Proin

Overview

Google
CONFIDENTIAL

ANDROID

First truly open, freely available Linux-based mobile phone stack built from the ground up

- Open source platform and applications
  - Linux 2.6 kernel and drivers
  - Java language framework and applications
- Complete product
  - Hardware designs
  - Full set of applications
  - Mobile operator certified

Google
CONFIDENTIAL

Hardware Requirements

- **Chipset**: MSM7200A with RTR6285 and PM7540
- **Memory**: at least 128MB RAM, 256MB Flash
- **External Storage**: Mini, Micro SD
- **Primary Display**: QVGA TFT LCD or larger, 16-bit color or better
- **Navigation Keys**: 5-way navigation with 4 application keys, power and volume control
- **Camera**: 2MP CMOS
- **USB**: Standard mini-B USB interface
- **Bluetooth** 1.2 or 2.0

Google
CONFIDENTIAL

Optional Hardware

- QWERTY keyboard
- WiFi
- GPS

Google
CONFIDENTIAL

Project Goals

- Great phone experience
- Great Google experience
- Open application architecture

Google
CONFIDENTIAL

Great Phone

- Multiple processes
  - Phone application must always be responsive
  - Wayward applications can’t crash phone
- Enable task switching
  - Can switch in and out of applications while in a call
  - Applications don’t lose state

Google
CONFIDENTIAL

Google Experience

- Integrated Google applications written with public APIs
- Complete two-way sync for user data
- Open APIs for application functionality

Gmail

GTalk

YouTube

Calendar

Contacts

Picasa

Maps

Earth

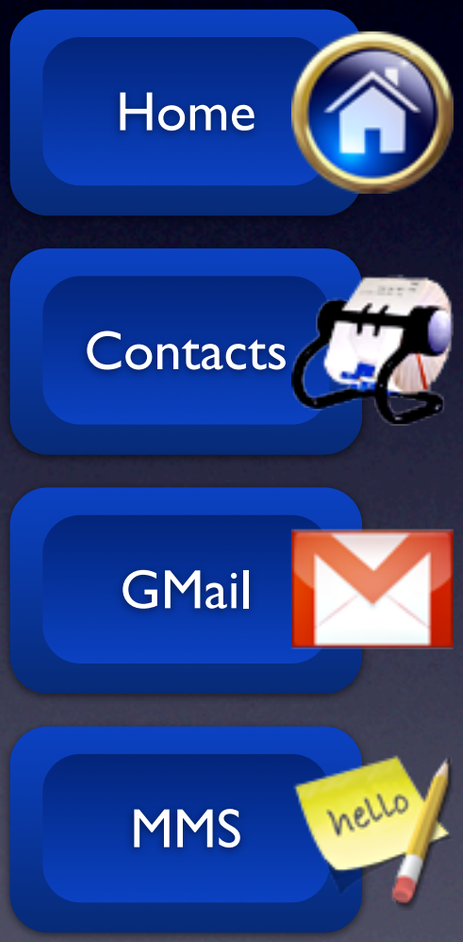
Google
CONFIDENTIAL

Open Architecture

- Powerful, simple application framework written in Java
  - Encourage reuse of components
- Scalable user experience
  - 12 key, QWERTY, touch screen, combinations
  - Portrait, landscape, pivoting, etc
- Rich graphics system & media experience
- Advanced, standards-based web browser

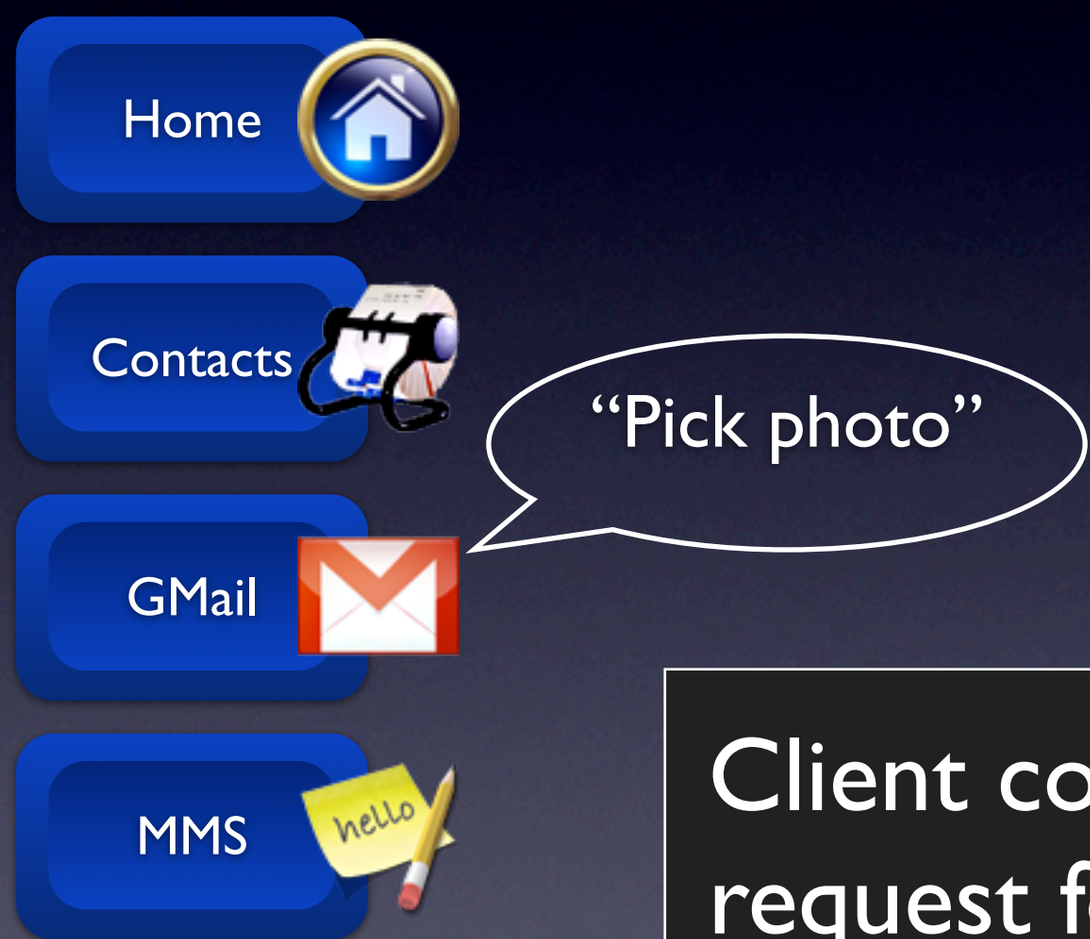
Google
CONFIDENTIAL

Open Architecture



Google
CONFIDENTIAL

Open Architecture



Client component makes a request for a specific action

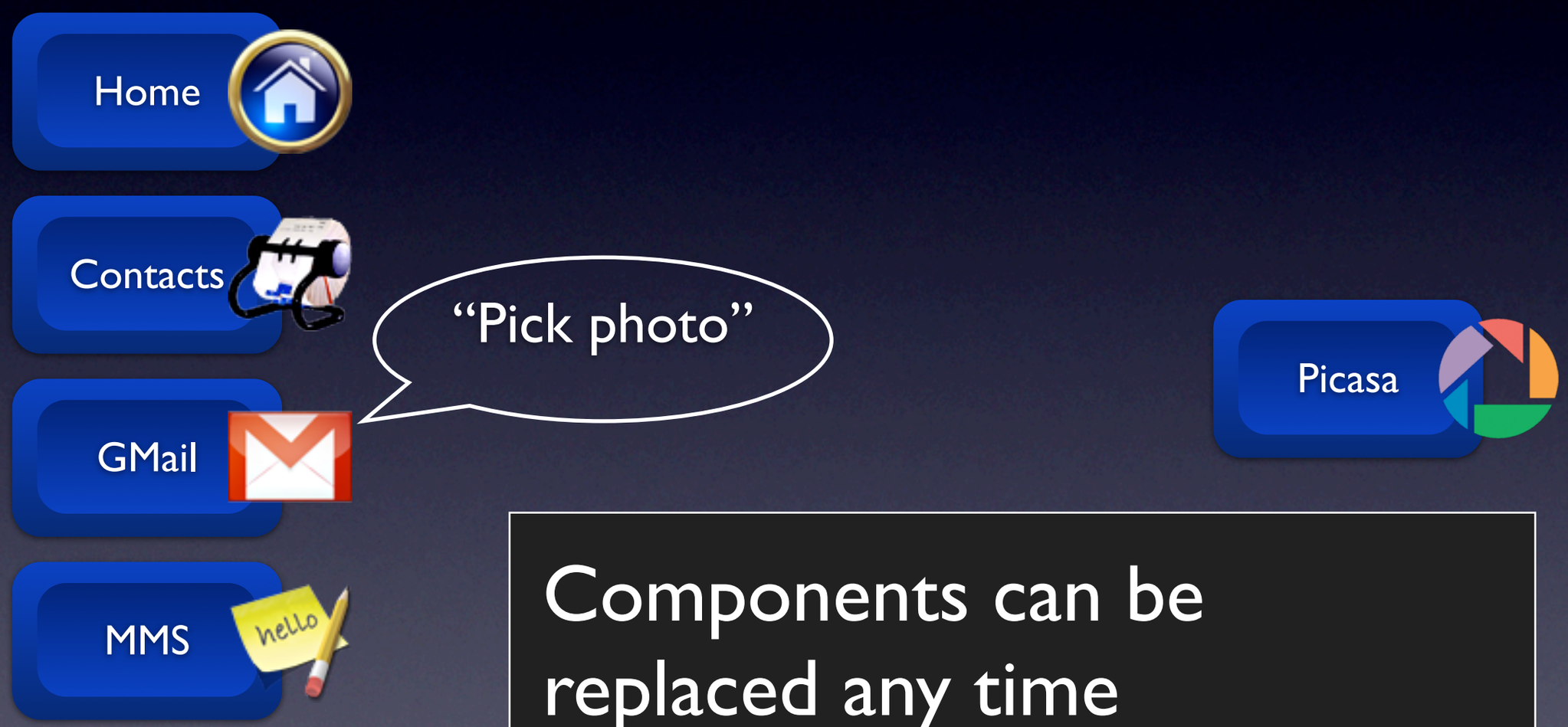
Google
CONFIDENTIAL

Open Architecture

System picks best component for that action

Google
CONFIDENTIAL

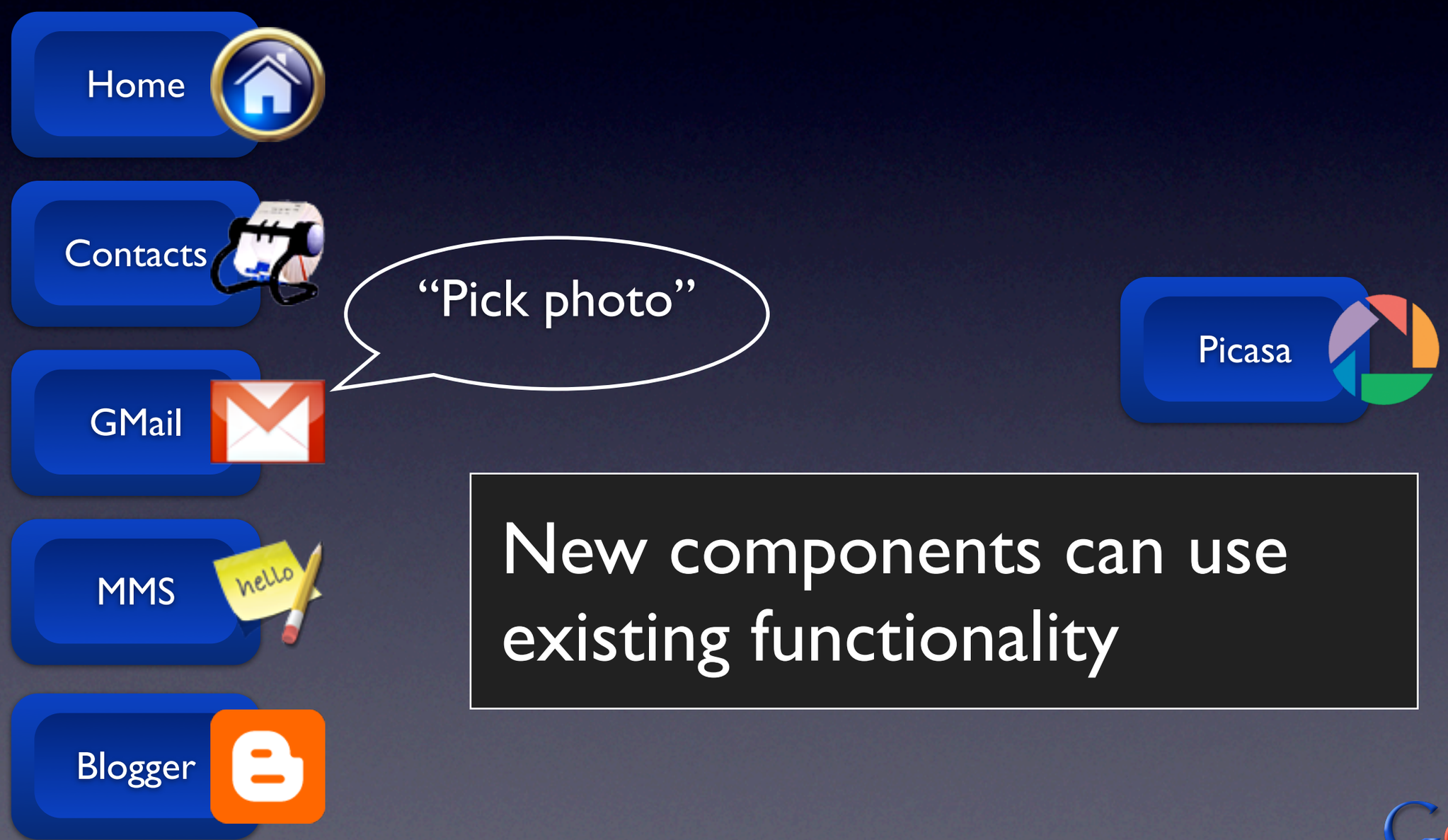
Open Architecture



Components can be replaced any time

Google
CONFIDENTIAL

Open Architecture



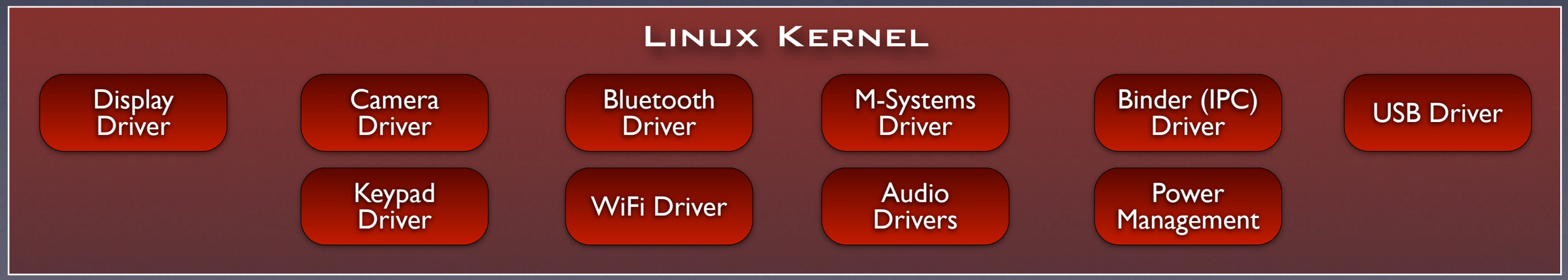
New components can use existing functionality

