphone-11-pro-max-teardown>



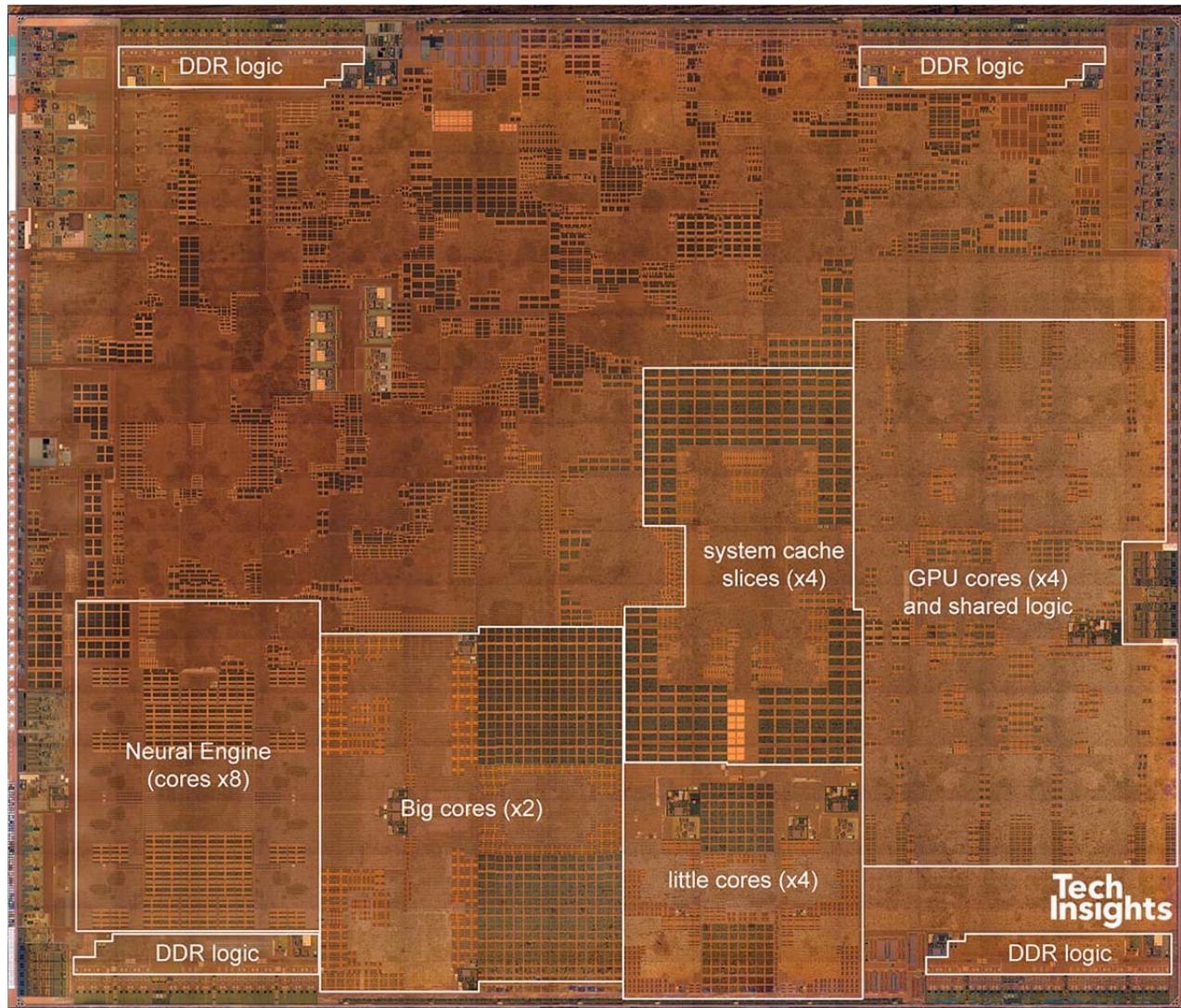
# Samsung S10+ blow up



Tech  
Insights



# Inside the Processor (A12)



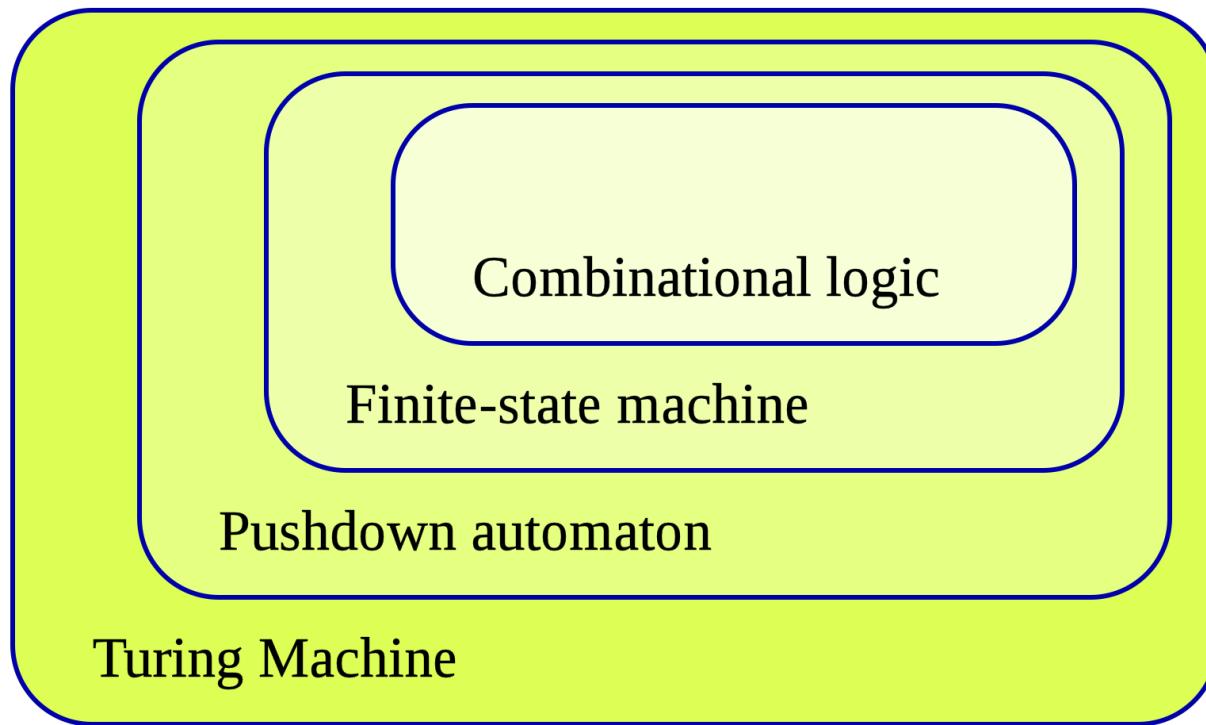
# FSM



**Santa Clara University**  
Center for Science, Technology, and Society

# Finite State Machine

Automata theory



Santa Clara University

Center for Science, Technology, and Society

# State Table

Present state	Inputs	Next State	Outputs

- A state table is similar to the truth table
  - present state
  - all inputs
  - next state
  - all outputs .

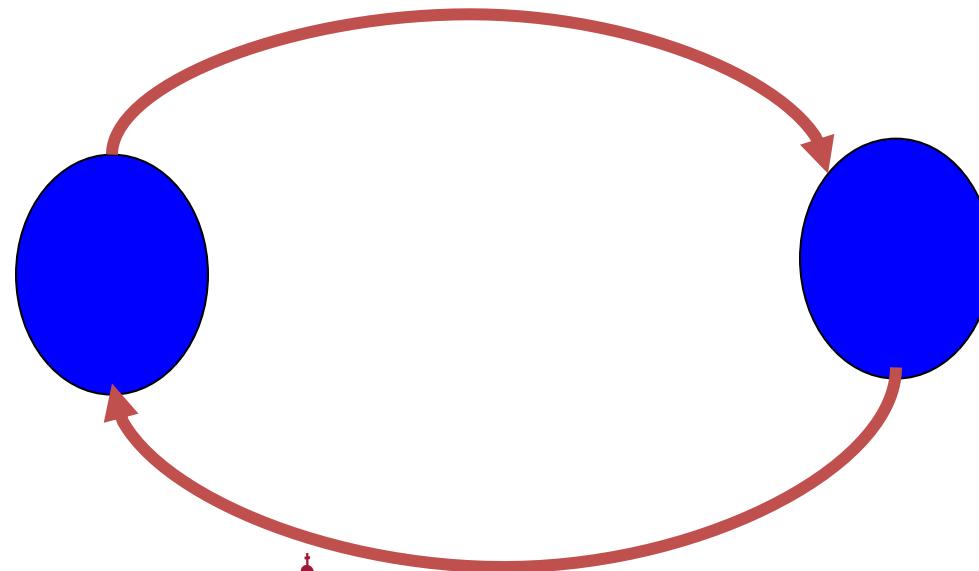
# Important Rule for State Table

- Complete state table must include
  - each possible combination of present states and input values
  - no such combination may match more than one row of the table



# State Diagrams

- A state diagram:
  - Each <sup>j</sup> state is represented by a circled **vertex**
  - Each row of the state table is shown as directed **arc**



# Important Rule for State Diagram

- State diagram has same situation as state table.
- Their conditions should be mutually exclusive,
- no input values should meet the condition of more than one arc.



