

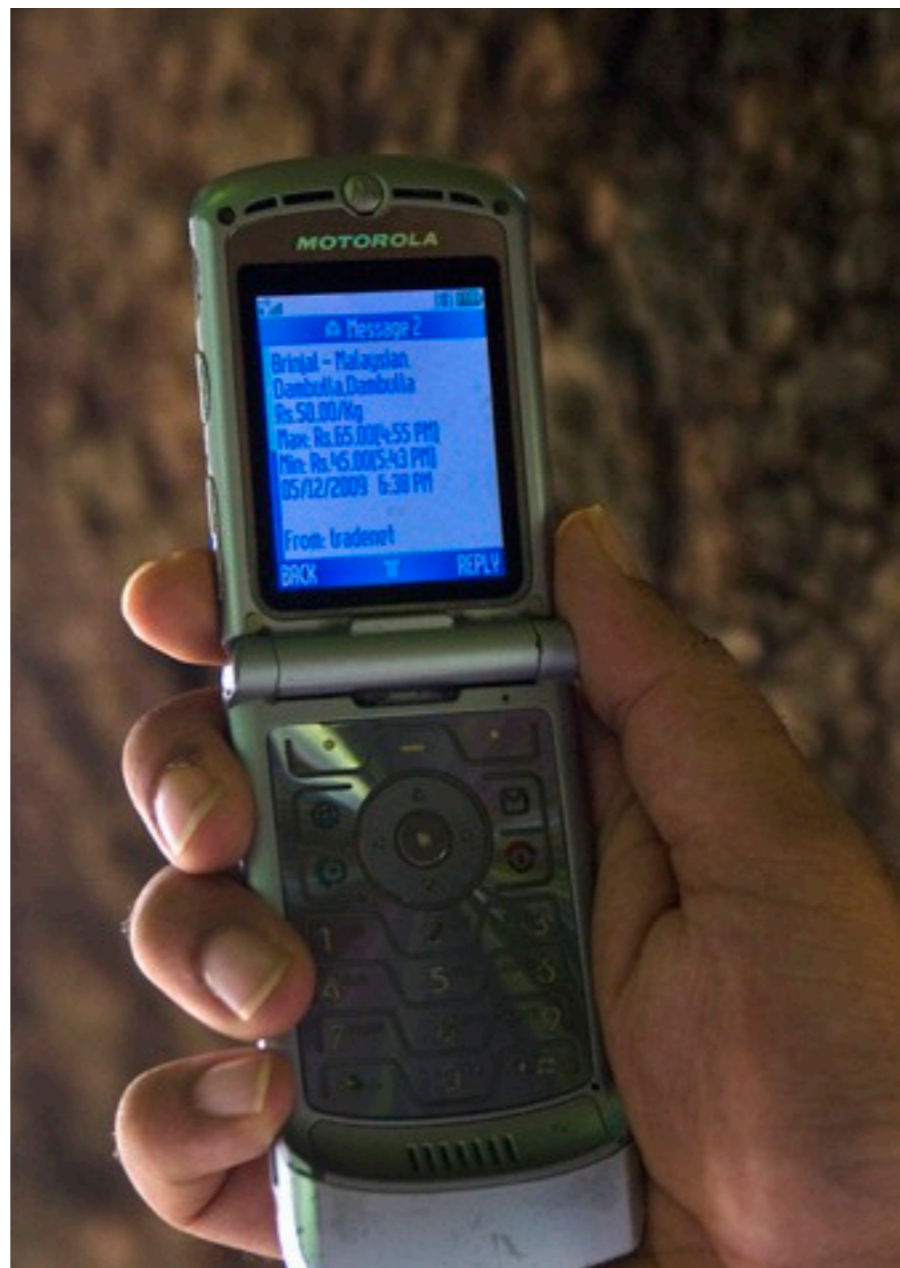


Accelerating Information Technology Innovation

<http://aiti.mit.edu>

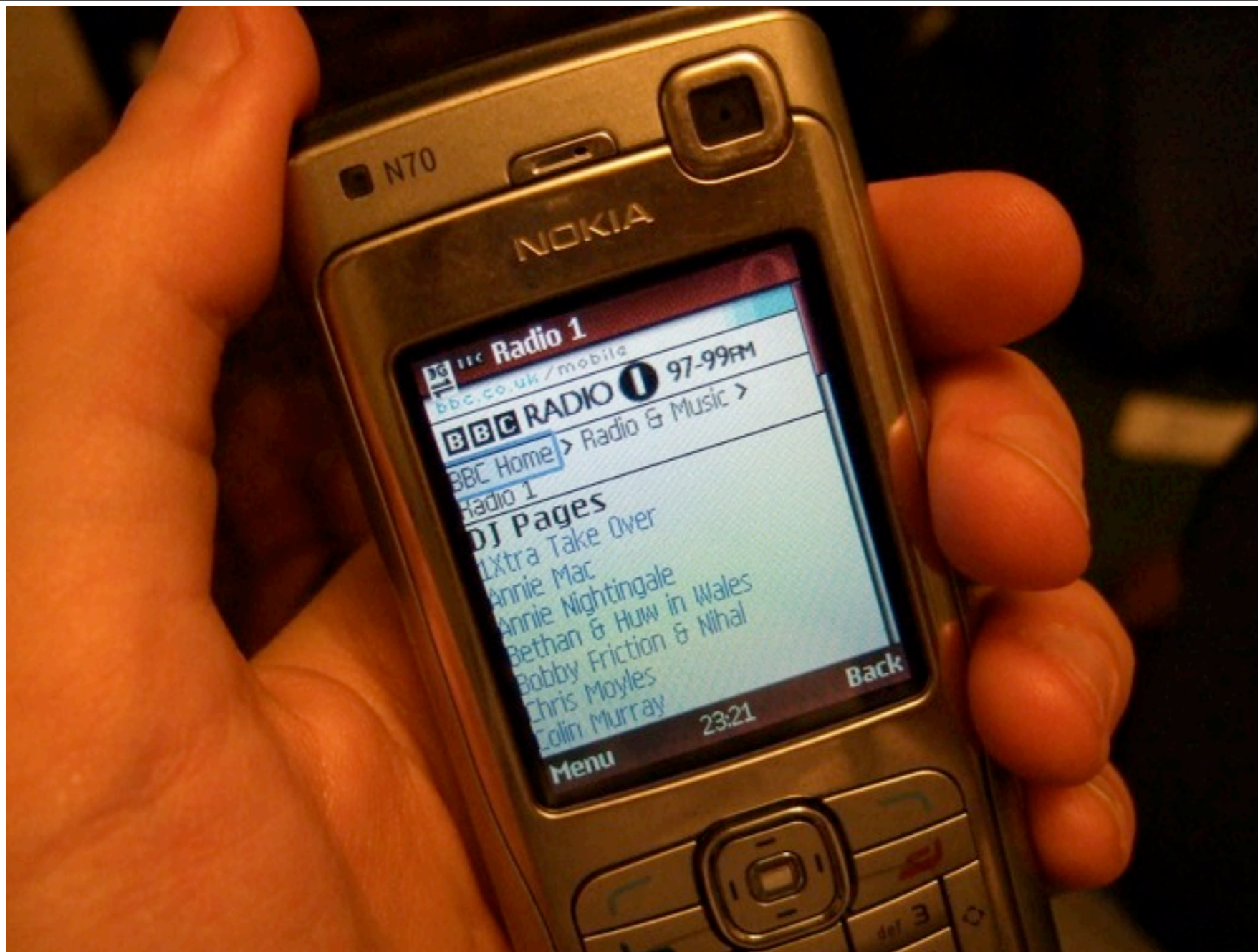
India Summer 2012
Lecture I – Introduction