Google
CONFIDENTIAL

Software Architecture

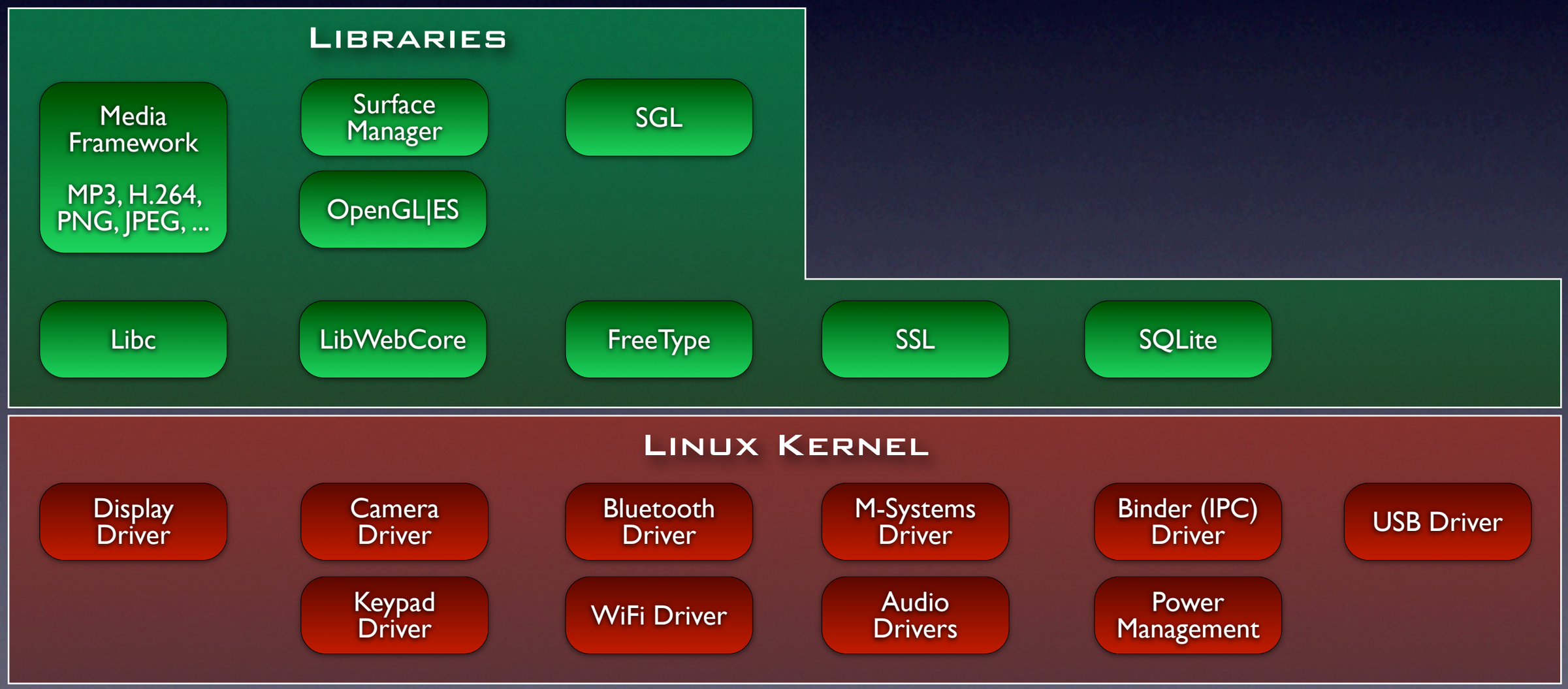
CONFIDENTIAL

Software Architecture



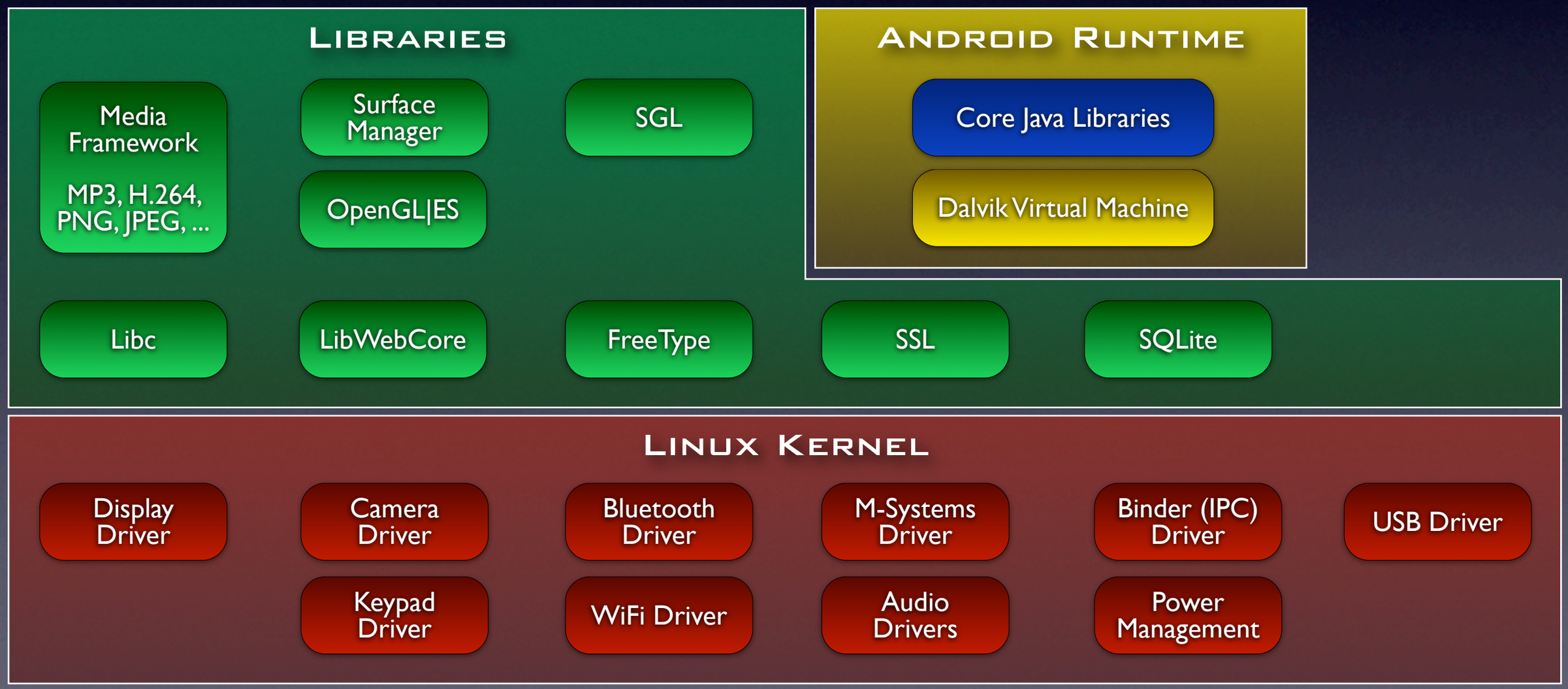
CONFIDENTIAL

Software Architecture



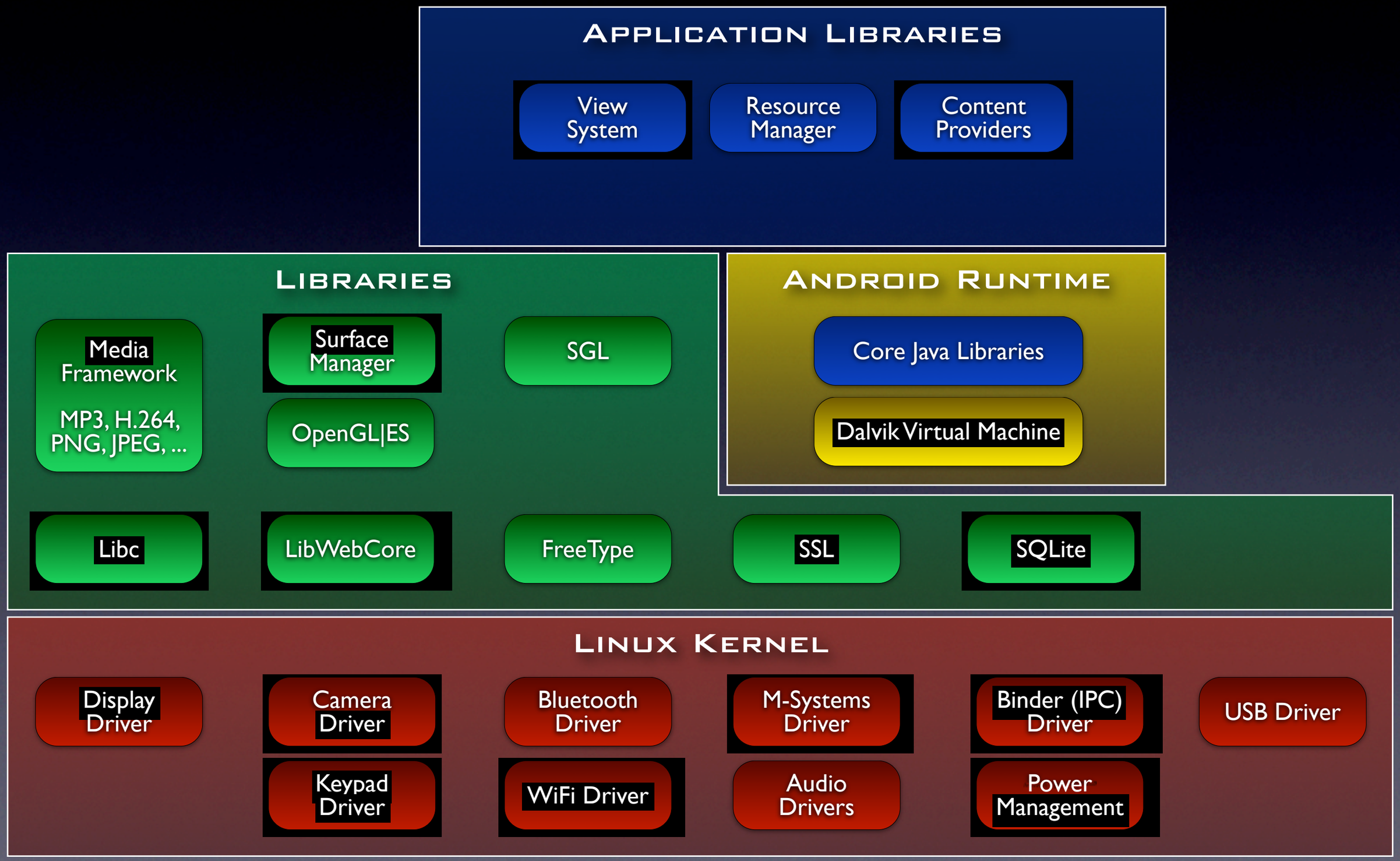
CONFIDENTIAL

Software Architecture

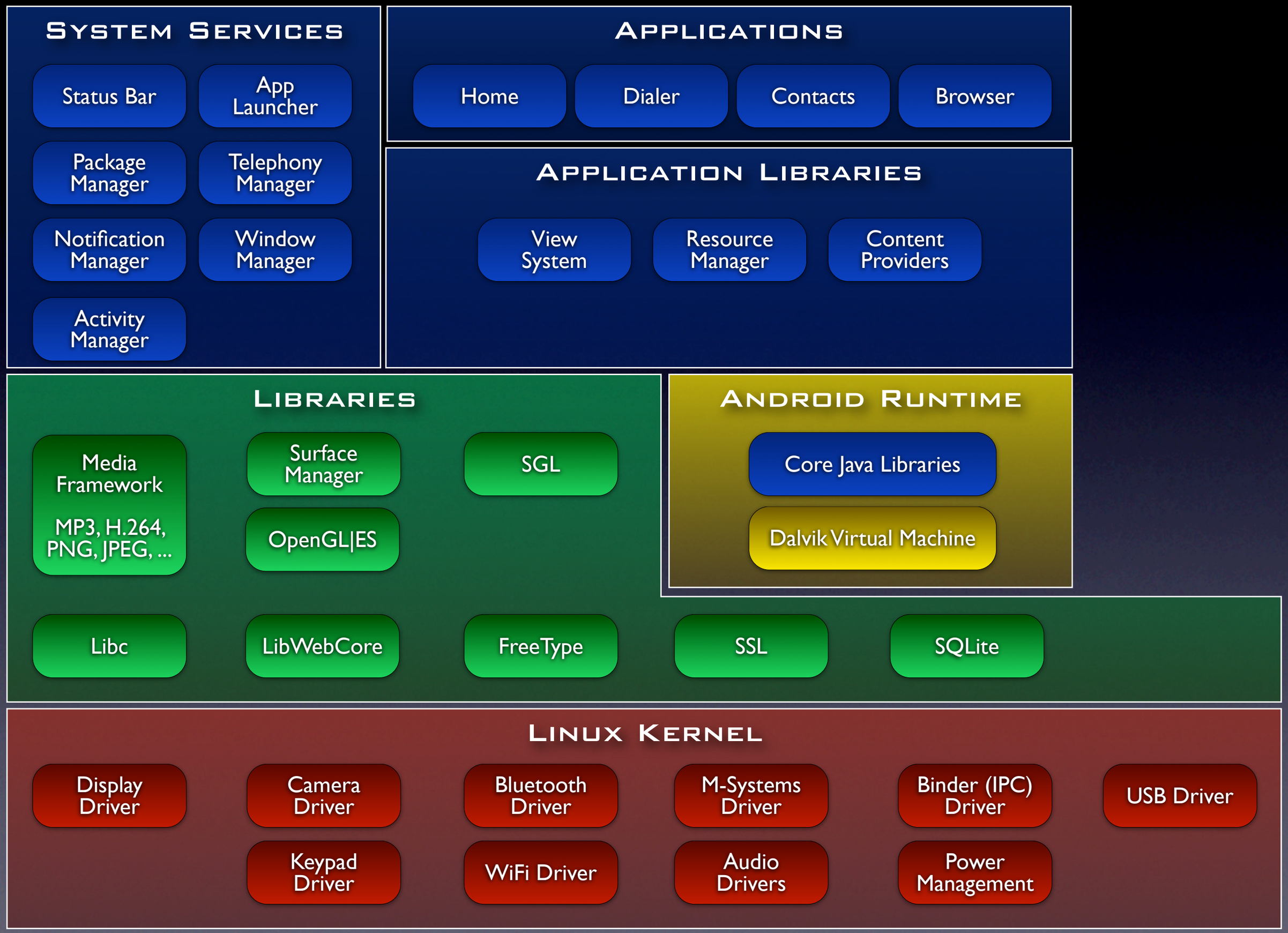


CONFIDENTIAL

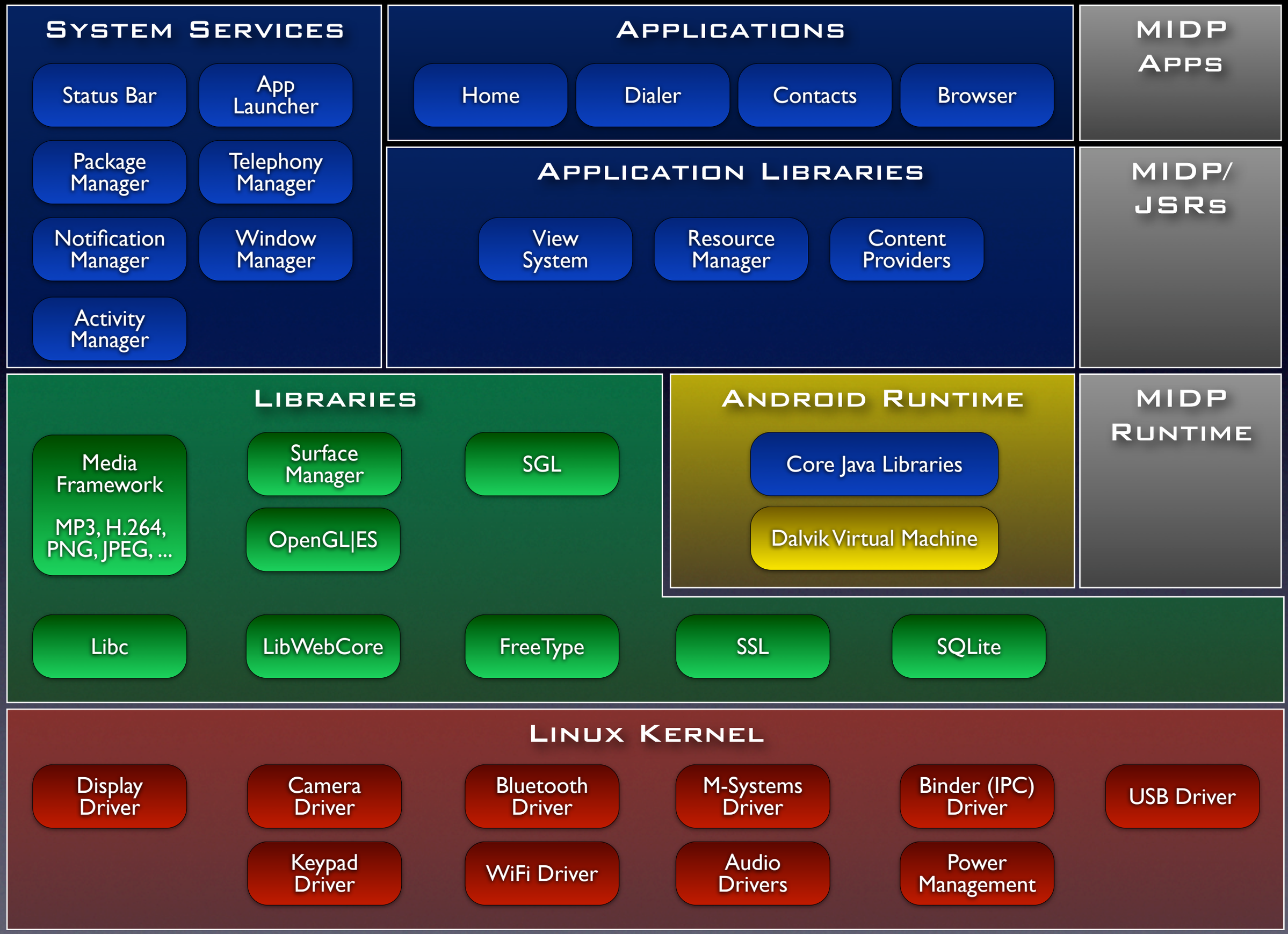
Software Architecture



CONFIDENTIAL



CONFIDENTIAL



CONFIDENTIAL

Processes

- Applications and System services run in separate processes
- Activity Manager runs each application in a separate process and manages application life cycle
- Boot process responsible for system startup
- System process hosts persistent core system services
  - *Window Manager, Surface Manager, Activity Manager, etc*
- Application processes can also be persistent
  - *Dialer application*

Google
CONFIDENTIAL

Processes

Google
CONFIDENTIAL

Application Framework

- **Window Manager**
  - Allows multiple apps with multiple windows to share screen
- **Package Manager**
  - Maintains information for all installed applications
- **Activity Manager**
  - Manages lifecycle of application UI components
  - Maintains cross-application history

Google
CONFIDENTIAL

Application Framework

- **Notification Manager**
  - Allows all applications access to status bar and onscreen notifications
- **Resource Manager**
  - Provide access to graphics, localized strings, XML layout descriptions, etc
- **View System**
  - Rich set of built-in controls (buttons, text, lists, etc)
  - Supports diverse screen sizes

Google
CONFIDENTIAL

Dalvik

- Android's custom Java-compatible virtual machine
- Runs optimized file format (.dex)
- Also runs unmodified Java .class / .jar files
- Designed for embedded environment
  - Uses runtime memory very efficiently
  - Highly CPU-optimized bytecode interpreter
  - Supports multiple virtual machine processes per device

Google
CONFIDENTIAL

Dalvik

- Custom file format (.dex) designed for runtime efficiency