# The Alarm Clock

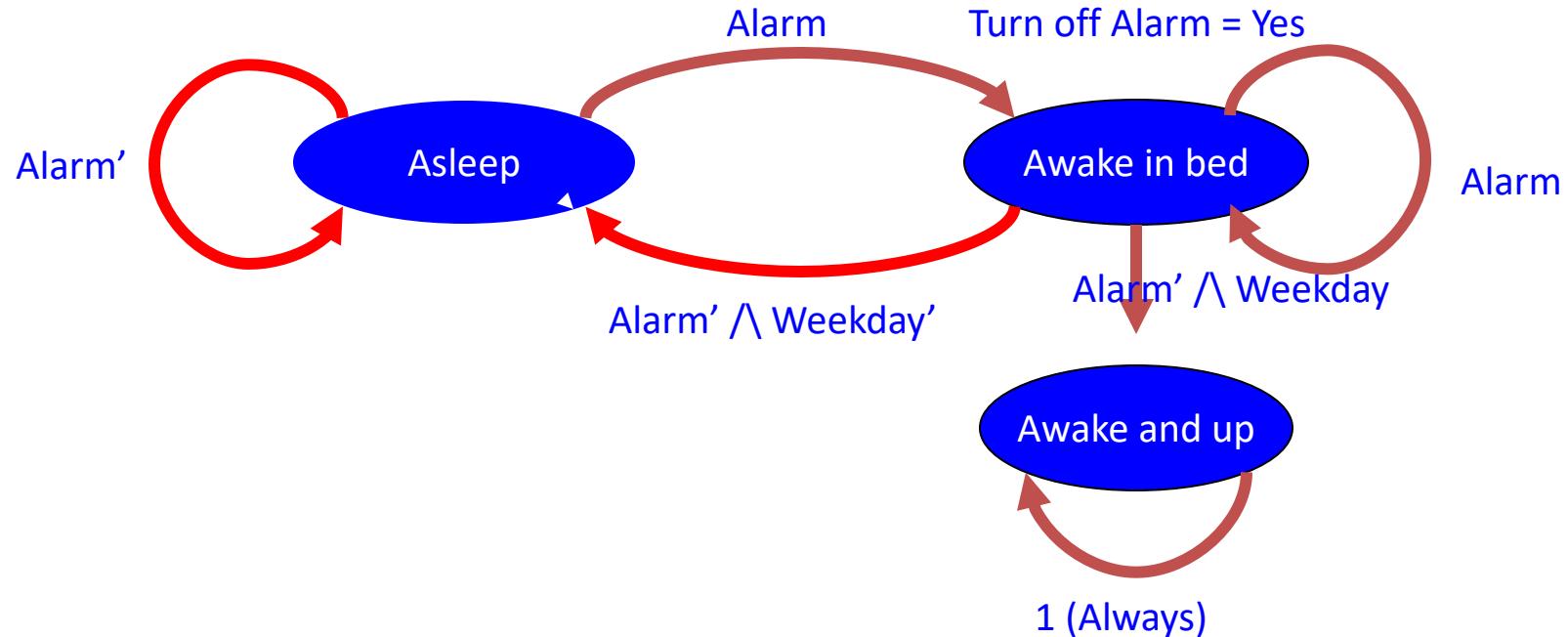
Present state	Alarm	Weekday	Next state	Turn off alarm
Asleep	On	X	Awake in bed	Yes
Awake in bed	Off	Yes	Awake and up	No
Awake in bed	Off	No	Asleep	No



Santa Clara University

Center for Science, Technology, and Society

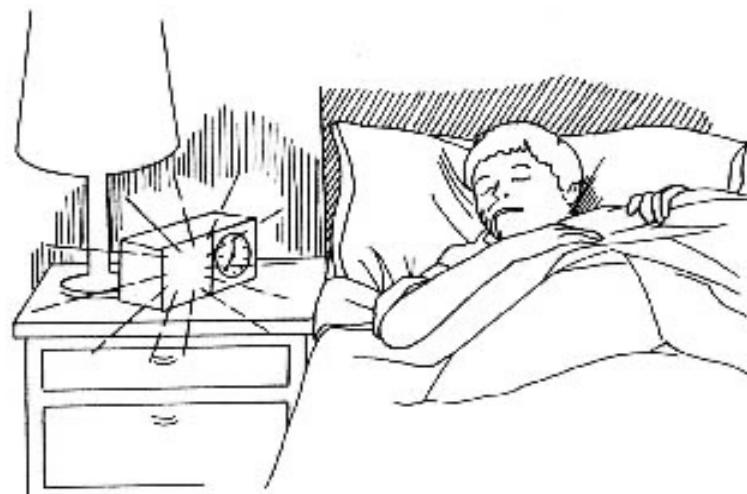
# State Diagram for The Alarm Clock (a)



