

# MOBILE OPERATING SYSTEMS AND APP DEVELOPMENT PART 2 - APPIFICATION

Ashish Agrawal  
Indian Institute of Technology Kanpur



**MOOC4D**

massive open online courses  
for development

MOOC on M4D 2013

# Appification

- Web application
  - Accessible through browser
  - May require changes in User Interface
  - Independent of device
- Mobile app
  - All/Some application components resides on client device
  - Better user experience
  - Better performance
- Questions?
  - What should we move ?

# App Development

- Native applications
  - Apps built using support provided by the Mobile OS
    - Platform specific software development kit (SDK) to use device features
  - Android : Java (compile on any platform)
  - iOS : Objective C (need mac to compile)
- Cross platform applications
  - HTML5, CSS, Javascript

# Native vs HTML5 Apps

- **Functionality**
  - HTML5 does not support all kind of device features
- **User experience**
  - Native apps can use OS UI components
- **Performance**
  - Native apps are faster
- **Reusability**
  - HTML5 apps can reuse existing code, Cross Platform Apps
  - Native apps: build separate apps for all platforms
- **Updates**
  - Native apps need updates for all platforms
- **Programming language**
  - Native: need expertise in platform specific languages

# Cross Platform Framework

- Build your application using
  - HTML, CSS, Javascript
  - Sometime vendor specific library and programming language
  - Framework will compile the code into platform specific hybrid app
    - HTML5 for most of the part
    - Native for features not supported by HTML5
- Developer needs to write only single code
- Vendor lock-in, less performance

# Cross Platform Frameworks

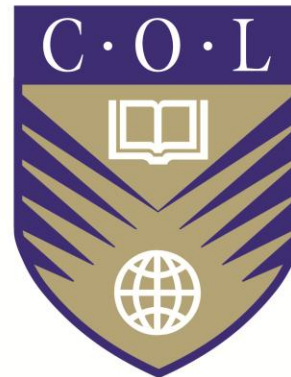


- <http://setandbma.files.wordpress.com/2011/12/wora-platforms.png>

# Summary

- Appification
  - Create an app for mobile device
- Appification vs Web
- Native app vs HTML5 app vs Hybrid app

# THANK YOU



**MOOC4D**  
massive open online courses  
for development

MOOC on M4D 2013