  - Shared constant pool reduces memory
  - Read-only structures improves cross-process sharing
  - Concise, fixed-width structures reduces parse time
  - Installed apps are translated to .dex at build time
- Bytecodes designed for speedy interpretation
  - Register-based rather than stack-based instructions reduce memory I/O and dispatch overhead
  - Only fixed-width instructions simplifies parsing
  - 16 bit code units minimizes reads

Google
CONFIDENTIAL

Telephony Manager

- Application level Java interface to telephony sub-system
- Event notifications
  - *Phone state, network state, SIM status, voicemail, etc.*
- Access to state information
  - *Network info, SIM info, voicemail presence, etc.*
- Initiate calls
- Query and control call state

Google
CONFIDENTIAL

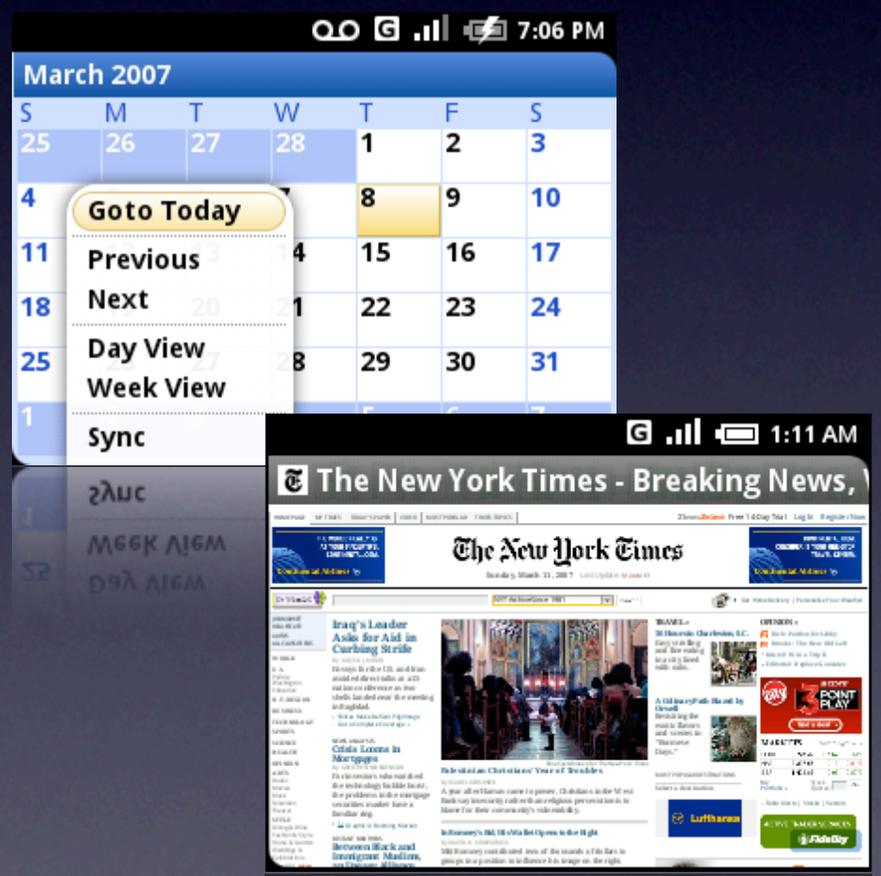
Browser

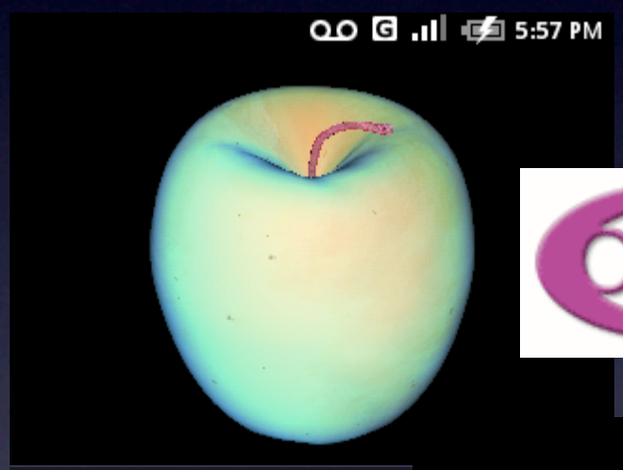
- Based on open source KHTML "WebKit" browser
- Active collaboration with Apple and industry partners
- Renders pages in full, desktop-like manner with navigation
- Allows single column, small screen rendering as well
- HTML views can be embedded in other applications

Google
CONFIDENTIAL

Graphics

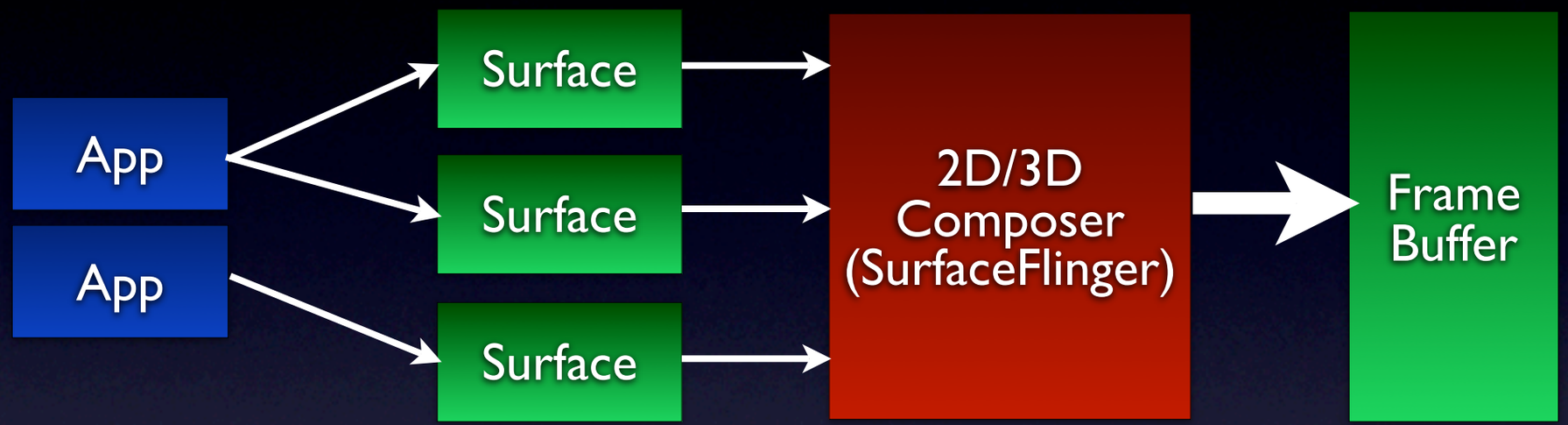
- SGL Library for 2D application drawing
- OpenGL|ES for gaming, 3D





Google
CONFIDENTIAL

Graphics



- SurfaceFlinger is system-wide “composer”
- Manages Surfaces and the Frame-buffer
- Handles window transitions
- Uses OpenGL|ES and 2D hardware accelerator for its compositions

Google
CONFIDENTIAL

Media Framework

- Based on PacketVideo source code license
- Supports standard video, audio, still-frame formats
  - MPEG4, H.264, MP3, AAC, AMR, JPEG
- Advanced framework with simple Java API layer

Google
CONFIDENTIAL

Bluetooth

- Based on BlueZ (3.8)
- Supports 2.0
- Profile support
  - Headset and Hands-Free
  - Dial-Up Networking and PAN
  - Audio Streaming (A2DP/AVRCP)
- Java bindings for scanning, pairing/unpairing, service queries, etc.