“GGS SMS” by Sriganesh Lokanathan/LIRNEasia
Licensed under a Creative Commons Attribution 2.0 Generic Licence
<<http://www.flickr.com/photos/lirneasia/4459872700/>>

SMS Apps



“Radio 1 wap site” by James Cridland
Licensed under a Creative Commons Attribution 2.0 Generic License
<<http://www.flickr.com/photos/jamescridland/1470202502/>>

Web Apps (and WAP)



www.nrkbeta.no - Marius Arnesen (CC)

“iPhone vs.Android” by nrkbeta.no

Licensed under a Creative Commons Attribution-Share Alike 3.0 Norway License

<<http://www.flickr.com/photos/nrkbeta/3905907681/>>

Smartphone Apps

**Goal: Make a great demo
that can become a product!**

What we will cover

- Mobile Component: Android (Java)
- Web Component: Django (Python)
- Advanced topics in software design
 - Technologies: Regular expressions, translation...
 - Software Lifecycle: Testing/QA/Version Control
 - Other topics as needed (SMS? Other platforms?)

This Week

- **Android basics**
 - User Interface: Layouts and controls
 - User Interface: App structure and navigation
 - Interacting with the web
- **Version control and Git, unit tests and JUnit**
- **Lab: Making a basic app, and building on it**

An Introduction to Android

Android Devices



“Samsung Galaxy Nexus” by Sham Hardy

Licensed under a Creative Commons Attribution-Share Alike 2.0 Generic License

<http://www.flickr.com/photos/xshamx/6552938889/>

Phones

Android Devices



“Android-based tablet” by Tom Purves
Licensed under a Creative Commons Attribution 2.0 Generic License
<<http://www.flickr.com/photos/thomaspurves/4365881879/>>

Tablets

Android Apps

- Java-based
- Single, signed app file
- Apps usually on an app store
 - e.g. Google Play
- Runs separate from all other apps (usually)

Components of an Android App

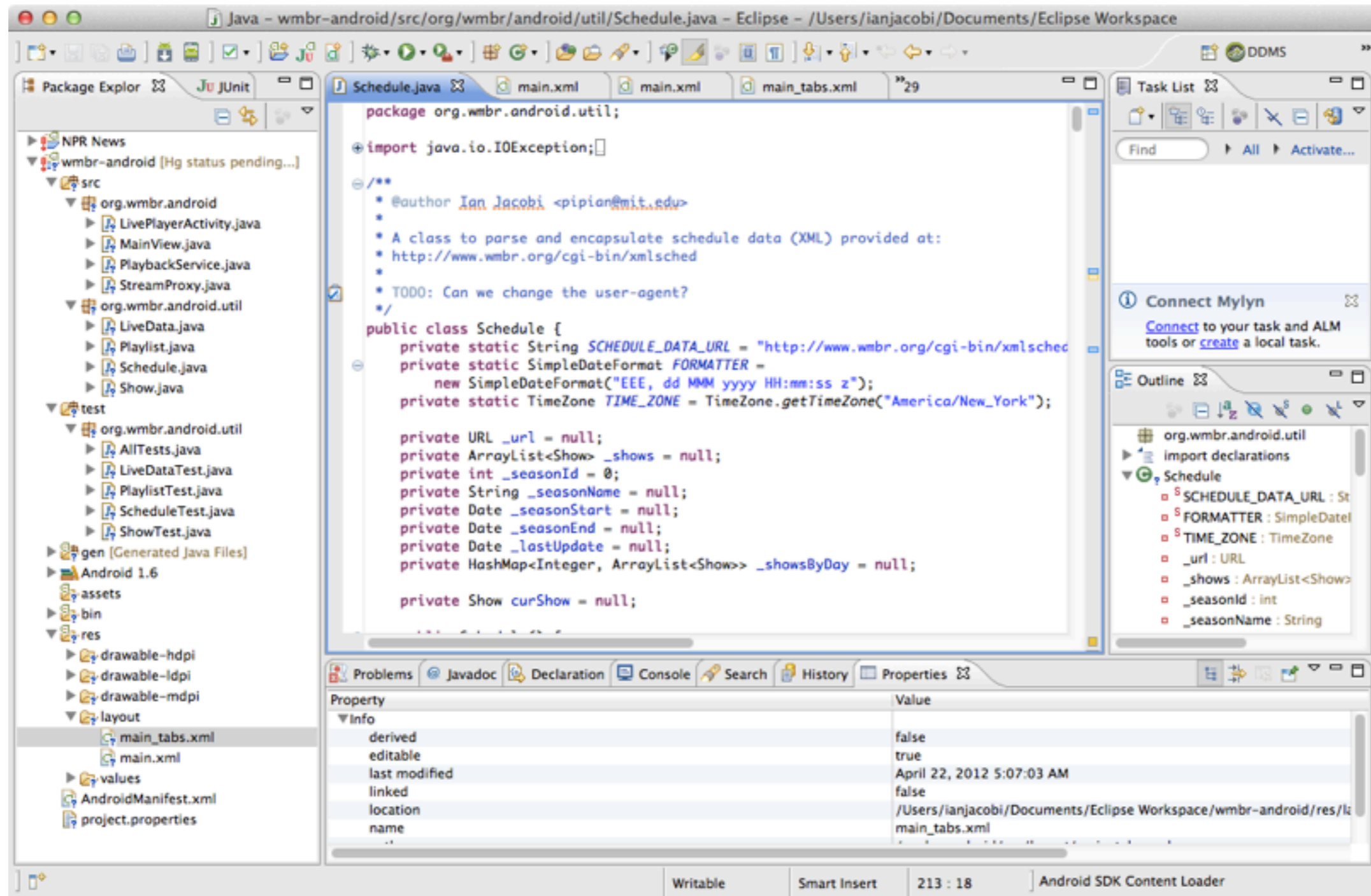
- *Activity* – Single screen
- *Service* – Background actions
- *Content Provider* – Shared persistent storage
- *Broadcast Receiver* – Receives notifications