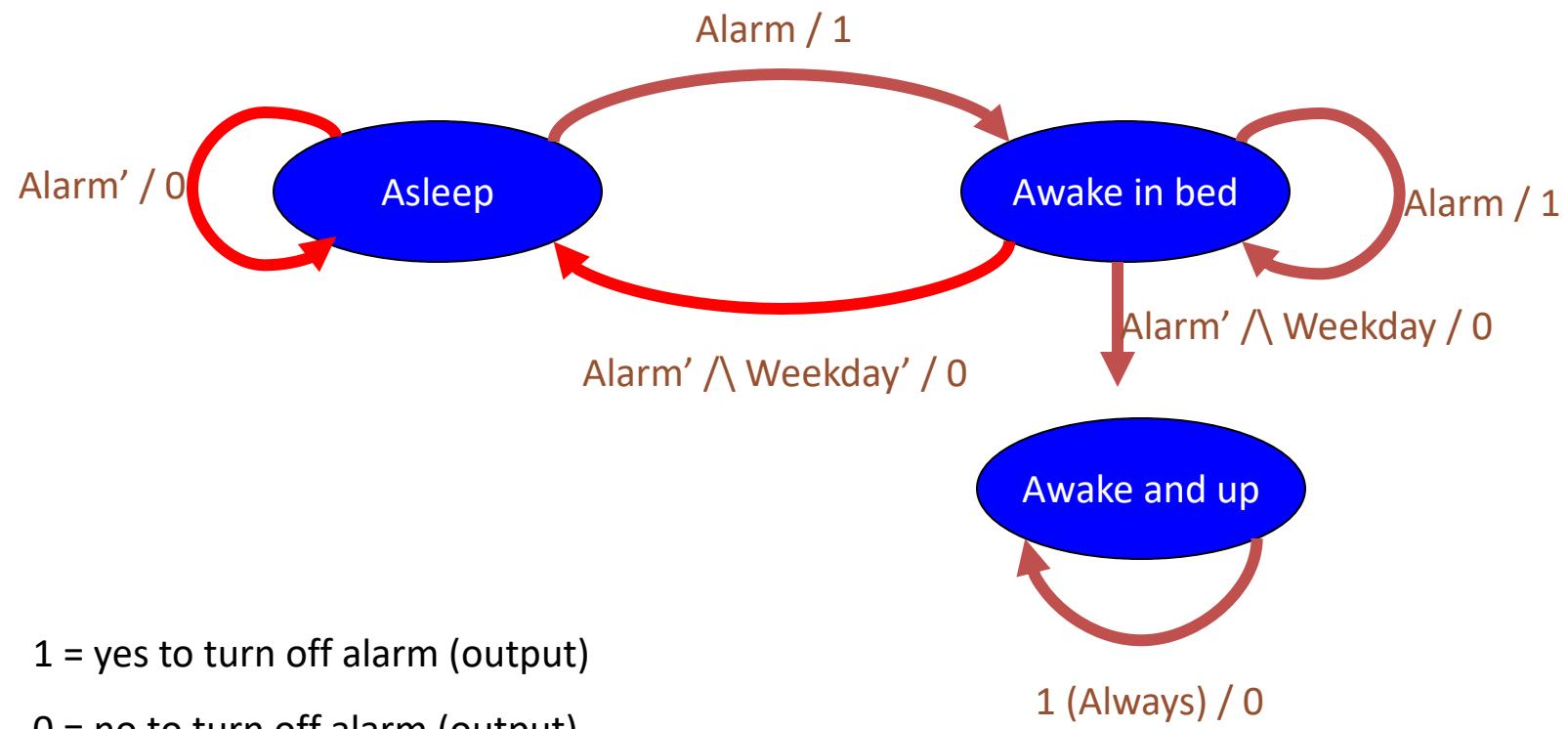
( a )

# The alarm clock problem with inaction states

Present state	Alarm	Weekday	Next state	Turn off alarm
Asleep	Off	X	Asleep	No
Asleep	On	X	Awake in bed	Yes
Awake in bed	On	X	Awake in bed	yes
Awake in bed	Off	Yes	Awake and up	No
Awake in bed	Off	No	Asleep	No



# State Diagram for The Alarm Clock (b)



# Mealy and Moore Machines

- A finite state machine can represent outputs in one of two ways
  - Moore Machines
  - Mealy Machines