Google
CONFIDENTIAL

Android Tools

- Compatible with standard IDEs
  - Eclipse, IntelliJ
- Android-specific tools to increase visibility into what is happening on device
  - Full support for on-device debugging
  - Performance & memory profiling

Google
CONFIDENTIAL

Application Architecture

Google
CONFIDENTIAL

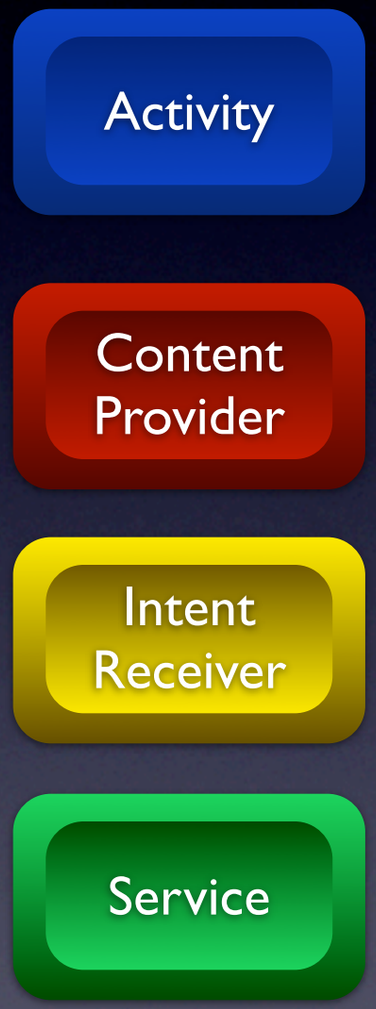
Design Goals

- Open platform
  - Anything we can do they can do
  - Encourage reuse of components
  - Encourage sharing data
- Work well on an embedded device
  - Constrained memory, battery, CPU, network, screen

Google
CONFIDENTIAL

Android Applications

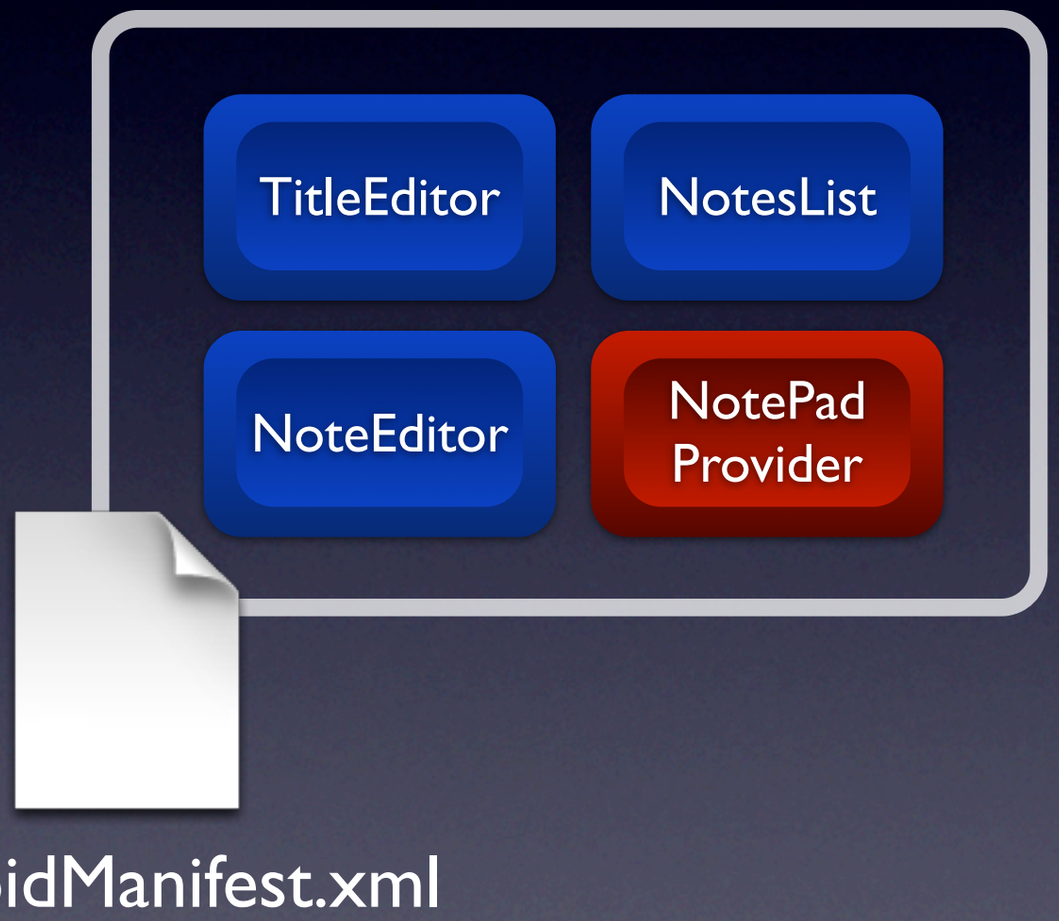
- Built out of 4 main building blocks:
  - Activity
  - ContentProvider
  - IntentReceiver
  - Service



Google
CONFIDENTIAL

AndroidManifest.xml

- Lists all of the components of your application
- Describes capabilities and behaviors of each component



samples/Notes/AndroidManifest.xml

Google
CONFIDENTIAL

Application Lifecycle

- Applications run in their own processes
  - Many Activities, Services, etc can run in the same process
- Processes are started and stopped as needed to run an application's components
- Processes may be killed to reclaim resources

Google
CONFIDENTIAL

Application Lifecycle

- Each application is assigned its own process (whose name is the application's package name)
- Individual parts of an application can be assigned another process name
- Activities and content providers can instead run in the process of the code that is invoking them

Google
CONFIDENTIAL

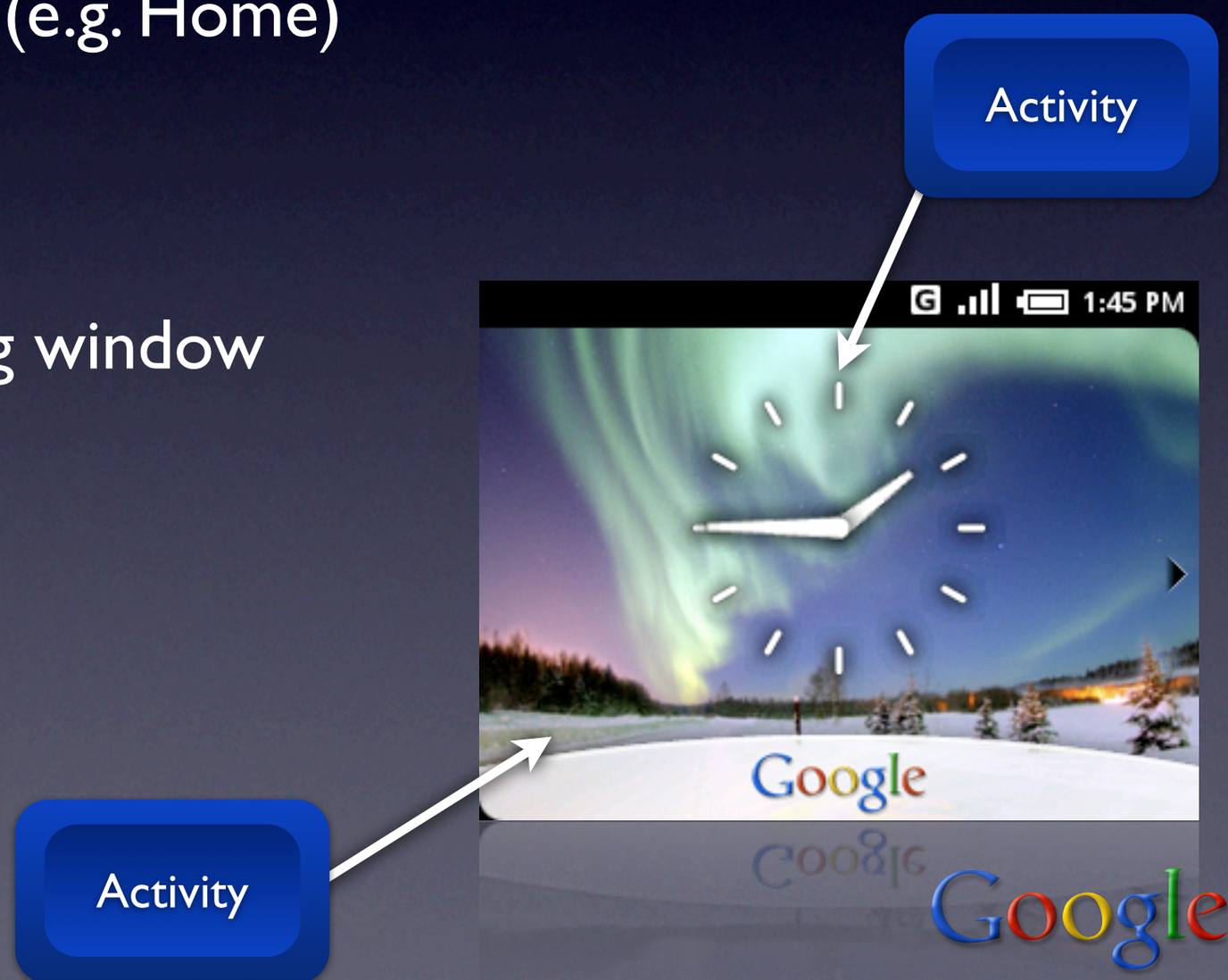
Application Lifecycle

- Activities, IntentReceivers, Services run in main thread
  - Can spawn other threads if needed
  - Be careful about thread safety of platform classes
- ContentProviders are always multi-threaded

Google
CONFIDENTIAL

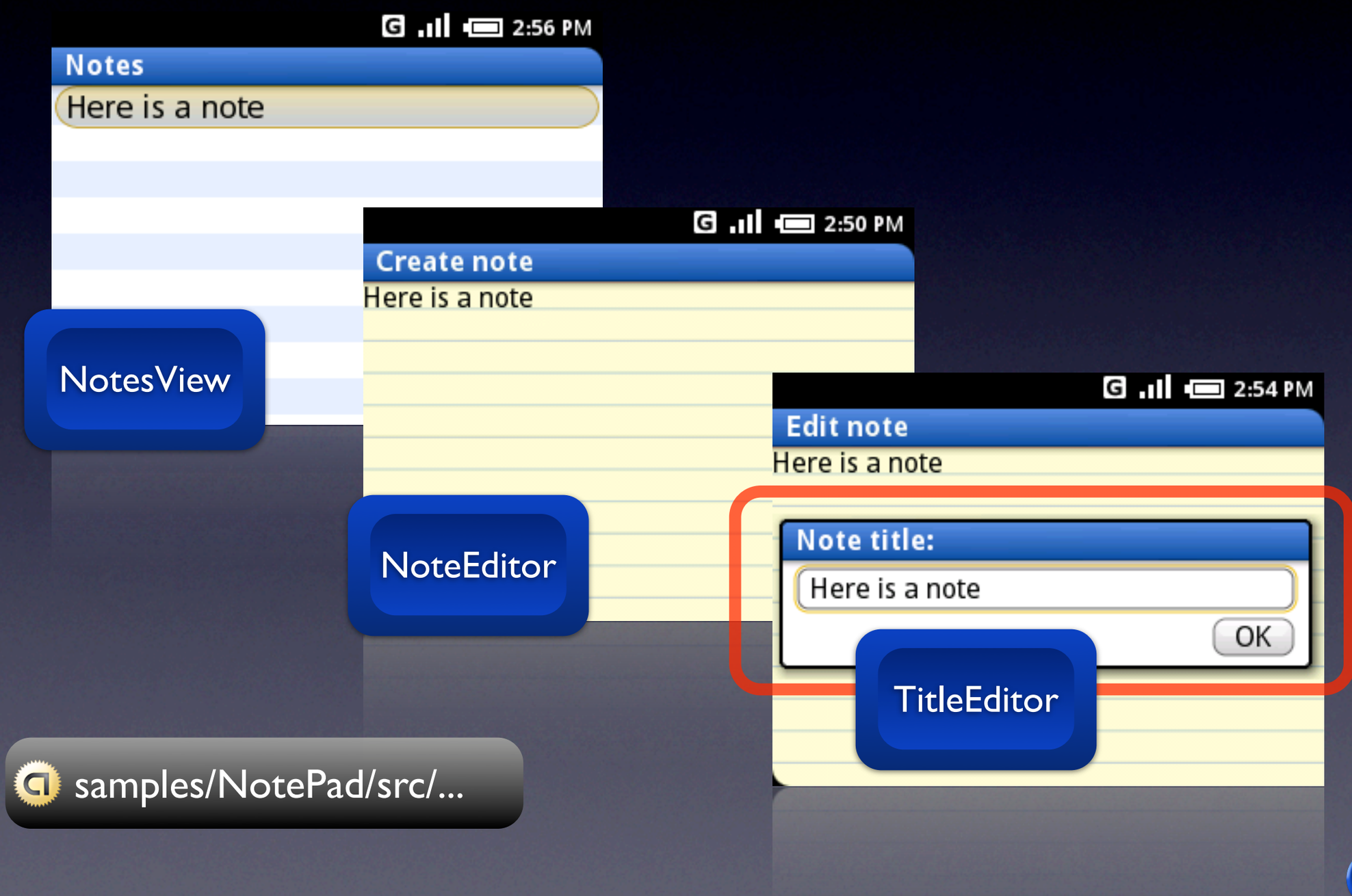
Activities

- Typically corresponds to one screen in the UI
  - Can be embedded (e.g. Home)
  - Can be faceless
  - Can be in a floating window
  - Can return a value