*An app may have only one type of component,
or it may have all four!*

Other Android App Concepts

- *Intent* – Message to activate a component
- *Manifest* – Describes app, its permissions
- *View* – Something drawn on the screen
 - Includes *layouts, controls, dialogs, etc.*
- *Fragment* – Composable sub-activity
- *Widget* – Mini-view (as in home screen)

Android & Eclipse



Android Versions and API Levels



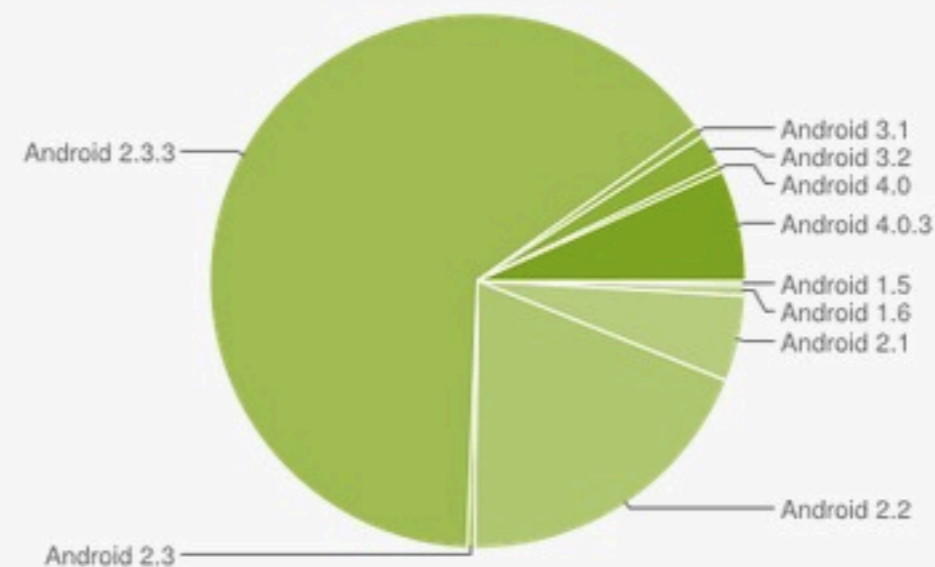
“Dessert plate” by Quinn Dombrowski
Licensed under a Creative Commons Attribution-ShareAlike 2.0 Generic License
<<http://www.flickr.com/photos/quinnanya/5847206255/>>

Android Versions and API Levels

Current Distribution

The following pie chart and table is based on the number of Android devices that have accessed Google Play within a 14-day period ending on the data collection date noted below.

Version	Codename	API Level	Distribution
1.5	Cupcake	3	0.3%
1.6	Donut	4	0.6%
2.1	Eclair	7	5.2%
2.2	Froyo	8	19.1%
2.3 - 2.3.2	Gingerbread	9	0.4%
2.3.3 - 2.3.7		10	64.6%
3.1	Honeycomb	12	0.7%
3.2		13	2%
4.0 - 4.0.2	Ice Cream Sandwich	14	0.4%
4.0.3 - 4.0.4		15	6.7%



From "Android | Dashboards: Platform Versions"
<<http://developer.android.com/about/dashboards/index.html>>