# Moore Machines

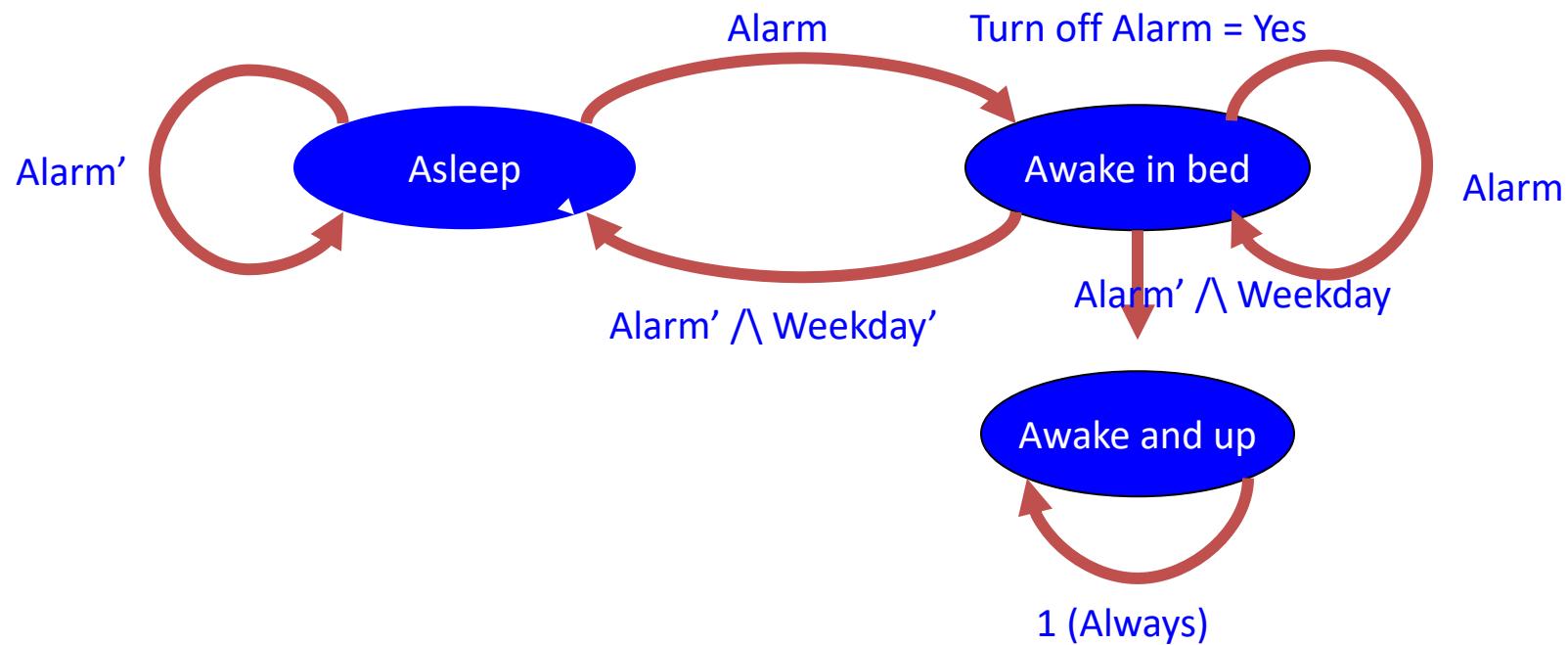
## – Moore Machines

- Associates its outputs with the states.
- Output values depend only on the state and not on the transitions.
- It requires less hardware to produce the output values
- It is well suited for representing the control units of microprocessors and cpu.



# State Diagram for The Alarm Clock (a)

## Moore Machine



Santa Clara University

Center for Science, Technology, and Society

# Mealy Machines

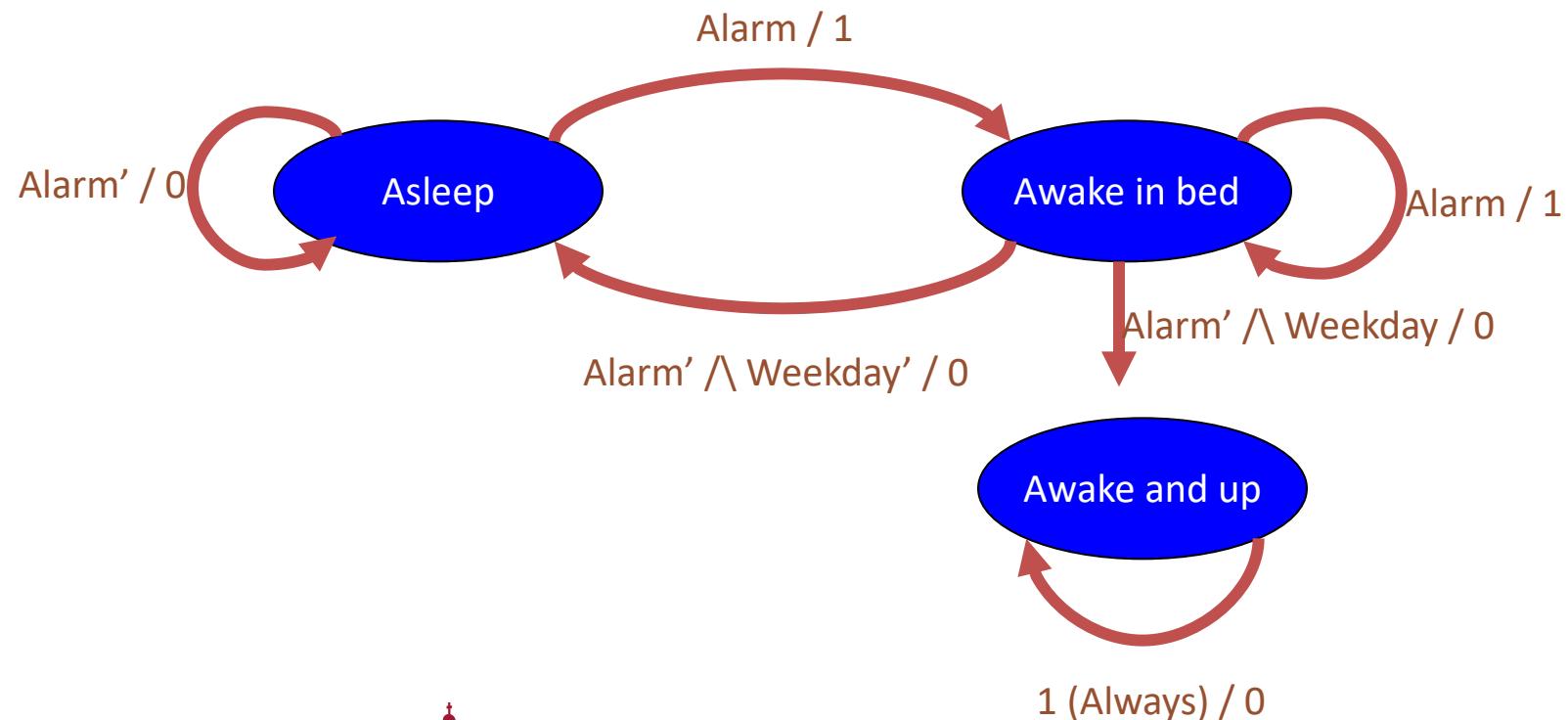
## – Mealy Machines

- Associates outputs with the transitions.
- It depends on both its state and its input values