Google
CONFIDENTIAL

Sample Activities



Google
CONFIDENTIAL

Activities

- Designed to be reused and replaced
  - Your application can invoke Activities from another application
  - Another application can invoke your Activities
- Existing Activities can be replaced with a new Activity that fulfills the same contract

Google
CONFIDENTIAL

Activity Lifecycle

- Moving between activities pauses the current one then starts/resumes the next
- Navigation facilitated by a common backstack:
  - Application and process boundaries are invisible to users
  - Activities can be frozen and thawed to conserve system resources

Google
CONFIDENTIAL

Intents & IntentFilters

- Intents: description of what you want done
- IntentFilter: what a piece of code (Activity or IntentReceiver) can do
- Activities publish their IntentFilters in AndroidManifest.xml

Google
CONFIDENTIAL

Intents & IntentFilters

- Forward navigation is accomplished by resolving Intents
  - Caller calls startActivity(myIntent)
  - System picks Activity whose IntentFilter best matches myIntent
  - New Activity is informed of the Intent which caused it to be launched

Google
CONFIDENTIAL

Intent Details

- Action: what to do
  - *MAIN*, *PICK*, *EDIT*, ...
- Data: data to act upon (expressed as URI)
  - Action + Data defines contract between caller and callee
- Category: additional detail about the action
- Extras: name/value pairs
- ClassName: force a specific Activity class to be used

Google
CONFIDENTIAL

ContentProviders

- Provide uniform APIs to:
  - query data (returns a Cursor)
  - delete, update, and insert rows
- Work across processes
- Customers of ContentProviders don't use them directly. ContentResolver acts as clearinghouse.

Google
CONFIDENTIAL

ContentProviders

- All content is represented by URIs
  - Convenience methods mean clients don't need to know syntax
- ContentProviders own URIs based on authority, e.g. content://contacts/...
- ContentProviders are responsible for mapping URIs they own to a MIME type

Google
CONFIDENTIAL

ContentProviders

- Example: *content://note_pad/notes*
  - Represents list of all notes
  - Owned by NotePadProvider
  - MIME type: vnd.android-dir/note

NotePad Provider

samples/NotePad/src/com/google/android/notepad/NotepadProvider.java

Google
CONFIDENTIAL

ContentProviders

- Example: *content://note_pad/notes/6*
  - Represents one particular note whose id is 6
  - Owned by NotePadProvider
  - MIME type: vnd.android/note

NotePad Provider

samples/NotePad/src/com/google/android/notepad/NotepadProvider.java

Google
CONFIDENTIAL

ContentProviders

- Example: *content://contacts/people/6/phones*
  - Represents the list of all phones for a particular person
  - Owned by ContactsProvider
  - MIME type: vnd.android-dir/phone

Google
CONFIDENTIAL

Cursor

- Provides random read/write access to a result set returned from a database query
  - count(), move(), moveTo(), next(), prev()
- Can get and set columns
  - getString(), getInt(), getLong()
  - updateString(), updateInt(), updateLong()

Google
CONFIDENTIAL

Cursor

- Can be bound to ListView, GridView, Gallery, Spinner, etc for display
- Client provides an Adapter which knows how to make a View for each row in Cursor
- Changes are visible across processes

Google
CONFIDENTIAL

IntentReceivers

- Code to run when an Intent is broadcast
- Will wake up process if it is not already running
- Can register in AndroidManifest.xml
- Can also register from code: Context.registerReceiver()

Intent Receiver

samples/SampleCode/src/com/google/android/samples/app/Alarm*.java

Google
CONFIDENTIAL

IntentReceivers

- Useful for responding to alarms and notifications of status change
- Can start Services for lengthy tasks (e.g. downloading new data)

Google
CONFIDENTIAL

Services

- Faceless classes that run in the background, for example:
  - Music player
  - Network download
- Can run in your application's process or their own process

Service

samples/SampleCode/src/com/google/android/samples/app/*Service*.java

Google
CONFIDENTIAL

Services

- System will try to keep processes with running services alive for as long as possible
- Can connect to Services in your process or another process
  - bindService(String className, ServiceConnection conn)

Google
CONFIDENTIAL

RSS Example

Rss Activity	Displays RSS data to user
Rss Provider	Stores downloaded data
RssIntent Receiver	Wakes up periodically to fetch new data using RssService
Rss Service	Runs in background to download RSS data

Google
CONFIDENTIAL

Application APIs

Google CONFIDENTIAL

Application APIs

- Resources
  - Drawables
- Views
  - Layouts & LayoutParams
  - AdapterViews
  - Animation
- Graphics
  - SGL
  - OpenGL|ES
- ContentResolver
- Storage
  - Raw Files
  - Preferences
  - Databases
- Alarm Manager
- Notification Manager
  - Transient Notifications
  - PersistnetNotifications

Google
CONFIDENTIAL

Resources

- Each application can access it's own resource directory as well as system resources
  - Strings, graphics, layouts, animations, etc.
  - Resource manager will pick correct version from alternatives based on language, aspect ratio, etc.

Google
CONFIDENTIAL

Resources

- Resource set is detected at build time by scanning source director
- Each resource is assigned a unique ID
- Tool generates java constants for you:
  - Local resource: R.layout.foo
  - System resource: android.R.layout.bar

Google
CONFIDENTIAL

Resource Examples

Java constant	XML attribute	Resource
R.layout. title_editor	@layout/ title_editor	layout XML file
R.drawable. lander_plain	@drawable/ lander_plain	
R.string. menu_edit	@string/ menu_edit	“Edit note”
R.drawable. earthrise	@drawable/ earthrise	

Google
CONFIDENTIAL

Drawables

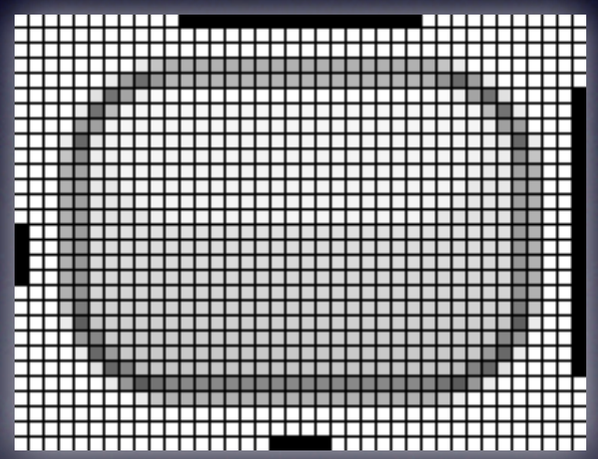
- Base class for something which can fill a rectangle on screen
- Usually loaded from resources
  - Bitmaps: res/drawable/*.png, *.jpg
  - Stretchable Bitmaps: res/drawable/*.9.png
  - Colors: defined in res/values/colors.xml

Google
CONFIDENTIAL

Stretchable Bitmaps

Horizontal stretch

Vertical stretch



Vertical padding

Horizontal padding

Button background can stretch to accommodate any content

Google
CONFIDENTIAL

Views

- Basic user interface building block:
  - TextView, EditText, Button, ImageView, Checkbox, etc
- Each activity has a tree of views
  - Views can be inflated from XML
  - Views can be instantiated from code
- Views responsible for handling events, drawing, measuring

Google
CONFIDENTIAL

Views

- Basic cycle:
  - Event is dispatched
  - View state is updated
  - Framework updates Views layout (if necessary)
  - Framework draws invalid region (if any)
- Let the framework position and draw Views for you

Google
CONFIDENTIAL

Views

- View tree is single-threaded
- Use Handler.post to invoke a Runnable on the event thread

Google
CONFIDENTIAL

Views

- Views with children are called ViewGroups
- ViewGroups are responsible for laying out their children
- ViewGroups route events to child that has focus
- Drawing is an in-order traveral:
  - Parents before children; siblings in order

Google
CONFIDENTIAL

Layouts

- FrameLayout: Each child is a layer
- LinearLayout: Arranges children in a single row or column
- RelativeLayout: Size and position are described relative to parent and other views
- AbsoluteLayout: Views given specific <x, y>

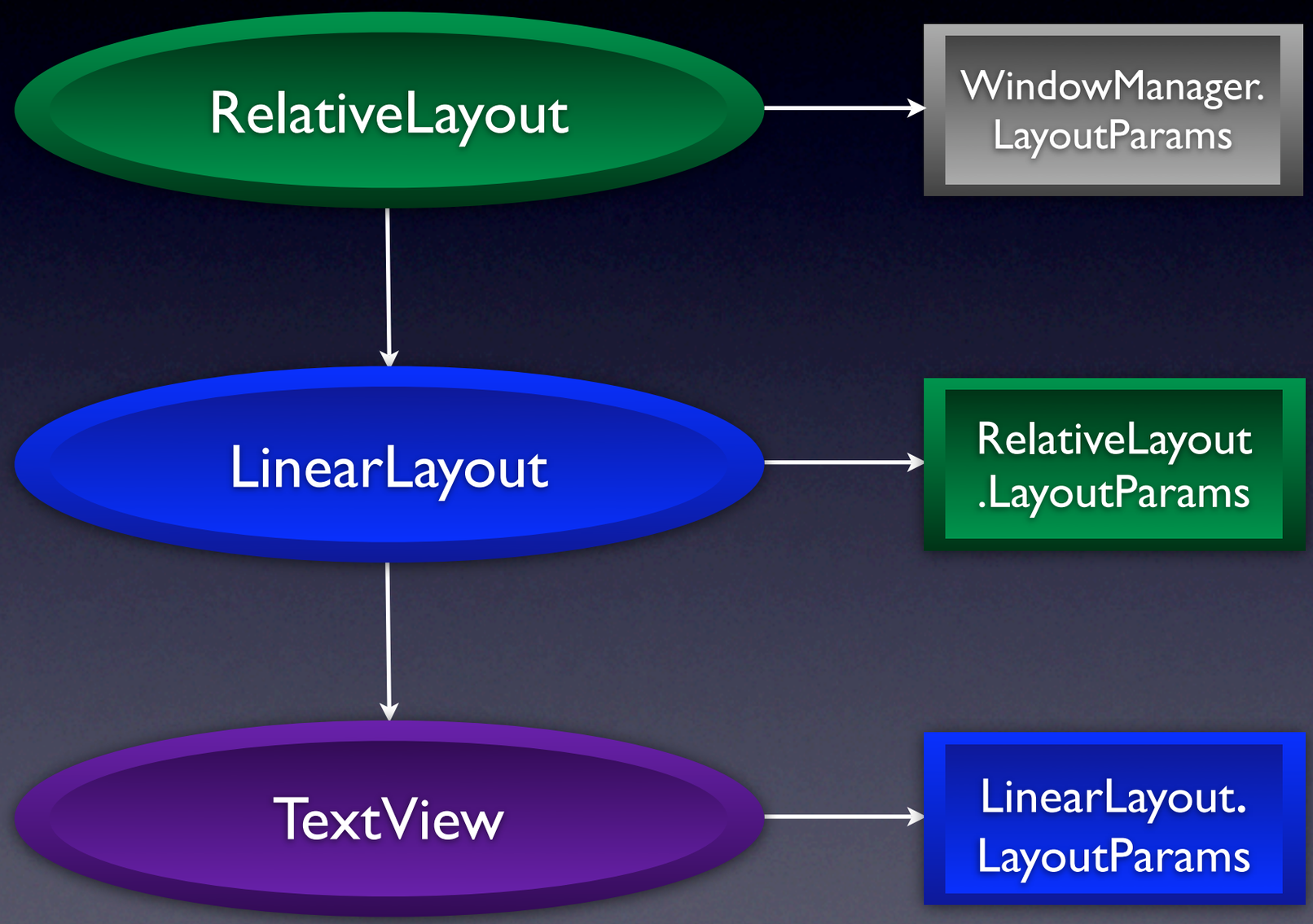
Google
CONFIDENTIAL

LayoutParams

- All Views have properties that are intrinsic to their type
  - TextViews have textSize, textColor, etc.
- All Views also have a LayoutParams object which describe how they should be sized & positioned
  - A TextView inside a LinearLayout will have a LinearLayout.LayoutParams object

Google
CONFIDENTIAL

LayoutParams



Sample view tree

Google
CONFIDENTIAL

Views in XML

```
<TextView
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:layout_weight="1"
  android:background="@android:drawable/box"
  android:text="This is the middle view." />
```

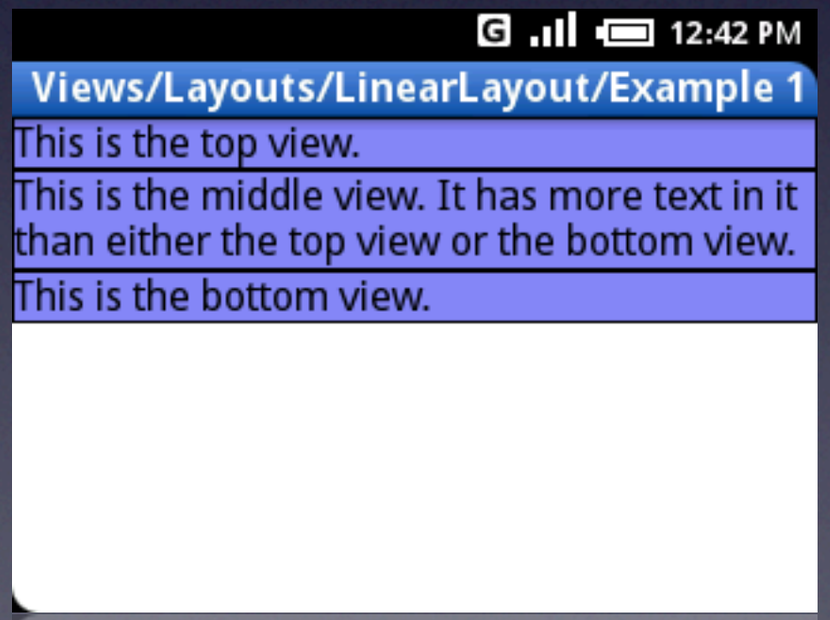
Red attributes go into the LayoutParams object and are interpreted by the parent.
Green attributes go into the TextView