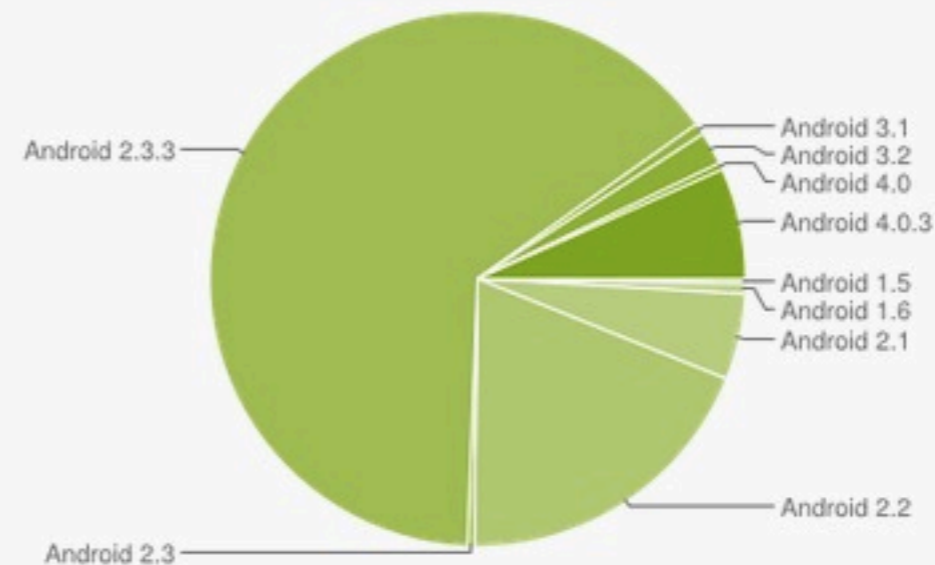
Data collected during a 14-day period ending on June 1, 2012

Android Versions and API Levels

Current Distribution

The following pie chart and table is based on the number of Android devices that have accessed Google Play within a 14-day period ending on the data collection date noted below.

Version	Codename	API Level	Distribution
1.5	Cupcake	3	0.3%
1.6	Donut	4	0.6%
2.1	Eclair	7	5.2%
2.2	Froyo	8	19.1%
2.3 - 2.3.2	Gingerbread	9	0.4%
2.3.3 - 2.3.7		10	64.6%
3.1	Honeycomb	12	0.7%
3.2		13	2%
4.0 - 4.0.2	Ice Cream Sandwich	14	0.4%
4.0.3 - 4.0.4		15	6.7%



From "Android | Dashboards: Platform Versions"
<<http://developer.android.com/about/dashboards/index.html>>

**Most still need 2.1
(API Level 7!)**

Data collected during a 14-day period ending on June 1, 2012

Android Versions and API Levels

public class

Spinner

extends [AbsSpinner](#)

implements [DialogInterface.OnClickListener](#)

Summary: [XML Attrs](#) | [Inherited XML Attrs](#) | [Constants](#) | [Inherited Constants](#) | [Inherited Fields](#) | [Ctors](#) | [Methods](#) | [Protected Methods](#) | [Inherited Methods](#) | [\[Expand All\]](#)

Since: **API Level 1**

[java.lang.Object](#)

↳ [android.view.View](#)

↳ [android.view.ViewGroup](#)

↳ [android.widget.AdapterView<T extends android.widget.Adapter>](#)

↳ [android.widget.AbsSpinner](#)

↳ [android.widget.Spinner](#)

Class Overview

A view that displays one child at a time and lets the user pick among them. The items in the Spinner come from the [Adapter](#) associated with this view.

See the [Spinner tutorial](#).

Summary

XML Attributes

Attribute Name	Related Method	Description
android:dropDownHorizontalOffset	setDropDownHorizontalOffset(int)	Horizontal offset from the spinner widget for positioning the dropdown in

Android Versions and API Levels

public class

Spinner

extends [AbsSpinner](#)

implements [DialogInterface.OnClickListener](#)

Summary: [XML Attrs](#) | [Inherited XML Attrs](#) | [Constants](#) | [Inherited Constants](#) | [Inherited Fields](#) | [Ctors](#) | [Methods](#) | [Protected Methods](#) | [Inherited Methods](#) | [\[Expand All\]](#)

Since: **API Level 1**

[java.lang.Object](#)

↳ [android.view.View](#)

↳ [android.view.ViewGroup](#)

↳ [android.widget.AdapterView<T extends android.widget.Adapter>](#)

↳ [android.widget.AbsSpinner](#)

↳ [android.widget.Spinner](#)

Class Overview

A view that displays one child at a time and lets the user pick among them. The items in the Spinner come from the [Adapter](#) associated with this view.

See the [Spinner tutorial](#).

Summary

XML Attributes

Attribute Name	Related Method	Description
android:dropDownHorizontalOffset	setDropDownHorizontalOffset(int)	Horizontal effect from the spinner widget for positioning the dropdown in

Version Control

What is version control?

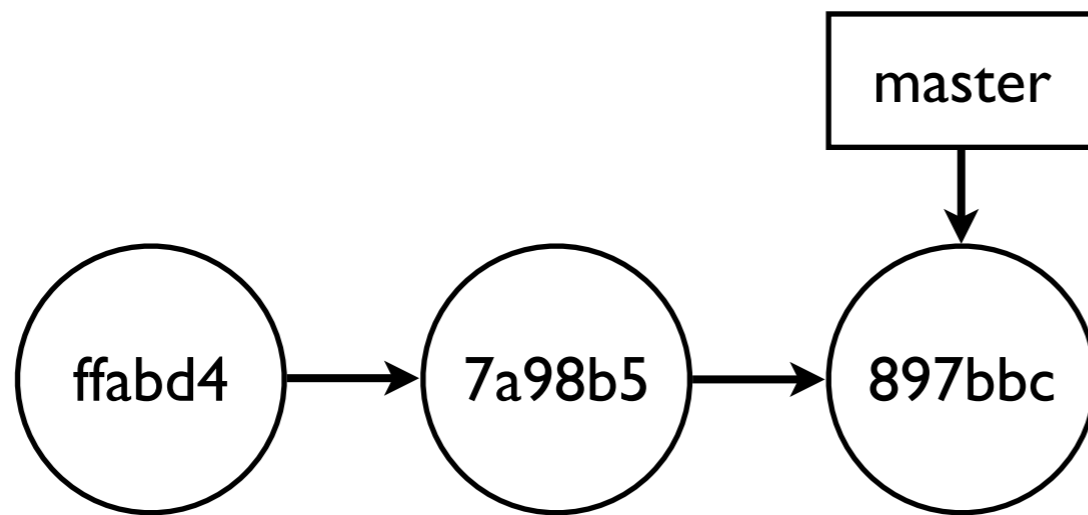
Version control is a way to manage the history of a project's source code.

