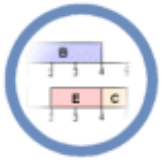


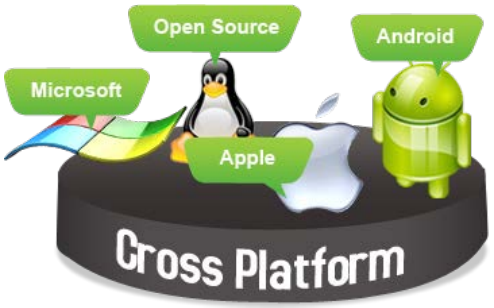


Lecture ”4”
Cross-Platform Development

<lecturer, date>

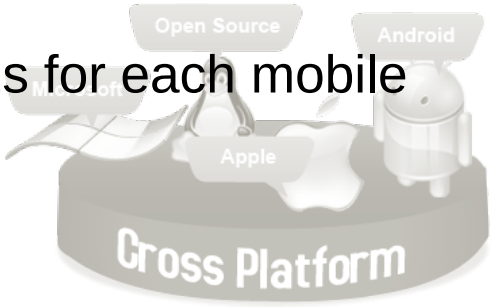
Outline

- **Cross-Platform Development**
- PhoneGap
- Appcelerator Titanium
- Xamarin
- References



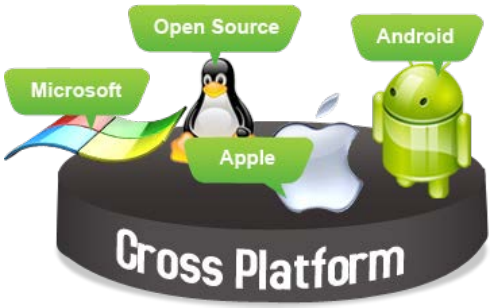
Native Development

- Represents the baseline for comparisons
- You essentially develope in the prescribed way for that specific OS, i.e.,
  - iOS = Xcode & Objective-C
  - Android = Android Studio/Eclipse & Java
  - Windows Phone = Visual Studio & C++/C#
- Your code base can not be reused across platforms
- You need to have skills in all languages and IDEs for each mobile platform you're targeting



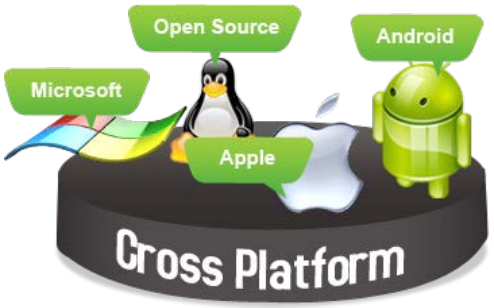
Cross-Platform Development

- An attempt to deliver the same app across multiple platforms
- Cross-platform mobile development
  - ☞ Sharing a code base between multiple platforms



Pros

- Reuse of code
- Less duplicated work
- Easier to maintain code base
- Better consistancy in implementations
- Leveraging the existing skills on different platforms
- Lower development cost
- Quicker to market for multiple platforms

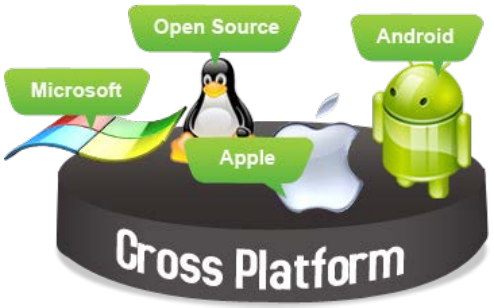


Cons

- Weaker processing performance
- Weaker user interface responsiveness
- User interface inconsistent with OS
- Slower start-up times
- Increased complexity due to additional layer of abstraction
- Inability to use latest features of the native OS
- Poor interoperability with 3d party APIs