Google
CONFIDENTIAL

LinearLayout Example

LinearLayout's width: fill_parent
LinearLayout's height: wrap_content

TextViews' width: fill_parent
TextViews' height: wrap_content



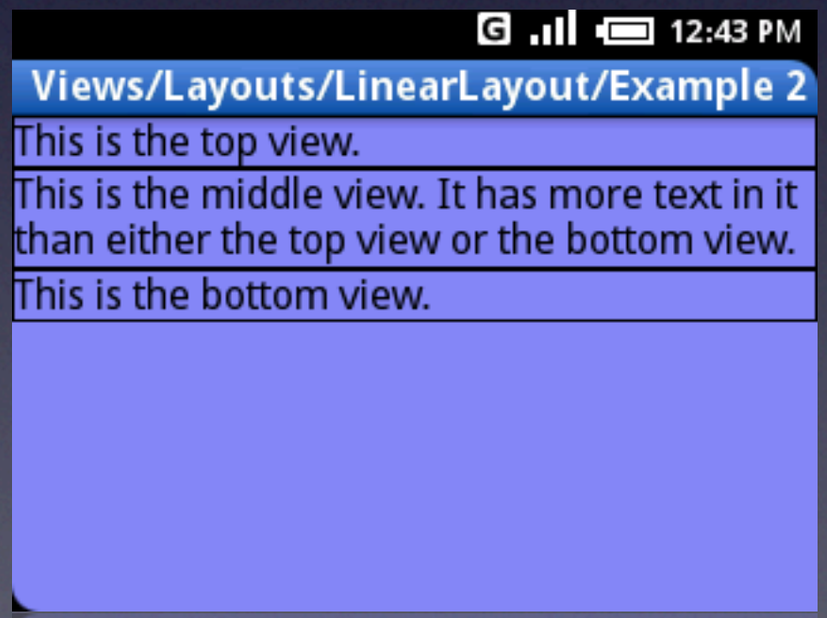
samples/SampleCode/res/layout/linear_layout*.xml

Google
CONFIDENTIAL

LinearLayout Example

LinearLayout's width: fill_parent
LinearLayout's height: fill_parent

TextViews' width: fill_parent
TextViews' height: wrap_content



Google
CONFIDENTIAL

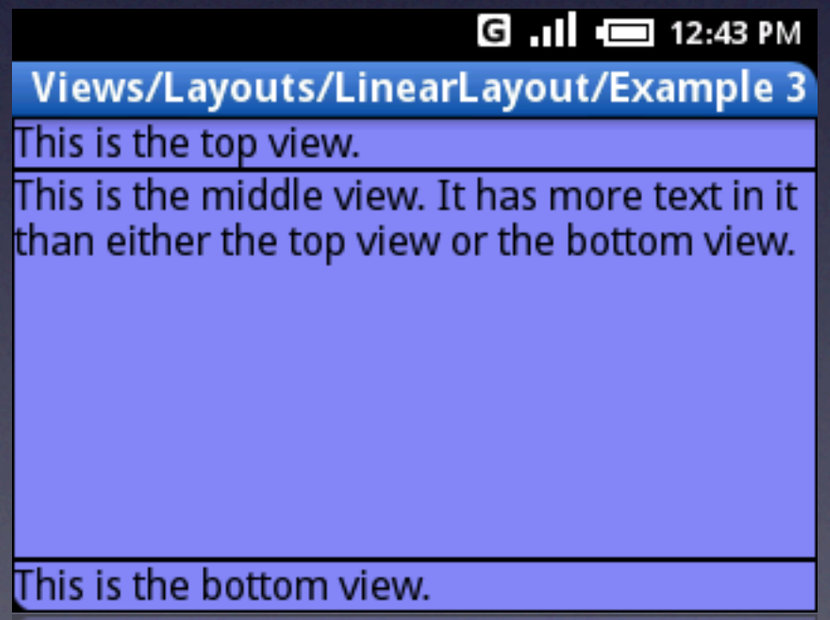
LinearLayout Example

LinearLayout's width: fill_parent
LinearLayout's height: fill_parent

TextView 1's width: fill_parent
TextView 1's height: wrap_content

TextView 2's width: fill_parent
TextView 2's height: wrap_content
TextView 2's weight: 1

TextView 3's width: fill_parent
TextView 3's height: wrap_content

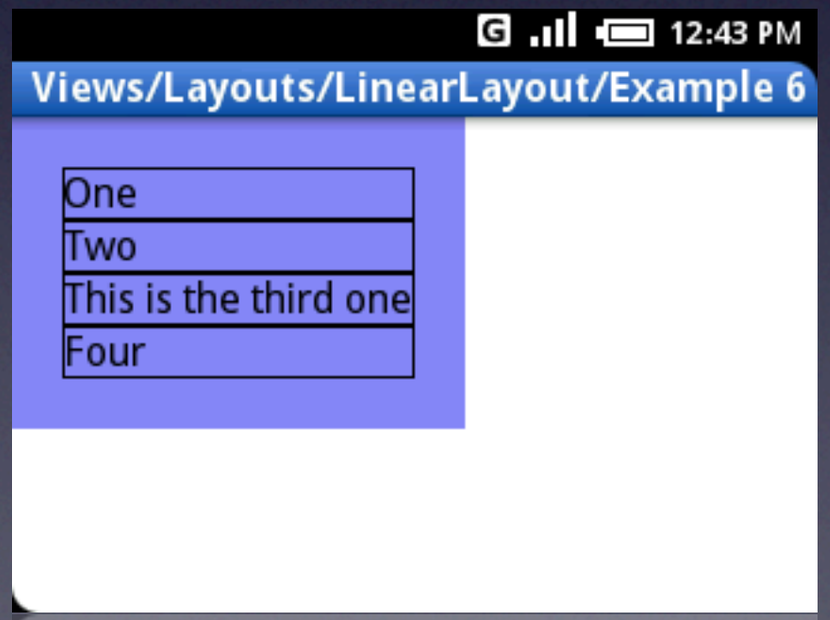


Google
CONFIDENTIAL

LinearLayout Example

LinearLayout's width: wrap_content
LinearLayout's height: wrap_content
LinearLayout's padding: 20

TextViews' width: fill_parent
TextViews' height: wrap_content

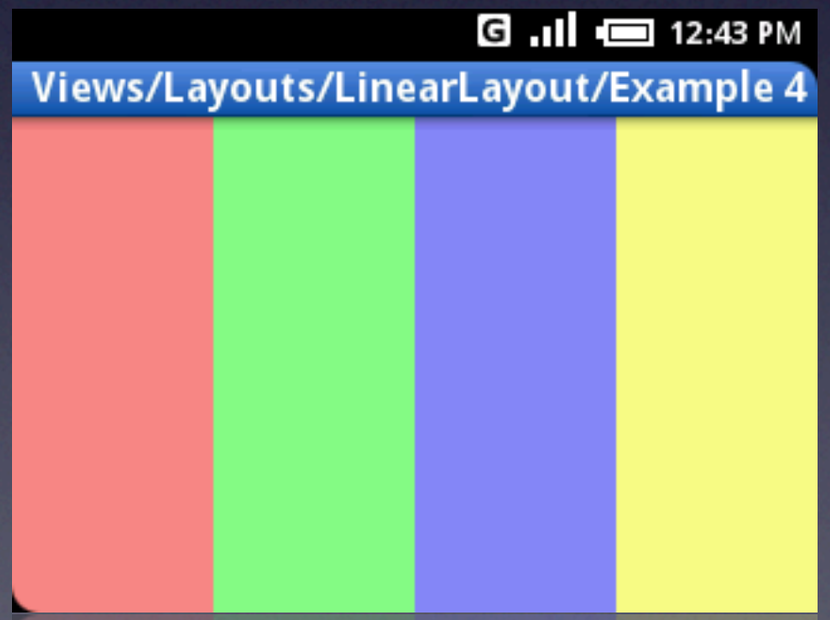


Google
CONFIDENTIAL

LinearLayout Example

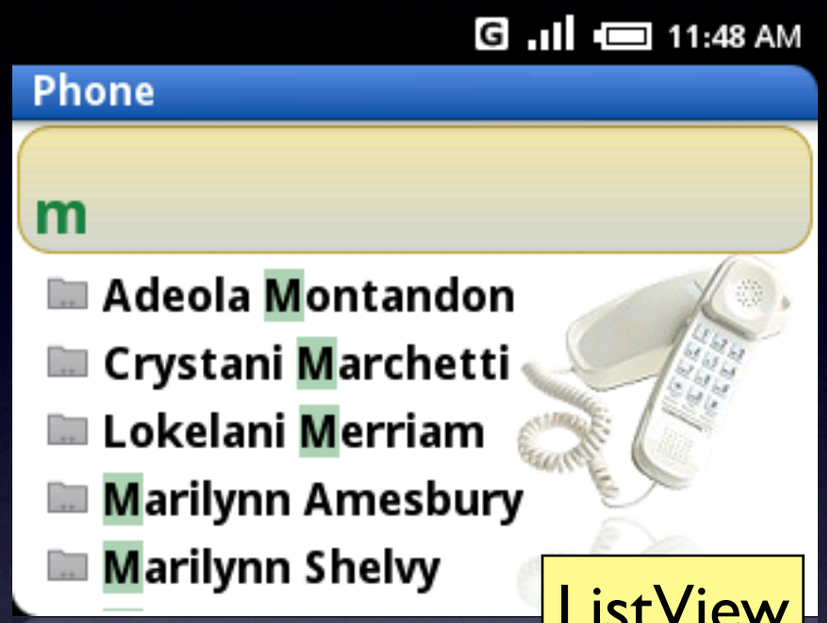
LinearLayout's width: fill_parent
LinearLayout's height: fill_parent
LinearLayout's orientation: horizontal

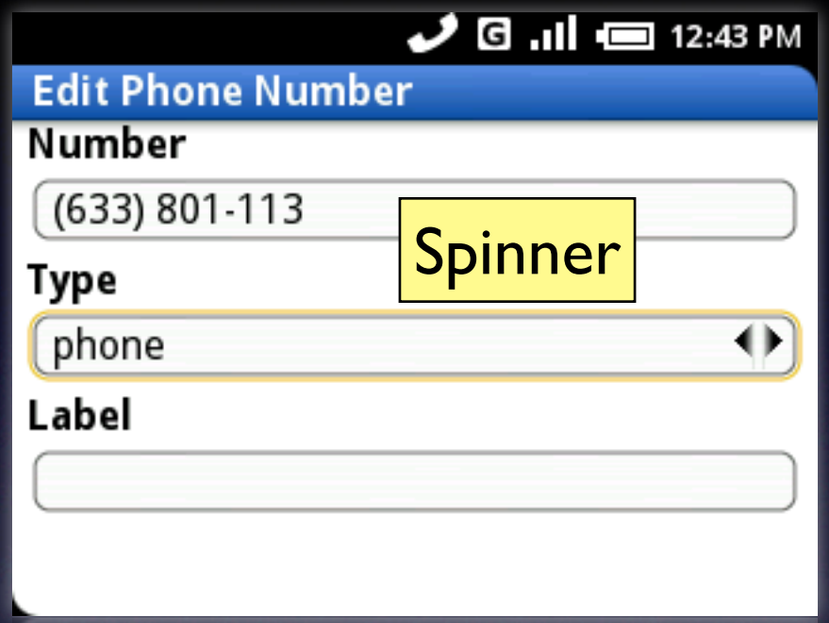
TextViews' width: 0
TextViews' weight: 1
TextViews' height: fill_parent

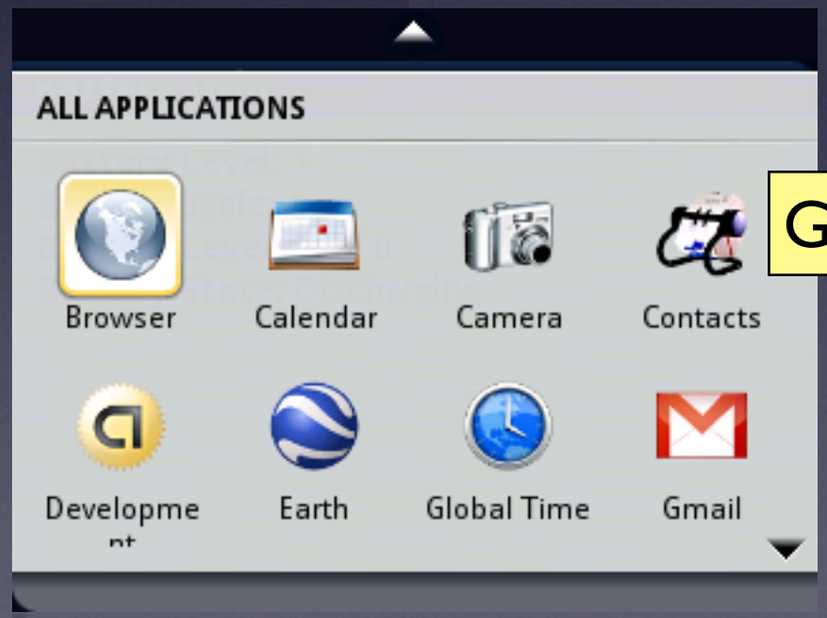


Google
CONFIDENTIAL

AdapterView Examples







Google
CONFIDENTIAL

Adapters

- Adapters are where AdapterViews get their data from.
  - getCount()
  - getItem(int position)
- Also responsible for providing a View to display data at a particular position
  - getView(int position, ...)
- Adapters notify their host when their data changes