Types of version control

- Client/Server (CVS, SVN, SourceSafe...)
- Distributed (Git, Mercurial, bazaar...)

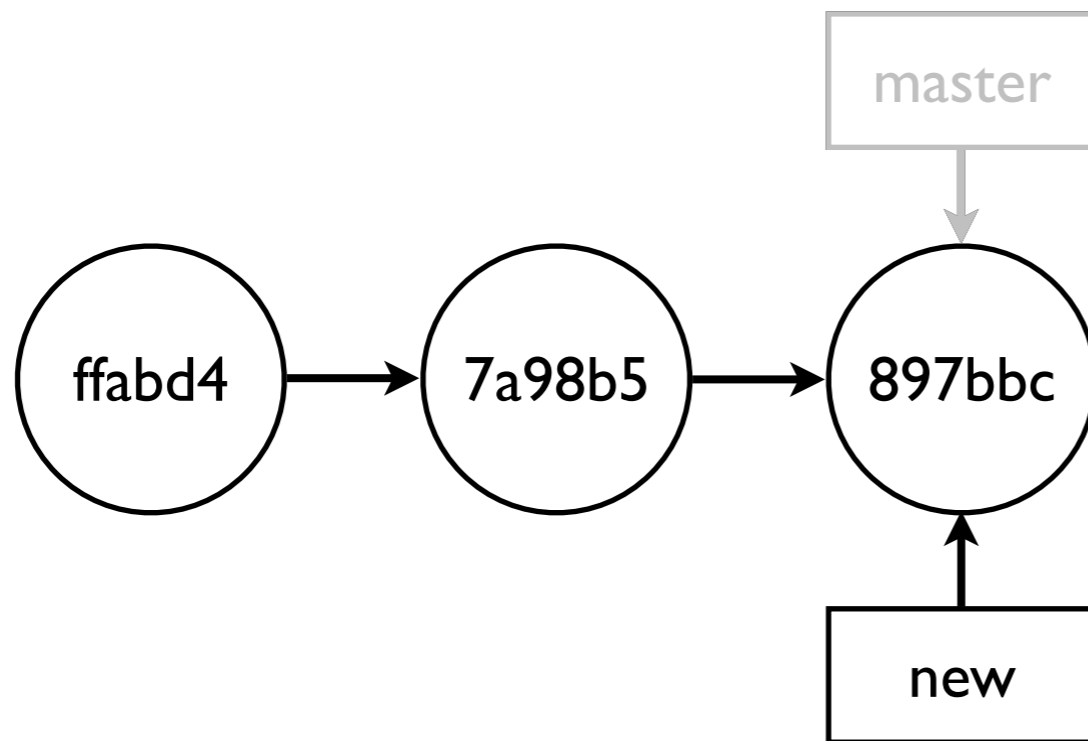
We will use Git in the labs.