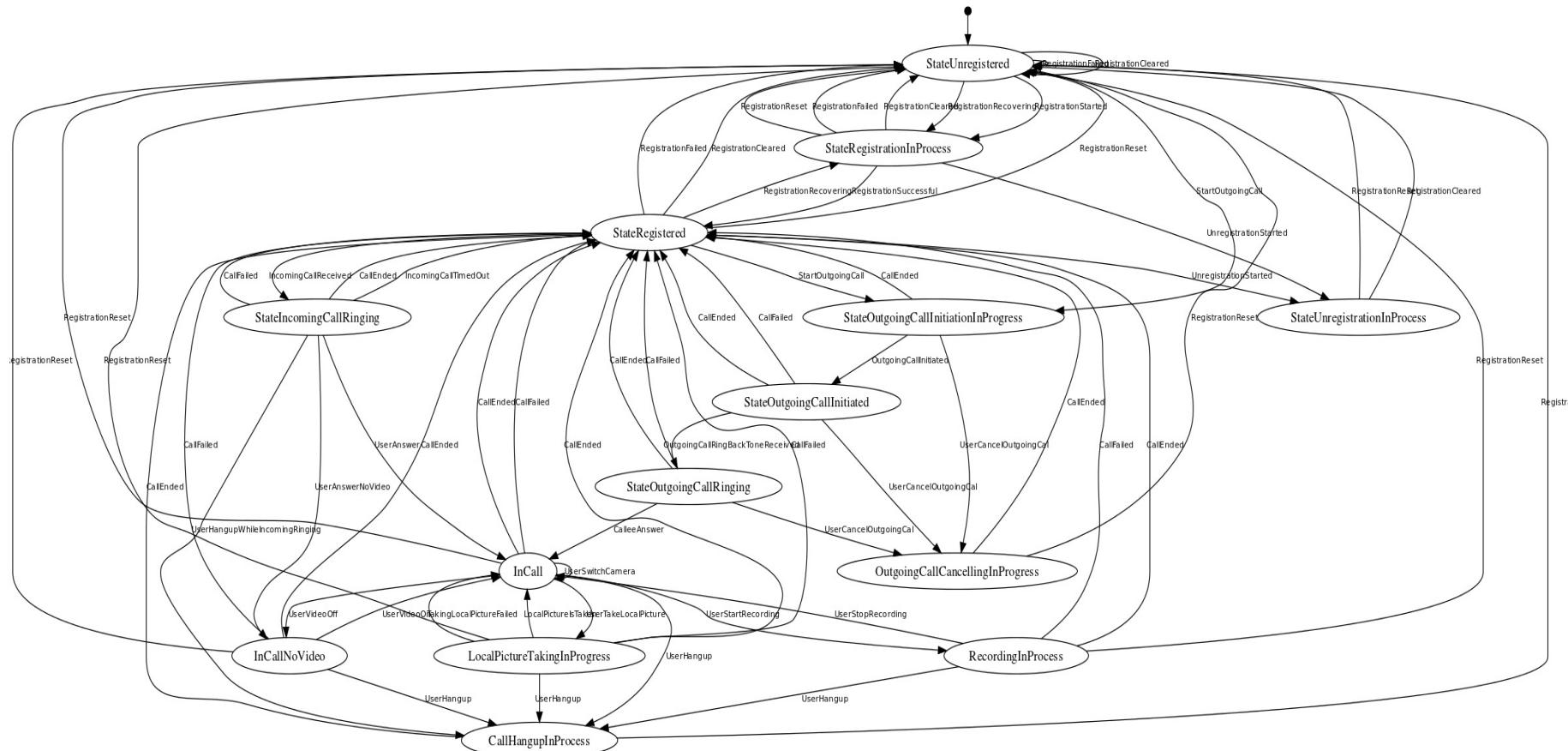


# State Diagram for The Alarm Clock (b)

## Mealy Machine



# FSM Example: Call Management



Santa Clara University

Center for Science, Technology, and Society

# Activity for Monday Class

- On your chosen app
  - Draw state diagram or state table
  - Create a 4-5 page slide deck
  - Upload slides to camino
  - Prepare to present your deck in the class

# Software Design Patterns



Santa Clara University

Center for Science, Technology, and Society

# Software Design Pattern

- In software engineering, a **software design pattern** is a general reusable solution to a commonly occurring problem within a given context (Wikipedia)
- software pattern is a template for solving a repeatable problem.
- Examples:
  - Concurrency Patterns: monitor, scheduler
  - Structural Patterns: translator, proxy
  - Creational Patterns: factory
  - Architectural/Behavioral Patterns: micro-services, peer-to-peer, MVC