Google
CONFIDENTIAL

Adapters

- You can write your own, or...
  - CursorListAdapter: data comes from a Cursor
  - SimpleCursorListAdapter: all views come from the same XML file; maps column names to views.
  - ArrayListAdapter: data lives in an array

Google
CONFIDENTIAL

Other Useful Views

- ScrollView: Add as parent of another View (typically LinearLayout) to make that View scroll
- TabHost/TabWidget: Can switch between embedded Activities (or Views)
- ViewFlipper: Shows child Views one at a time
- ViewSwitcher: Double-buffers Views to enable transition effects

Google
CONFIDENTIAL

Animation

- Views have built-in support for animations
- Animations can be calculated from code or loaded from resources
- Primitive animations (scale, translate, rotate, alpha) can be composed for more complex effects
- Can apply different Interpolators to vary tweening calculations (EaseIn, EaseOut, etc)

Google
CONFIDENTIAL

Animation Resources

```
Animation shake =
    AnimationUtils.loadAnimation(this,
                                 R.anim.shake);
v.startAnimation(shake);
```

```
<translate
  android:fromX="0"
  android:toX="10"
  android:duration="1000"
  android:interpolator="@anim/cycle_7" />
```

Google
CONFIDENTIAL

Graphics: SGL

- Postscript-like 2D API
  - SVG capable
  - Object Oriented
  - Complex polygons
  - Powerful paints
  - 2D and 3D Transformations
- High quality
  - Floating-point coordinates
  - Everything can be anti-aliased
  - Dithered gradients

Google
CONFIDENTIAL

Graphics: SGL

- **Canvas**
  - Destination of rendering operations (a Surface or a Bitmap)
  - Contains current clipping and transform
  - Rasterization operations (fill, bitmap, text, paths)
- **Paint**
  - Defines all parameters used to fill a shape
  - Color, Gradient, Pen Size, Stippling, Anti-aliasing, etc...
- **Path**
  - Used to define the shapes to render
  - Comprised of Lines segments, Bezier curves and Arcs
  - Paths can be stroked and used as baseline for text
  - Can be concave and self-intersecting

Google
CONFIDENTIAL

Graphics: OpenGL|ES

- Embedded version of the industry-standard OpenGL API
- Available to applications through JSR 239
- Software implementation
  - Fast, uses dynamic code generation
  - Version 1.0 with extensions
- Hardware implementation where available

Google
CONFIDENTIAL

Graphics: OpenGL|ES

- Syncronize drawing (e.g. in View.draw)
  - OpenGLContext.waitNative()
  - OpenGLContext.waitGL()

```
OpenGLContext glc = new OpenGLContext(0);
GL10 gl = (GL10)( glc.getGL() );

glc.waitNative( canvas, ... );

// use JSR239 here
gl.glViewPort( ... )

glc.waitGL();
```

Google
CONFIDENTIAL

ContentResolver

- Perform operations on URIs
- Find ContentProvider responsible for URI's authority and call corresponding API on it
- Loads/starts ContentProvider if needed
- Uses information in AndroidManifest.xml to match authority to ContentProvider

Google
CONFIDENTIAL

ContentResolver

- Cursor query(ContentURI uri, ...)
  Perform a SQL-like query on the provider
- String getType(ContentURI uri)
  Retrieve MIME type of a provider's URI
- ContentURI insert(ContentURI uri, ...)
  Add a new row of data to the provider
- int delete(ContentURI uri, ...)
  Remove one or more rows of data
- int update(ContentURI uri, ...)
  Modify the data in a row

Google
CONFIDENTIAL

Persistent Storage

- Every application is given a directory (on internal flash) in which to store its data
- Sub-directories hold different kinds of data
  - Raw files the application can manage
  - Structured files for “preferences”
  - SQLite Databases / Content Providers
- Use Context to access your data

Google
CONFIDENTIAL

Storage: Raw Files

- Context APIs to read/write/delete/list files:
  - FileInputStream openFileInput(String name)
  - FileOutputStream openFileOutput(String name)
  - boolean deleteFile(String name)
  - String[] fileList()
- Other applications can not access these files
- Example: used by maps to cache tiles

Google
CONFIDENTIAL

Storage: Preferences

- Context APIs to read/write data in a Map
  - void writePreferences(String name, Map prefs)
  - Map readPreferences(String name)
- Activity also has APIs for per-Activity prefs
  - writePreferences(Map prefs)
  - Map readPreferences()
  - Simply calls the former with Activity's class name
- Preferences currently stored as XML

Google
CONFIDENTIAL

Storage: Databases

- Context APIs to create/open databases:
  - SQLiteDatabase createDatabase()
  - SQLiteDatabase openDatabase()
- Provides raw SQLite access, plus helper functions: query(), insert(), delete(), update()
- High-level Cursor interface that can be used across processes

Google
CONFIDENTIAL

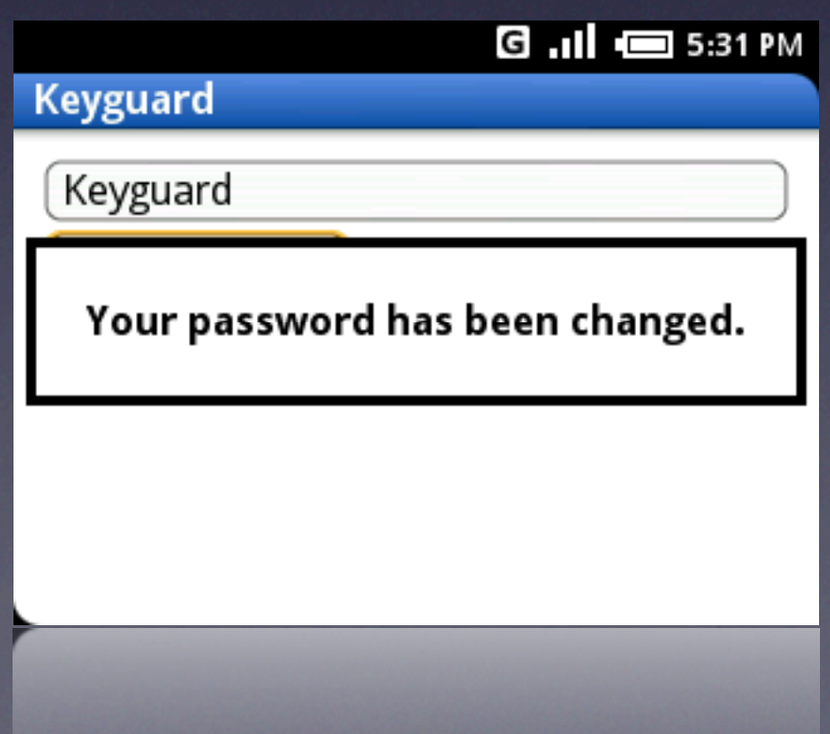
Alarm Manager

- Alarms will cause your app to be started
  - Can wake up device
  - Can be based on real or elapsed time
- set(int type, long triggerAtTime, Intent intent);
- setRepeating(int type, long triggerAtTime,
  long interval, Intent intent);
- cancel(Intent intent);

Google
CONFIDENTIAL

Transient Notifications

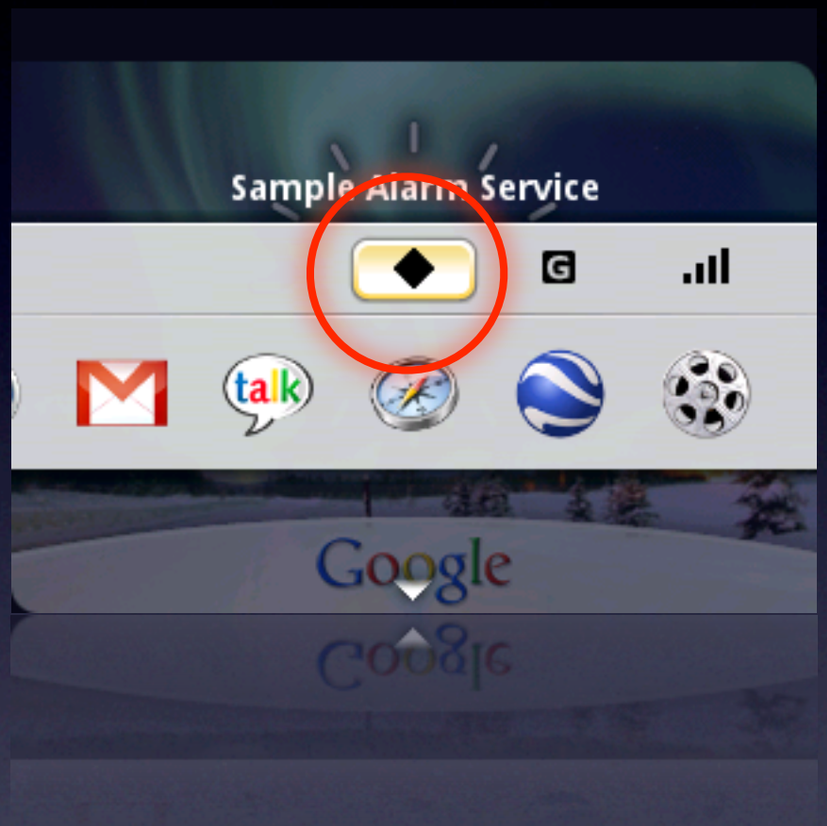
```
notify(int id,
  CharSequence transientMessage,
  int transientDuration);

notifyView(int id,
  View transientView,
  int transientDuration);
```



Google
CONFIDENTIAL

Persistent Notifications

- Apps can add notifications to status bar and app launcher
- Notifications can have an associated action (Intent)



Google
CONFIDENTIAL

Persistent Notifications

```
void notify(int id,
    CharSequence transientMessage,
    int transientDuration,
    int persistentIcon,
    CharSequence persistentLabel,
    Intent peristentIntent,
    int persistentHow);
```

Google
CONFIDENTIAL

Context

- Activity is a Context
- Access to system APIs
  - Resources
  - ContentResolver
  - Find system Services:
    - *Window Manger*
    - *Alarm Manager*
    - *NotificationManager*
    - *View Inflation*
- Start/Stop/Bind Services
- Register/Broadcast Intents
  - Preferences
  - Storage

Google
CONFIDENTIAL

Development

Tools

Google
CONFIDENTIAL

Android Tools

- Compatible with standard IDEs
  - Eclipse, IntelliJ
- Android-specific tools to increase visibility into what is happening on device
  - Full support for on-device debugging
  - Performance & memory profiling

Google
CONFIDENTIAL

Eclipse Device Debugging

Debug – View.java – Eclipse SDK – /Users/mcleron/Documents/workspace-x

Debug

TelephonyEditor [Remote Java Application]
DalvikVM[localhost:8700]
Thread [<3> Main] (Suspended)
PhoneWindow$DecorView(View).draw(Canvas, Transformation
PhoneWindow$DecorView.draw(Canvas, Transformation, long
PhoneWindow$DecorView(View).drawTraversal(Canvas, Transf
PhoneWindow$DecorView(FrameLayout).drawTraversal(Canva
PhoneWindow$DecorView.drawTraversal(Canvas, Transformat
ViewRoot.draw(boolean) line: 373
ViewRoot.performTraversals() line: 288
ViewRoot.handleMessage(Message) line: 421
ViewRoot(Handler).dispatchMessage(Message) line: 40

Call stack

Variables | Breakpoints

Name	Value
this	PhoneWindow$DecorView (id=830011
canvas	Canvas (id=830011571256)
mBitmap	null
mNativeCanvas	1249144
mSurfaceFormat	4
currentTransformation	null
drawTime	1179363240090

Examine variables

View.java

```
1823     * @param drawTime The time at which the current draw pass started
1824     */
1825    protected void draw(Canvas canvas, Transformation currentTransformation, long drawTime) {
1826        Drawable bg = mBGDrawable;
1827        if (bg == null) {
1828            mBackgroundSizeChanged = false;
1829            return;
1832        if (mBackgroundSizeChanged) {
1833            bg.setBounds(0, 0,  mRight - mLeft, mBottom - mTop);
1834        }
```

Breakpoints, single stepping

Console | Tasks | Display | Search

CONFIDENTIAL

Dalvik Debug Monitor Server

- Makes Java Debugging Easier
  - Shows process names and remote-debug port numbers
  - Provides debugger network pass-through to Java apps
  - Shortcut to debug current app: select it, connect to port 8700
  - Apps that are told to wait for debugger are highlighted in DDMS UI; Allows debugging startup problems

Google
CONFIDENTIAL

Dalvik Debug Monitor Server

- Finds all Java apps running on the device
- Automatically detects device when attached to USB
- Displays statistics on thread and memory usage.
- Commands to gather for bug reports
  - *screen shots, radio logs, system state*

Google
CONFIDENTIAL

Dalvik Debug Monitor Server

Dalvik Debug Monitor

ADB proxy (device) | Update Threads | Update Heap | Halt | GC

Examine thread & heap state

Java processes

	Client Name
00	system_process
adb 8001	com.google.android.phone
adb 8002	com.google.android.home
adb 8004	com.google.android.gtalk
adb 8005	<pre-initialized>

Info | Threads | Java Heap | Native Heap

ID	Tid	Status	utime	stime	Name
3	279	wait	1353	104	Main
*5	284	vmwait	8	0	HeapWorker
*7	285	vmwait	0	0	Signal Catcher
*9	286	running	45	102	JDWP
11	292	native	20	1	Binder Thread #1
				0	Binder Thread #2
				0	Binder Thread #3

Remote Command

Executing 'logcat '*:d jdwp:w"'