Branches



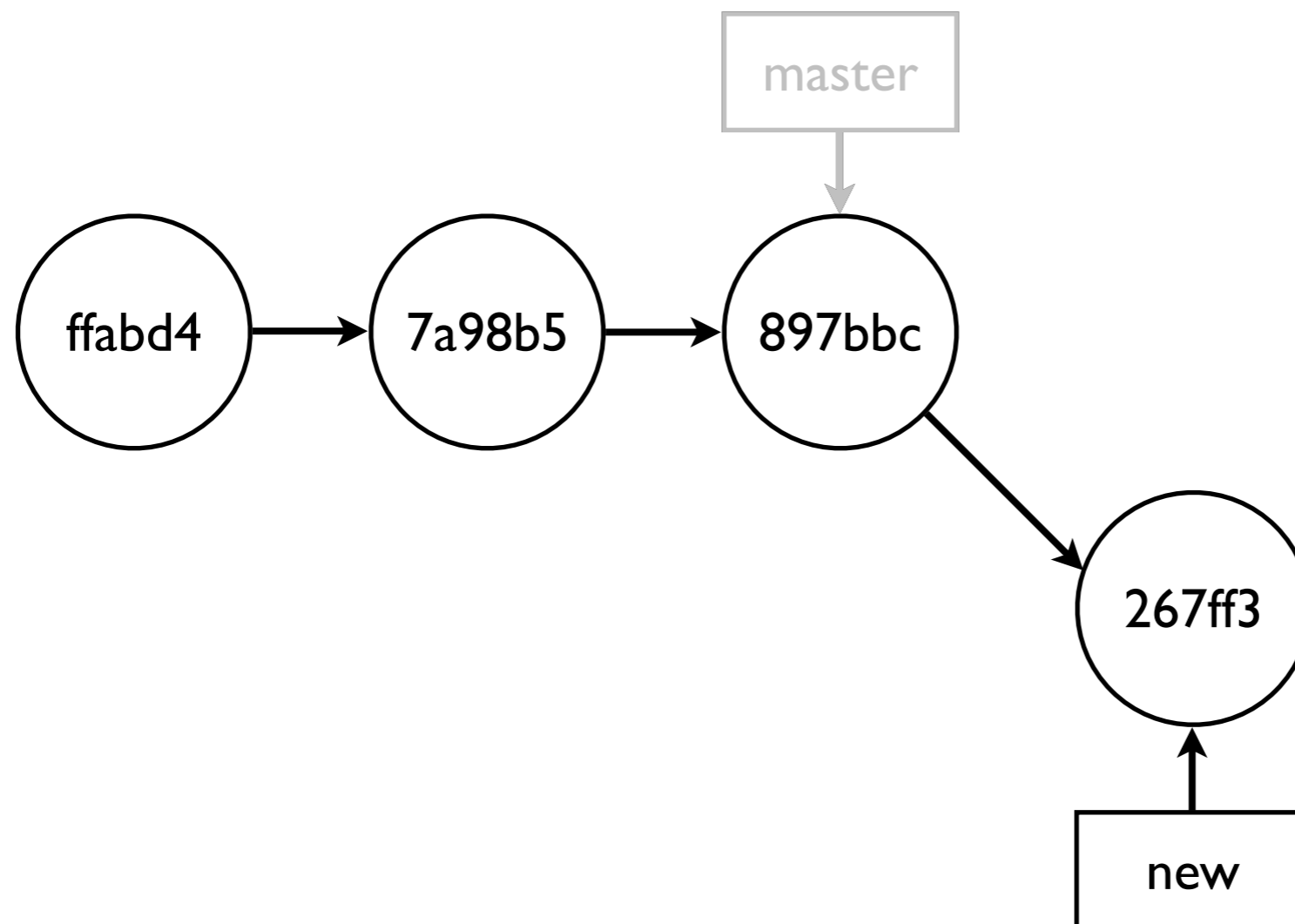
Branches

`git branch new`



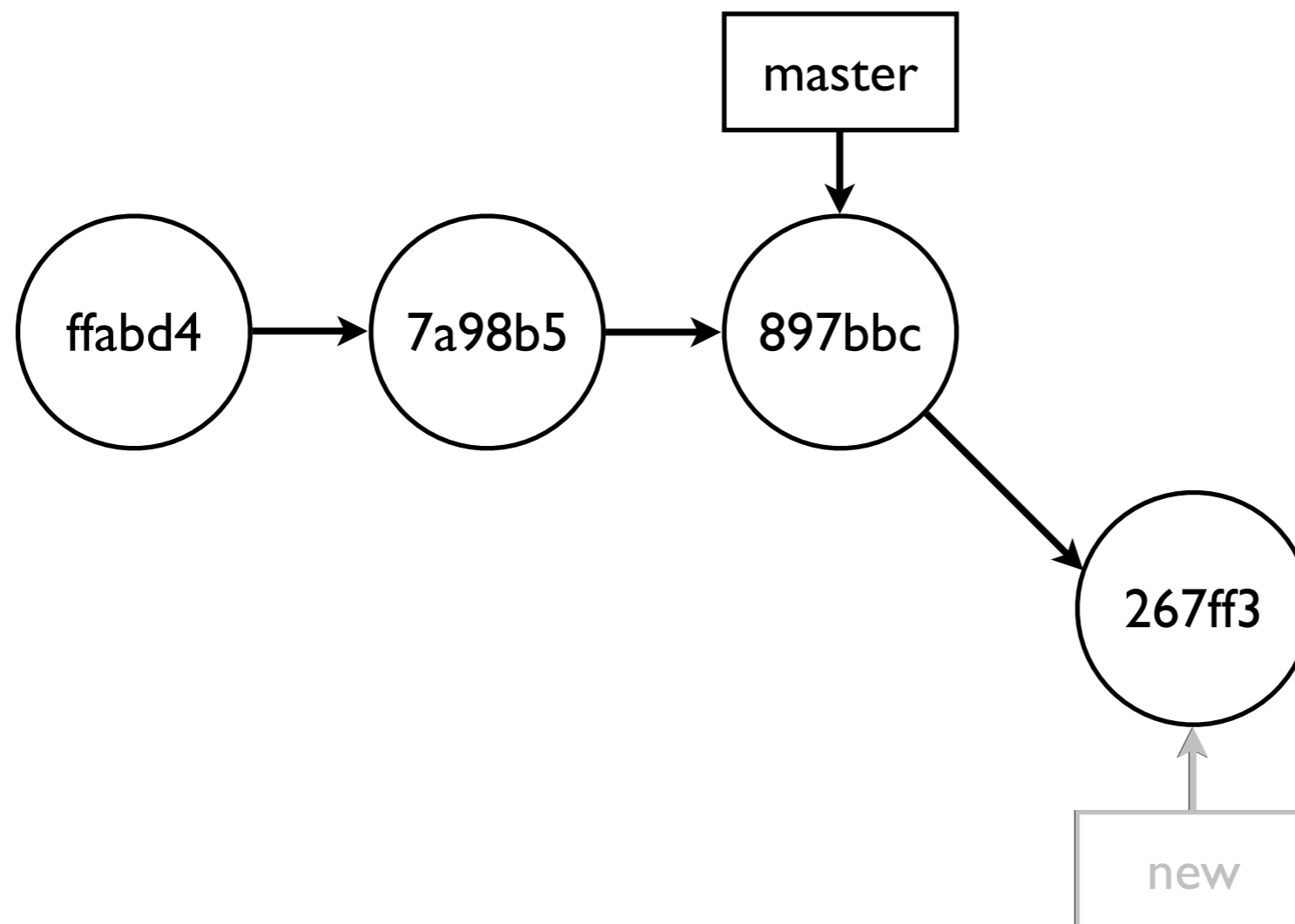
Branches

```
[edit]; git commit
```



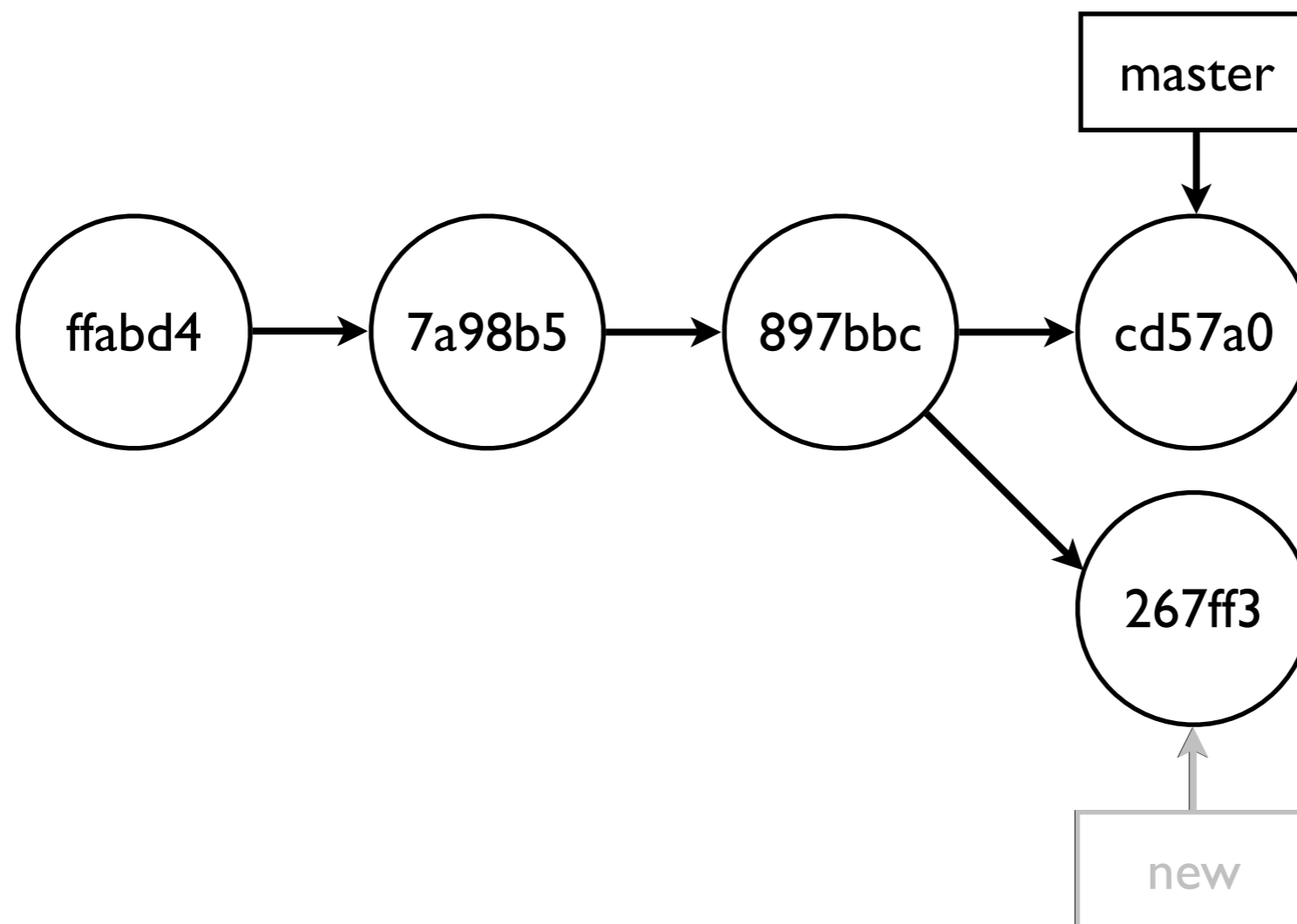
Branches

`git checkout master`