☼ However, they do not all apply to all frameworks

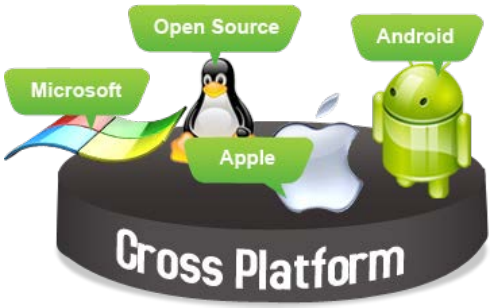
☼ The level of impact may be negligible



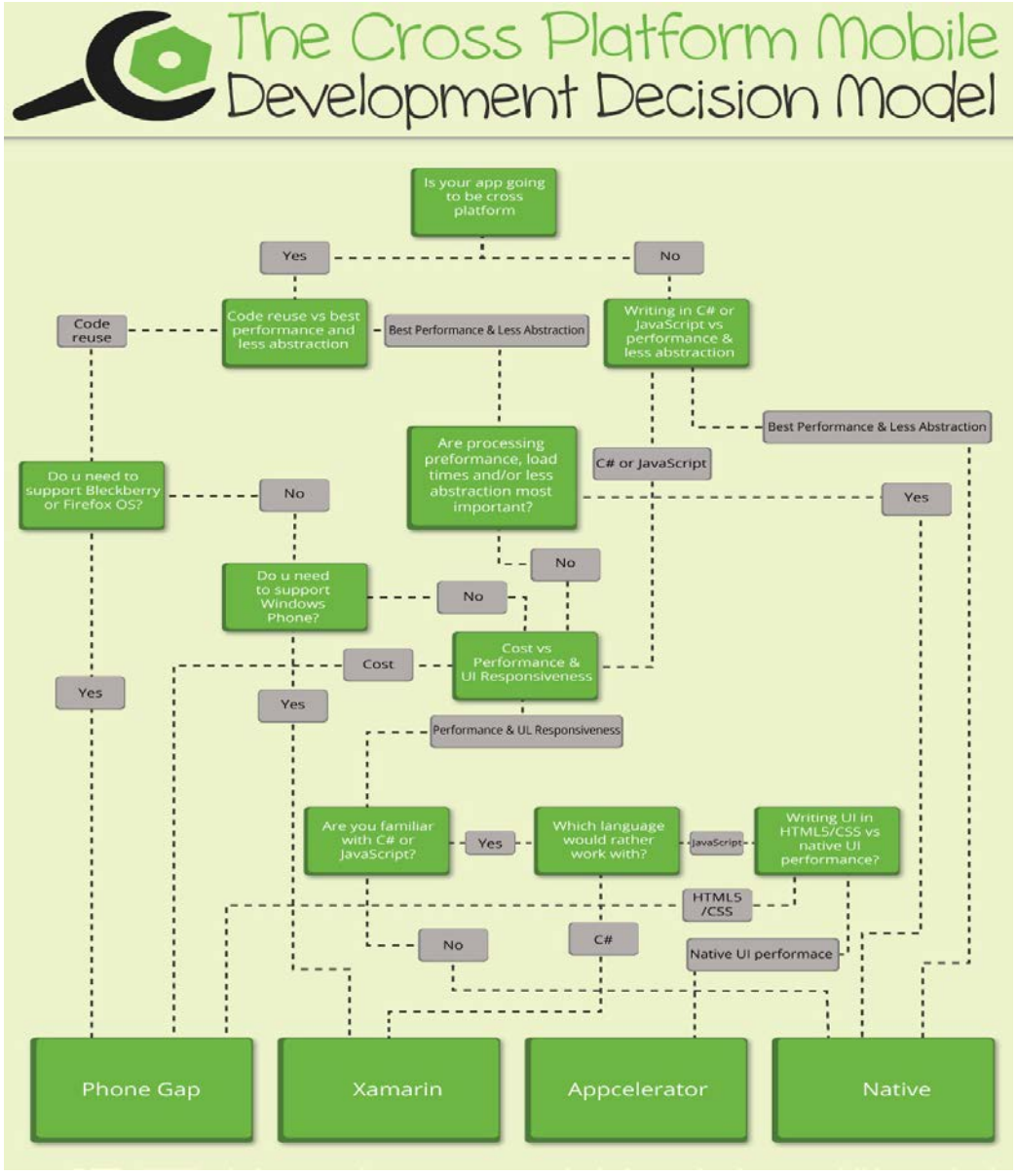
Cross-Platform Development Decision

✓ Prioritisation

- Reusable code base?
- Processing performance?
- User interface responsiveness?, …