# App Design Pattern

Any user facing client app can be broadly composed of three main layers

- **The Presentation / UI Layer:** Layer that a user interacts with.
- **The Business Logic Layer:** Layer that handles the data model and interactions for local/remote databases
- **The Application Logic Layer :** The glue between the Presentation and Business Logic Layers

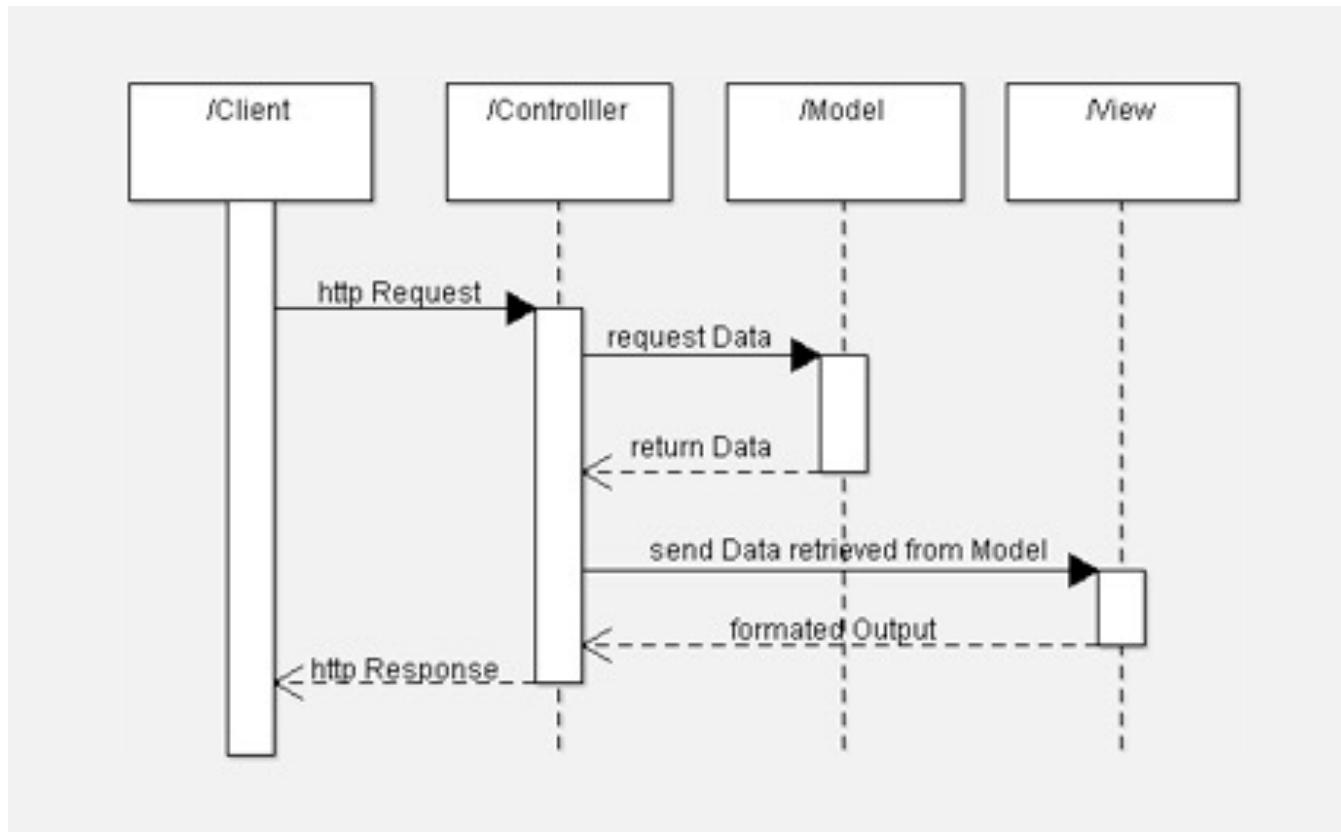


# Model-View-Controller Pattern

- The model view controller pattern is the most used pattern for today's world web applications
- It has been used for the first time in Smalltalk and then adopted and popularized by Java
- At present there are more than a dozen web frameworks based on MVC pattern



# MVC Flow Diagram



Santa Clara University

Center for Science, Technology, and Society

# Model-View-Controller Pattern

The MVC is a well-known architectural pattern used for building iOS and Android Apps