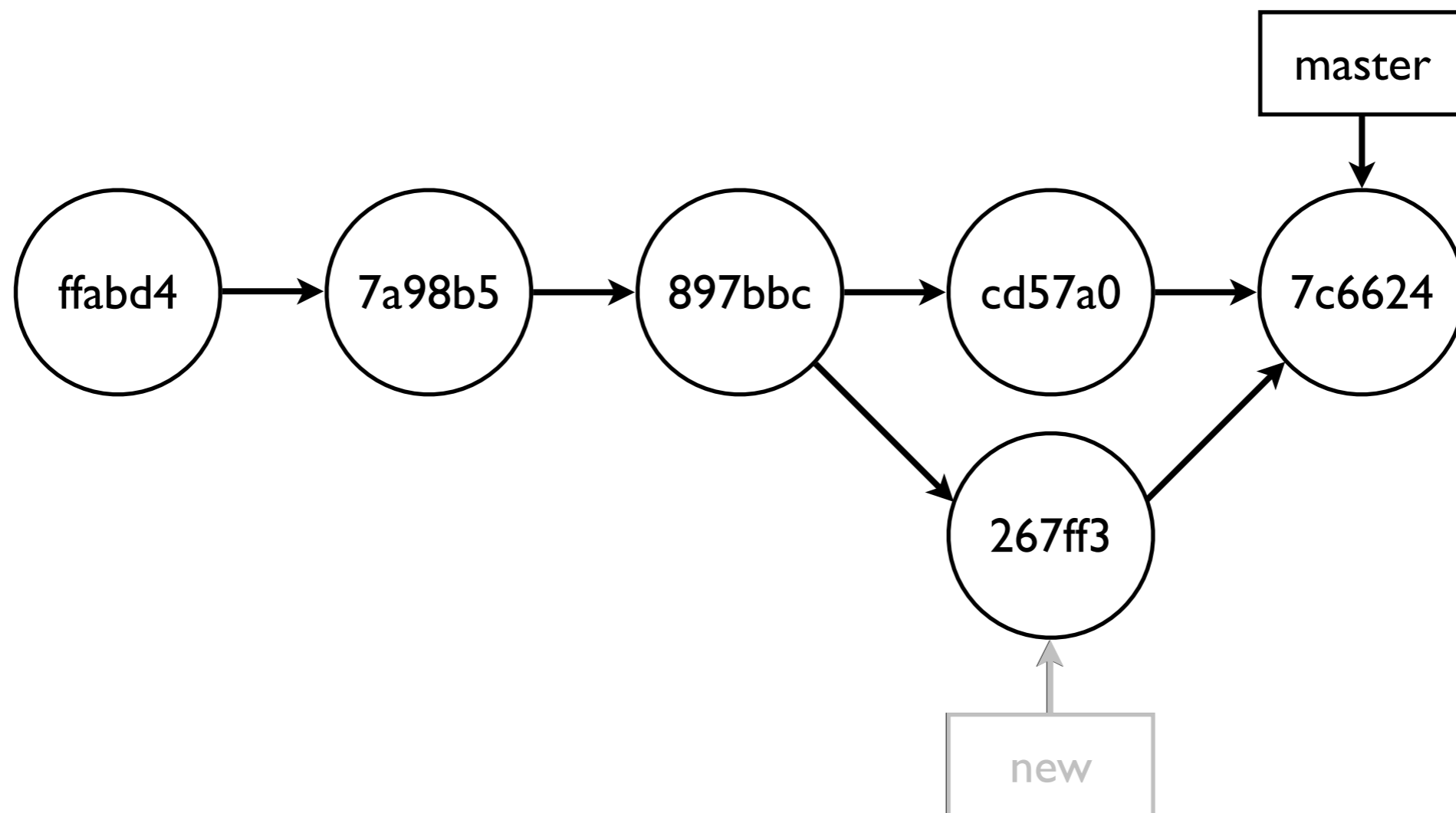
Branches

`[edit]; git commit`



Branches

`git merge new`



Version Control Concepts

- *Revision/Commit* – Saved version of source
- *To commit (verb)* – To save the version
- *Repository* – Where the history is stored