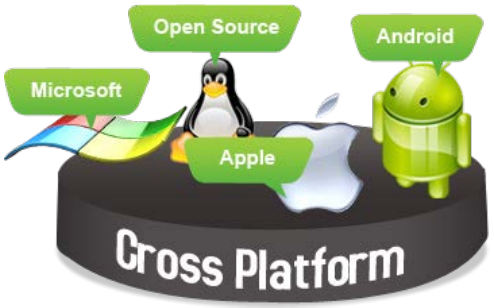


Cross-Platform Development Decision



Cross-Platform Native Framework

- A software that allows a common development approach across platforms
- Builds to an app that is indistinguishable by a user from one built with native code
- Many cross-platform frameworks have emerged in the past few years

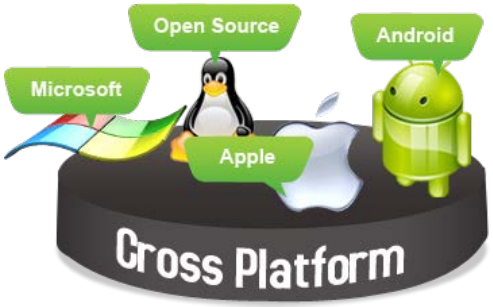


Framework Categories

1. Frameworks that let you create a native mobile application using cross-platform APIs, HTML/CSS/Javascript frameworks

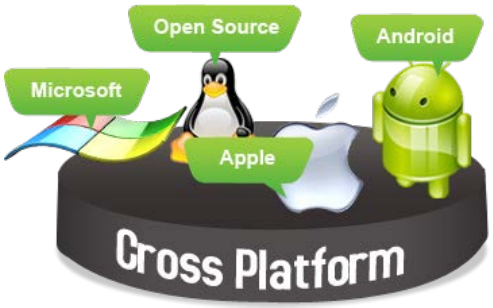
2. Those that let you build cross-platform interfaces that run in a web browser

☞ It is common to combine these to create cross-platform native apps



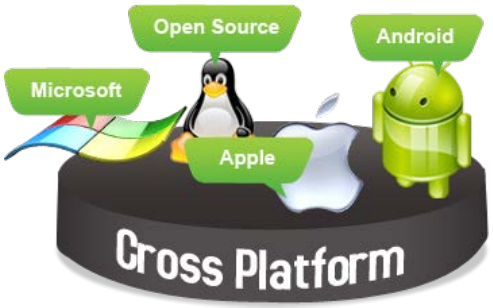
Outline

- Cross-Platform Development
- **PhoneGap**
- Appcelerator Titanium
- Xamarin
- References



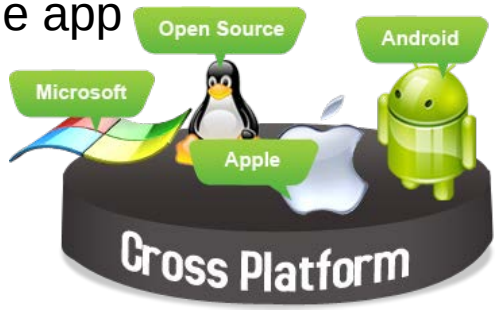
PhoneGap

- From Nitobi
- An open source framework for building native mobile apps using HTML, CSS, and Javascript for iPhone, Android, BlackBerry
- PhoneGap then provides a standard API which allows you to work with features of the device in a JavaScript.



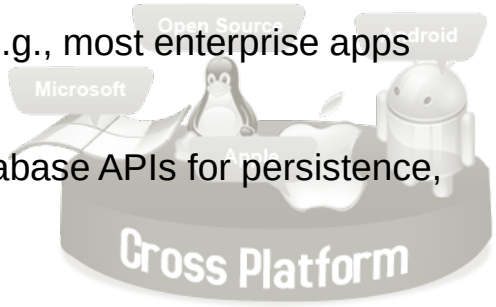
Pros

- ✓ The big positives to PhoneGap are that the level of reuse is very high
- ✓ PhoneGap has the best presence across different platforms as it currently supports iOS, Android, Windows Phone, Blackberry, Firefox OS, …
- ✓ Also depending on your background, JavaScript, HTML5 and CSS are pretty well known and if you have experience with web development and as such it might provide an easier entry into mobile development
- ✓ Perfect for transforming a mobile web app to a native app
- ✓ PhoneGap provides a rich collection of client-side JavaScript APIs with a method for hosting your web app within a native mobile app