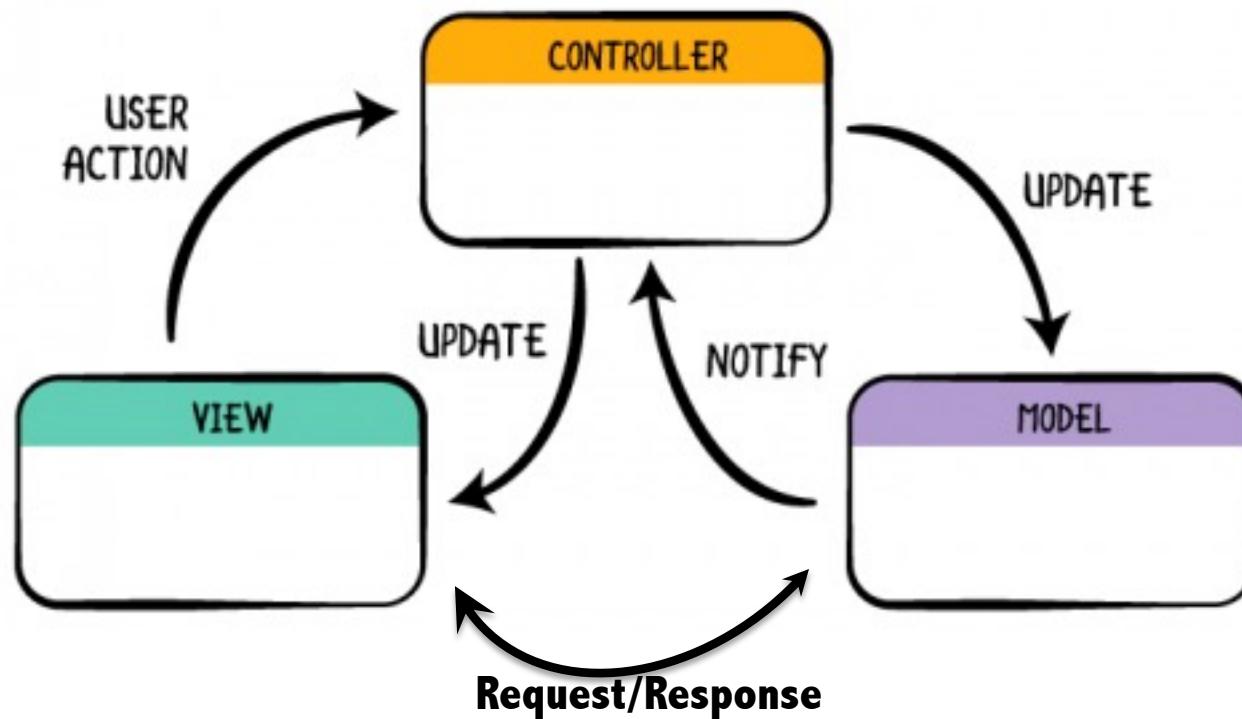
- View , which is generally not passive corresponds to the Presentation Layer
- Model corresponds to the Business Logic Layer
- Controller sits between the View and the Model , corresponds to the Application Logic Layer.



Santa Clara University

Center for Science, Technology, and Society

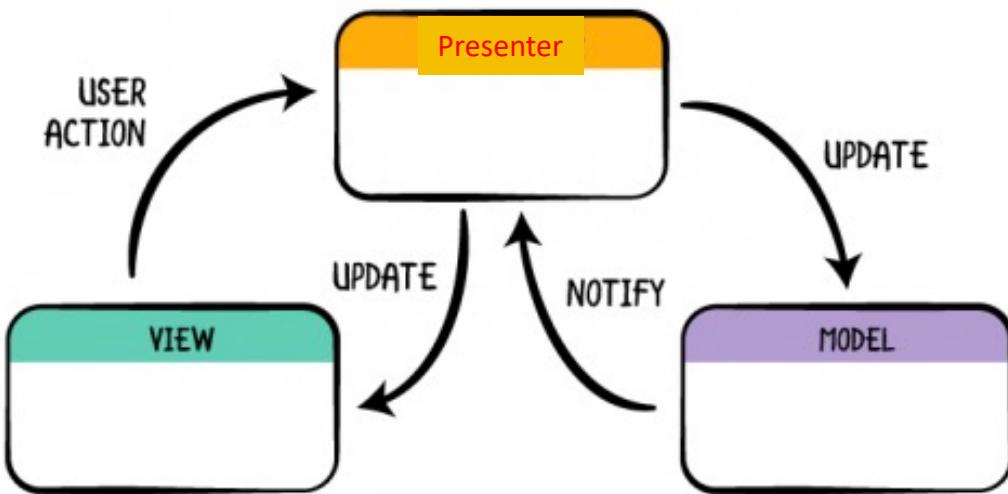
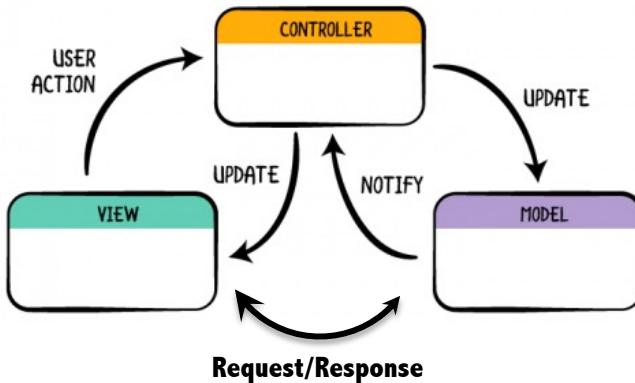
# MVC Pattern (2)



Santa Clara University

Center for Science, Technology, and Society

# MVC->MVP in Mobile App



Santa Clara University

Center for Science, Technology, and Society

# Thank You