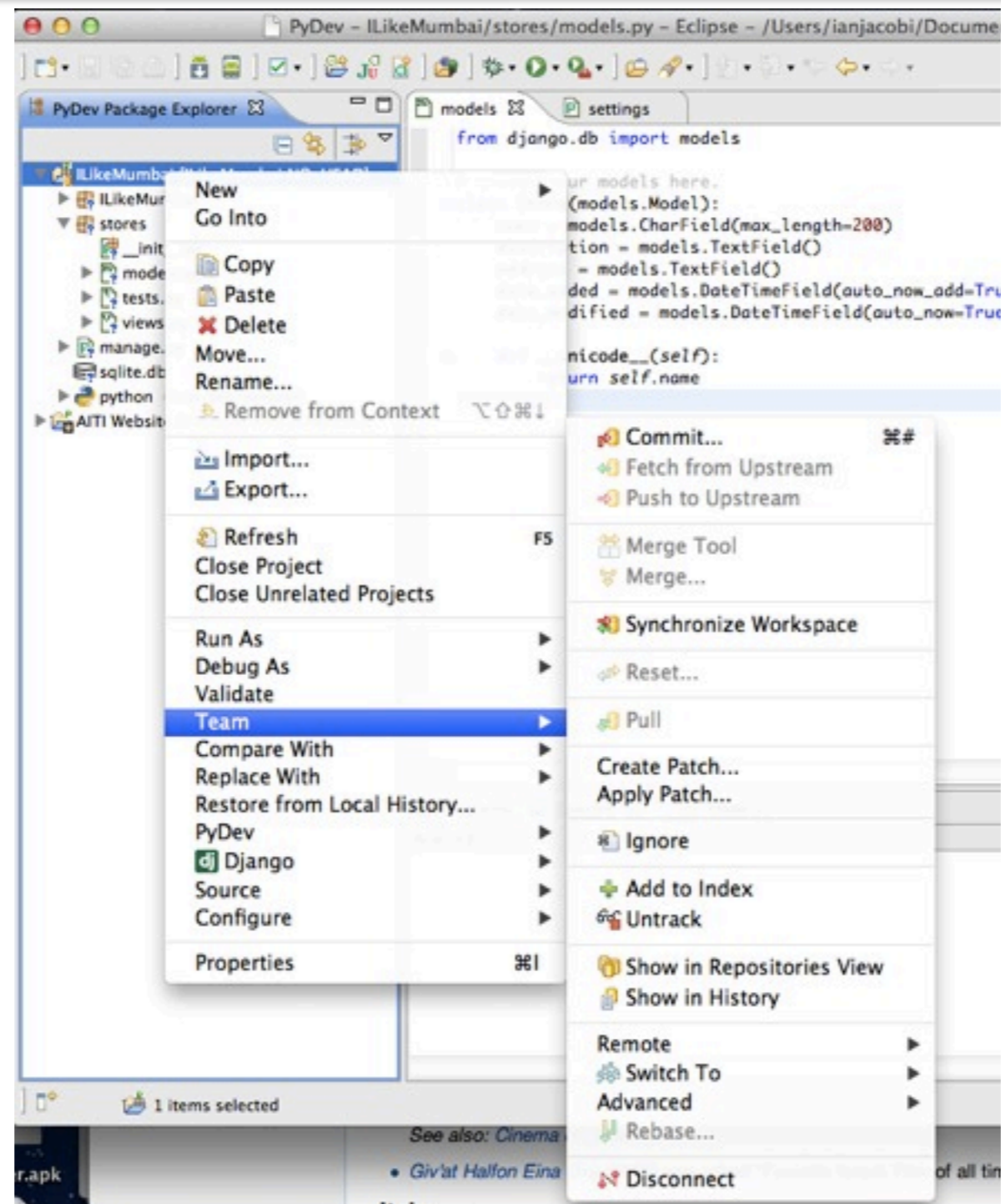
Remote Storage: Bitbucket



Atlassian

bitbucket

Git & Eclipse: EGit



Getting Set Up

- Download Eclipse IDE for Java Developers:
<<http://eclipse.org/downloads/>>
- Install the Android SDK:
<<http://developer.android.com/sdk/installing/>>
- For later: Install Python 2.7.3 and PyDev:
 - Python 2.7.3: <<http://www.python.org/download/>>
 - PyDev: <<http://pydev.org/download.html>>
- How to install Eclipse plugins
(Android; for PyDev, replace URL with the “main” URL listed in above link):
<<http://developer.android.com/sdk/installing/installing-adt.html>>

References

- Android Developer Site (esp. API Guides):
<<http://developer.android.com/>>
- Videos: Java and Eclipse for Total Beginners:
<<http://eclipsetutorial.sourceforge.net/totalbeginner.html>>
- A good Git book: *Pro Git* by Scott Chacon
<<http://git-scm.com/book>>
- Using Bitbucket:
<<https://confluence.atlassian.com/display/BITBUCKET/Bitbucket+101>>
- For more on EGit:
<http://wiki.eclipse.org/EGit/User_Guide>

**Don't forget to sign in
and register when you get
the e-mail!**