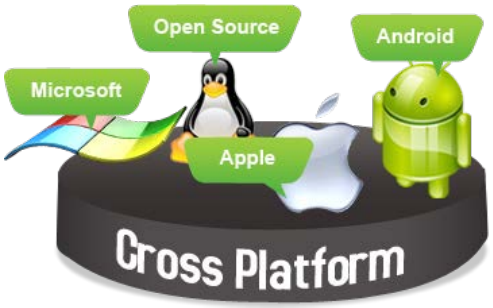
Cons

- ✕ Main actual language with PhoneGap is JavaScript
  - building large apps with it is more difficult than with a language such as C# or Java
- ✕ Anything related to performance is going to be the worst of all options with PhoneGap (responsiveness and processing especially)
  - Processing is generally not that important for most mobile apps
  - Responsiveness isn't terrible
    - Its more that you can notice it if you're looking for it, but it probably wouldn't stop you from using an app
- ✕ Not well-suited for apps that require intense math calculations or 3-D animations
- ✕ Neither well-suited for developers needing to write data-driven apps e.g., most enterprise apps that must work offline using sync'd local data
- ✕ Does not provide specific database support and relies on HTML5 database APIs for persistence, which are not widely available



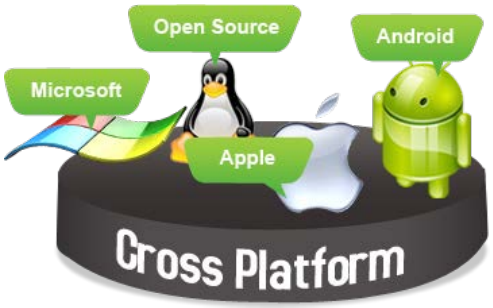
PhoneGap API

- When running inside PhoneGap, your app can access certain devices capabilities from JavaScript that are not otherwise available to web apps.
- The PhoneGap API provides access to the following device capabilities:
  - ✓ Geolocation
  - ✓ Contacts
  - ✓ Vibration
  - ✓ Accelerometer
  - ✓ Camera
  - ✓ Sound playback
  - ✓ Device information
  - ✓ Click to call



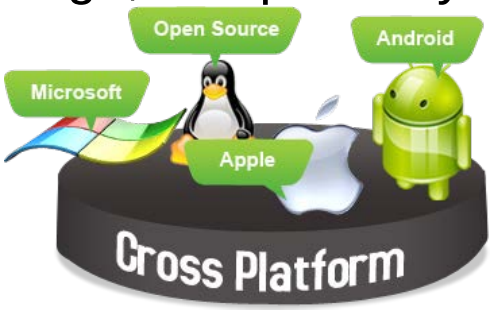
Outline

- Cross-Platform Development
- PhoneGap
- **Appcelerator Titanium**
- Xamarin
- References



Appcelerator Titanium

- Writing your app with JavaScript
  - However, no HTML5 and CSS for the user interface
  - You'll be using a custom API or XML language
- The result of this JavaScript and UI API is then compiled into an app that is like a native & JavaScript hybrid
- With Appcelerator the level of code re-use is very high, and probably as high as PhoneGap

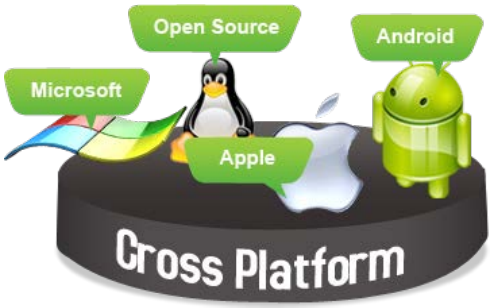


Appcelerator Titanium

- For 100% level of code reuse you have to program to the lowest common denominator, i.e.,
  - Appcelerator API provides a way to create a UI for both platforms in a common way
  - However, not all platforms support the same features
  - So you can only develop the UI using features that all target platforms support
- If you're confident with JavaScript then you already have a jump start with Appcelerator, but there will be a little more learning curve as you have to work with the UI API
- Additional complexity in building a large app

Pros

- ✓ Performance and responsiveness of an Appcelerator app is going to be noticeably better than that of PhoneGap
- ✓ UI is essentially native, i.e., user interface will feel and handle just like a native app, even though everything you've written will be in JavaScript



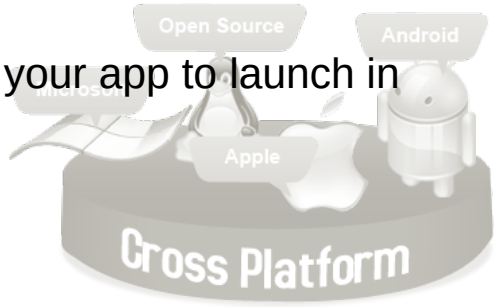
Cons

✕ May be a slight impact on processing performance as

- Appcelerator has to proxy code through to a JavaScript execution engine
- However, unless you're doing some fairly heavy processing it should be unnoticeable

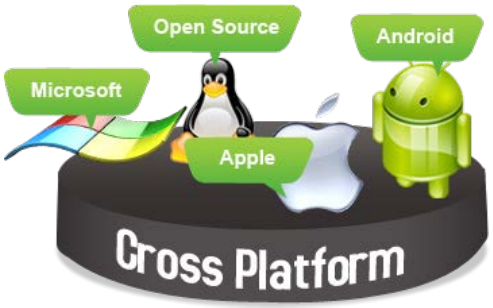
✕ May also be a small impact on app load times as

- With any of these cross platform libraries they need loading up
- However, shouldn't be a huge problem unless you need your app to launch in fractions of a second



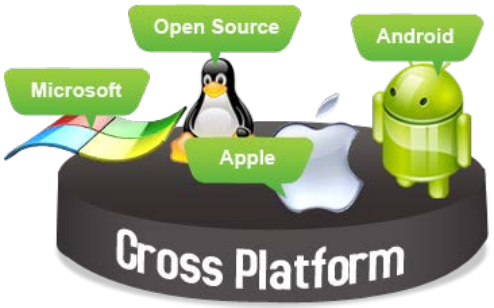
Cons

- ✕ Appcelerator currently only supports iOS and Android, so
  - If you need your app to run on Windows Phone or Blackberry its a non-starter
  - However, if you're only concerned with iOS and Android its a great option
- ✕ Pricing for anything deployed to an app store/significant downloads is high



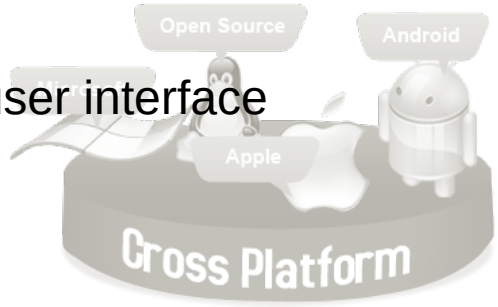
Outline

- Cross-Platform Development
- PhoneGap
- Appcelerator Titanium
- **Xamarin**
- References



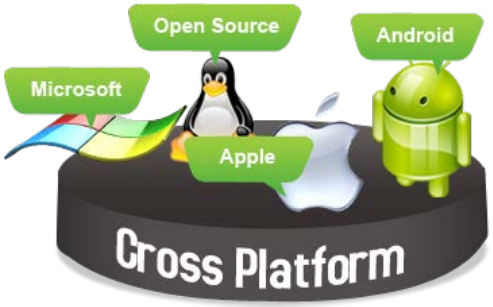
Xamarin

- In some respects very similar to Appcelerator
  - Except that you're writing your code in C#
  - Your C# app logic can be reused across all platforms
- For your UI you have a couple of options:
  1. Use a native designer
  2. Use Xamarin.Forms
- The result of this is a native app, running with native user interface components



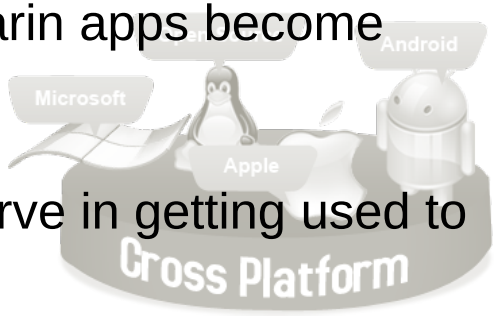
Xamarin

- Strongly typed languages such as C# are considered easier to build larger apps with than JavaScript
  - As you get a lot better early warning signs from the compiler
  - better refactoring options
  - However, if you're already familiar with JavaScript you may find Appcelerator or PhoneGap easier to pick up
- Performance and responsiveness will be similar to Appcelerator as the code is basically native
  - May be some slight differences
  - But for the most part it should be unnoticeable
  - However does impact initial load times as it has to load its own runtime
- Currently supports iOS, Android and Windows
  - Better coverage than Appcelerator,
  - Slightly worse coverage than PhoneGap



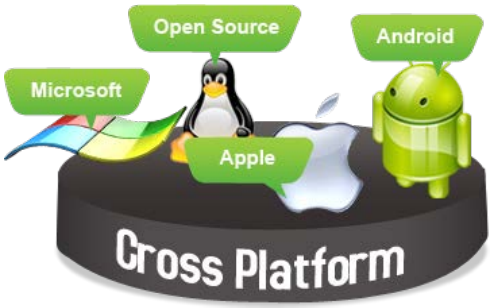
Pros

- ✓ The level of code reuse is very high with all app logic being shared across platforms
- ✓ Depending on your approach the UI could also be shared
- ✓ For 100% reuse option with Xamarin.Forms
  - Similar to Appcelerator, your UI will be built to a lowest common denominator
  - However, you have the option to cater your UI's for specific platforms
- ✓ If you're used to using C# and .NET, then C# Xamarin apps become easy
- ✓ Similarly to Appcelerator there will be a learning curve in getting used to the custom API and Xamarin's nuances



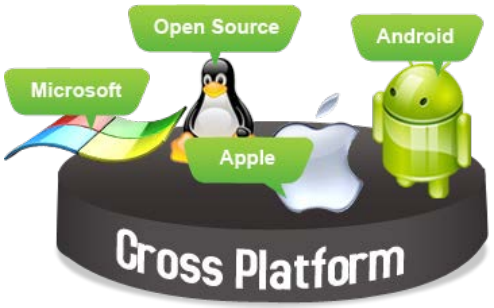
Cons

- Does have a considerable cost associated with it (you'd be looking at over a thousand $)
- The exploratory version is far more restrictive than Appcelerator



Outline

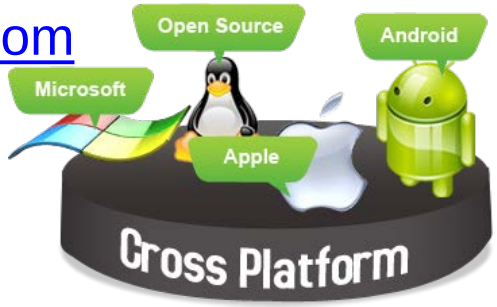
- Cross-Platform Development
- PhoneGap
- Appcelerator Titanium
- Xamarin
- **References**



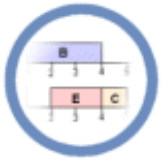
References

- Lundrigan, L., Graupera, V.,, Allen, S. (2010). Pro smartphone cross-platform development - iPhone, BlackBerry, Windows Mobile, and Android Development and Distriubution. PA USA: Apress.

- PhoneGap:
  - http://www.phonegap.com/
  - http://wiki.phonegap.com/Roadmap
- Appcelerator Titanium: http://www.appcelerator.com
- Xamarin: http://xamarin.com/







Lab ”4”

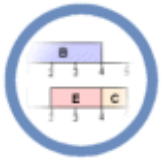
Basic App Development in Android

<lecturer, date>

Lab ”4”

- Lab 4 includes
  - A brief introduction to the basic application development,
  - Getting familiar with the User Interface (UI) components,
  - Making the connection of the representation of the Water Tank Controller design with these components (widgets),
  - Also includes exercises that are useful for the Water Tank Controller development. References and links for additional studying are also provided.



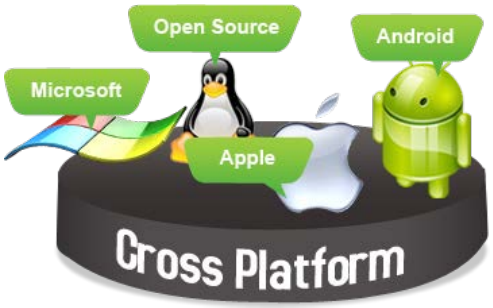


Seminar ”4”
Cross-Platform Development

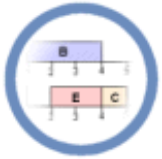
<lecturer, date>

Seminar ”4”

- Run your Hello World app on the Emulator
  - Discuss the steps and compare to when running the app on the real device







Mini-Project ”4”
Cross-Platform Development

<lecturer, date>

Seminar ”4”

- Discuss Rhodes platform features and compare it to the ones discussed in this lecture
  - Resources
    - Lundrigan, L., Graupera, V.,, Allen, S. (2010). Pro smartphone cross-platform development - iPhone, BlackBerry, Windows Mobile, and Android Development and Distriubution. PA USA: Apress.
    - http://rhomobile.com/