View logs

```
D/dalvikvm(  433): Sending app name
I/dalvikvm(  433): JDWP has attached
I/dalvikvm(  433): Broadcasting DDM connect
I/ddm-hello(  433): Connected!
D/ddm-hello(  433): Server version is 1
D/ActivityManager(  248): Broadcast: Intent { action=android.intent.action.TIME_TICK }
I/AlarmManager(  248): Intent: Intent { action=android.intent.action.TIME_TICK }
D/ActivityManager(  248): Broadcast: Intent { action=android.intent.action.TIME_TICK }
I/AlarmManager(  248): Intent: Intent { action=android.intent.action.TIME_TICK }
D/ActivityManager(  248): Broadcast sticky: Intent { action=android.intent.action.BATTERY_CHANGED
extras={scale=9, icon-large=16973963, level-raw=69, draw-level=6666, icon-small=16973940,
state=Normal, level=6} }
D/ActivityManager(  248): Broadcast: Intent { action=android.intent.action.TIME_TICK }
I/AlarmManager(  248): Intent: Intent { action=android.intent.action.TIME_TICK }
D/ActivityManager(  248): Broadcast: Intent
{ action=android.intent.action.SIGNAL_STRENGTH_CHANGED extras={asu=27, phoneName=GSM} }
D/ActivityManager(  248): Broadcast: Intent { action=android.intent.action.TIME_TICK }
I/AlarmManager(  248): Intent: Intent { action=android.intent.action.TIME_TICK }
D/ActivityManager(  248): Broadcast: Intent { action=android.intent.action.TIME_TICK }
I/AlarmManager(  248): Intent: Intent { action=android.intent.action.TIME_TICK }
D/ActivityManager(  248): Broadcast sticky: Intent { action=android.intent.action.BATTERY_CHANGED
extras={scale=9, icon-large=16973963, level-raw=70, draw-level=6666, icon-small=16973940,
state=Normal, level=6} }
D/ActivityManager(  248): Broadcast: Intent { action=android.intent.action.TIME_TICK }
I/AlarmManager(  248): Intent: Intent { action=android.intent.action.TIME_TICK }
```

Save | Cancel

Idle

Device Screen Capture

Captured image:

Screen capture

Refresh | Save | Done

Emulator

- ARM emulator runs same code as device
- Uses same tool set
- Allows profiling of native code

Google
CONFIDENTIAL

Performance Tools

- Custom-built tools offers accurate, detailed performance analysis of code running on device
- Linked views to help navigate large data sets
- Powerful visualizations to identify performance bottlenecks

CONFIDENTIAL

Performance: Traceview

Traceview: /Users/mcleron/Desktop/jtrace/com.google.android.pressmwarminst.PressMWarmInstrumentationProfiled.jtrace

msec: 452.295 max msec: 526.464

24 android/view/View.drawTraversal (Landroid/graphics/Canvas;Landroid/view/animation/Transformation;)V

0 50 100 150 200 250 300 350 400 450 500

mInstrumentationProfiled

[3] Main

[17] Binder Thread #3

Timeline view

Name	Incl %	Inclusive	Excl %	Exclusive	Calls+Recur Calls/Total
8 android/view/ViewRoot.performTraversals ()V	32.2%	169.472	0.1%	0.545	2+0
Parents					
7 android/view/ViewRoot.handleMessage (Landroid/os/Message;)V	100.0%	169.472			2/2
Children					
self	0.3%	0.545			
37 android/view/View.setFrameTraversal (IIIIII)V	38.0%	64.377			1/17
41 android/view/ViewRoot.draw (Z)V	37.2%	63.036			2/2
62 android/policy/PhoneWindow$DecorView.measure (II)V	20.1%	34.085			1/1
198 android/view/IWindowSession$Stub$Proxy.relayout (Landroid/view/IWindo	3.4%	5.688			2/2
388 android/view/IWindowSession$Stub$Proxy.finishDrawing (Landroid/view/IV	0.8%	1.430			1/1
775 android/view/ViewRoot.getRootMeasureSpec (II)I	0.1%	0.152			2/2
774 android/graphics/Rect.width ()I	0.0%	0.058			4/11
782 android/graphics/Rect.height ()I	0.0%	0.050			4/11
420 android/view/View.getVisibility ()I	0.0%	0.028			2/86
574 android/view/View.getLayoutParams ()Landroid/view/ViewGroup$LayoutP	0.0%	0.023			2/43
9 com/google/android/phone/Dialer.initializeCursor ()V	28.1%	147.994	0.0%	0.107	1+0
10 android/widget/DialerFilter.onKeyDown (ILandroid/os/KeyEvent;)Z	28.0%	147.414	0.0%	0.107	1+0
11 android/text/SpannableStringBuilder.replace (IILjava/lang/CharSequence;II)I	24.0%	126.522	0.0%	0.231	3+0
12 android/text/SpannableStringBuilder.change (IILjava/lang/CharSequence;II)I	23.8%	125.414	0.0%	0.093	3+0
13 android/text/SpannableStringBuilder.change (ZIILjava/lang/CharSequence;II)I	23.8%	125.321	0.2%	1.175	3+0
14 android/text/SpannableStringBuilder.send	22.9%	120.885	0.1%	0.388	3+0
15 android/widget/TextView.onKeyDown (ILar	18.7%	98.727	0.0%	0.041	1+0
16 android/text/method/MultiTapInputMethod	18.7%	98.686	0.0%	0.070	1+0
17 android/text/method/QwertyInputMethod.	18.7%	98.616	0.1%	0.455	1+0
18 android/text/SpannableStringBuilder.repla	18.2%	95.813	0.0%	0.042	1+0
19 android/pim/ContactPickerActivity$ContactPickerTextWatcher.onTextChanged (Ljava	17.1%	90.067	0.0%	0.045	1+0
20 android/pim/ContactPickerActivity.search (Ljava/lang/String;)V	17.0%	89.593	0.0%	0.102	1+0
21 android/policy/PhoneWindow$DecorView.drawTraversal (Landroid/graphics/Canvas	16.4%	86.270	0.0%	0.098	2+0
22 android/widget/DialerFilter.setMode (I)V	16.3%	85.880	0.0%	0.205	2+0

Profile view

Find:

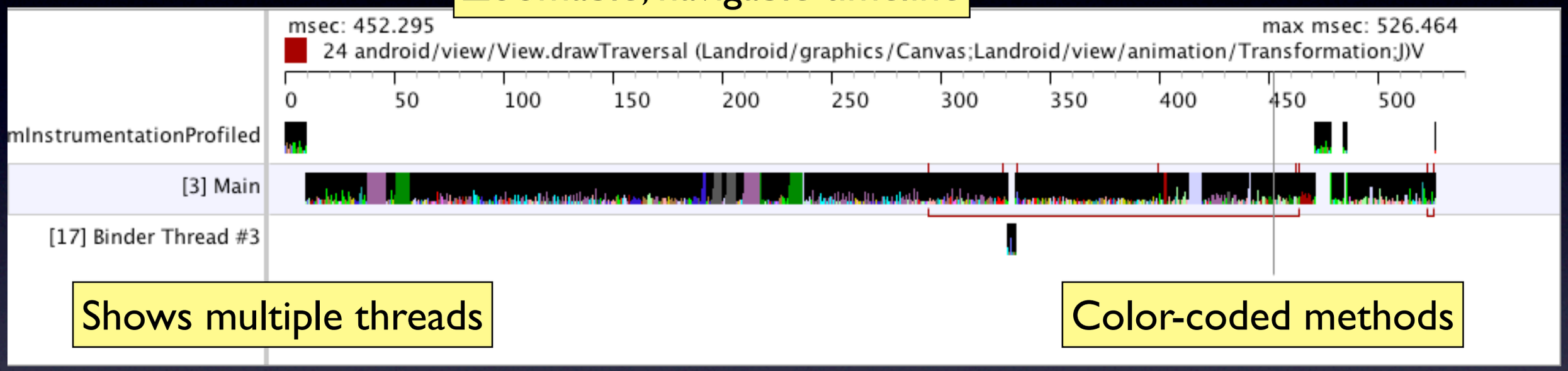
Per-thread Call Tree	Cpu time	% of Total
183 android/os/MessageQueue.next ()Landroid/os/Message;	0.128	0.0%
1 android/os/Handler.dispatchMessage (Landroid/os/Message;)V	1.991	0.4%
308 android/os/Message.recycle ()V	0.067	0.0%
183 android/os/MessageQueue.next ()Landroid/os/Message;	4.51	0.9%
1 android/os/Handler.dispatchMessage (Landroid/os/Message;)V	2.934	0.6%
308 android/os/Message.recycle ()V	0.066	0.0%
183 android/os/MessageQueue.next ()Landroid/os/Message;	0.136	0.0%
1 android/os/Handler.dispatchMessage (Landroid/os/Message;)V	35.407	6.7%
2 android/app/ActivityThread$H.handleMessage (Landroid/os/Messa	35.377	6.7%
59 android/app/ActivityThread.access$1000 (Landroid/app/Act	35.335	6.7%
60 android/app/ActivityThread.handleStopActivity (Landroid/	35.248	6.7%
241 java/util/HashMap.get (Ljava/lang/Object;)Ljava/lar	0.218	0.0%
894 android/app/ActivityThread$StopInfo.<init> (Landr	0.058	0.0%
69 android/app/ActivityThread.performStopActivityInner	30.939	5.9%
86 android/app/ActivityThread.createThumbnailBitn	26.248	5.0%
596 android/graphics/Bitmap.createBitmap (IIZ)	0.438	0.1%
797 android/graphics/Bitmap.eraseColor (I)V	0.133	0.0%
711 android/graphics/Canvas.<init> (Landroid/	0.233	0.0%
88 android/app/Activity.onCreateThumbnail (Lar	25.318	4.8%
1033 android/app/Activity.onCreateDescription ()Lja	0.012	0.0%
220 android/app/Activity.performStop ()V	4.612	0.9%
296 android/view/View.setVisibility (I)V	1.1	0.2%
883 android/app/ActivityManagerNative.getDefault ()Lan	0.016	0.0%
282 android/app/ActivityManagerProxy.activityStopped (	2.751	0.5%
308 android/os/Message.recycle ()V	0.064	0.0%
183 android/os/MessageQueue.next ()Landroid/os/Message;	0.134	0.0%
1 android/os/Handler.dispatchMessage (Landroid/os/Message;)V	0.074	0.0%
308 android/os/Message.recycle ()V	0.054	0.0%
183 android/os/MessageQueue.next ()Landroid/os/Message;	0.195	0.0%
1 android/os/Handler.dispatchMessage (Landroid/os/Message;)V	3.277	0.6%
308 android/os/Message.recycle ()V	0.063	0.0%
183 android/os/MessageQueue.next ()Landroid/os/Message;	0.41	0.1%
[17] Binder Thread #3	3.696	0.7%

Callstack view

Google CONFIDENTIAL

Timeline View

Zoomable, navigable timeline



Shows multiple threads

Color-coded methods

Google
CONFIDENTIAL

Profile View

Summarizes cost of each method

Inclusive & exclusive method times

Name	Incl %	Inclusive	Excl %	Exclusive	Calls+Recur Calls/Total
8 android/view/ViewRoot.performTraversals ()V	32.2%	169.472	0.1%	0.545	2+0
Parents					
7 android/view/ViewRoot.handleMessage (Landroid	00.0%	169.472			2/2
Children					
self	0.3%	0.545			
37 android/view/View.setFrameTraversal (IIIIII)V	38.0%	64.377			1/17
41 android/view/ViewRoot.draw (Z)V	37.2%	63.036			2/2
62 android/policy/PhoneWindow$DecorView.measure (II)V	20.1%	34.085			1/1
198 android/view/IWindowSession$Stub$Proxy.relayout (Landroid/view/IWindo	3.4%	5.688			2/2
388 android/view/IWindowSession$Stub$Proxy.finishDrawing (Landroid/view/IV	0.8%	1.430			1/1
775 android/view/ViewRoot.getRootMeasureSpec (II)I	0.1%	0.152			2/2
774 android/graphics/Rect.width ()I	0.0%	0.058			4/11
782 android/graphics/Rect.height ()I	0%	0.050			4/11
420 android/view/View.getVisibility ()I	0%	0.028			2/86
574 android/view/View.getLayoutParams ()Landrol	0%	0.023			2/43
9 com/google/android/phone/Dialer.initializeCursor ()V	28.1%	147.994	0.0%	0.107	1+0
10 android/widget/DialerFilter.onKeyDown (ILandroid/os/KeyEvent;)Z	28.0%	147.414	0.0%	0.107	1+0
11 android/text/SpannableStringBuilder.replace (IILjava/lang/CharSequence;II)I	24.0%	126.522	0.0%	0.231	3+0
12 android/text/SpannableStringBuilder.change (IILjava/lang/CharSequence;II)I	23.8%	125.414	0.0%	0.093	3+0
13 android/text/SpannableStringBuilder.change (ZIILjava/lang/CharSequence;II)I	23.8%	125.321	0.2%	1.175	3+0
14 android/text/SpannableStringBuilder.sendTextChange ([Ljava/lang/Object;III)V	22.9%	120.885	0.1%	0.388	3+0
15 android/widget/TextView.onKeyDown (ILandroid/os/KeyEvent;)Z	18.7%	98.727	0.0%	0.041	1+0
16 android/text/method/MultiTapInputMethod.onKeyDown (Landroid/text/Editable;ILa	18.7%	98.686	0.0%	0.070	1+0
17 android/text/method/QwertyInputMethod.qwerty (Landroid/text/Editable;ILandroid,	18.7%	98.616	0.1%	0.455	1+0
18 android/text/SpannableStringBuilder.replace (IILjava/lang/CharSequence;)I	18.2%	95.813	0.0%	0.042	1+0
19 android/pim/ContactPickerActivity$ContactPickerTextWatcher.onTextChanged (Ljava	17.1%	90.067	0.0%	0.045	1+0
20 android/pim/ContactPickerActivity.search (Ljava/lang/String;)V	17.0%	89.593	0.0%	0.102	1+0
21 android/policy/PhoneWindow$DecorView.drawTraversal (Landroid/graphics/Canvas	16.4%	86.270	0.0%	0.098	2+0
22 android/widget/DialerFilter.setMode (I)V	16.3%	85.880	0.0%	0.205	2+0

Calls to method

Calls from method

Find:

CONFIDENTIAL

Callstack View

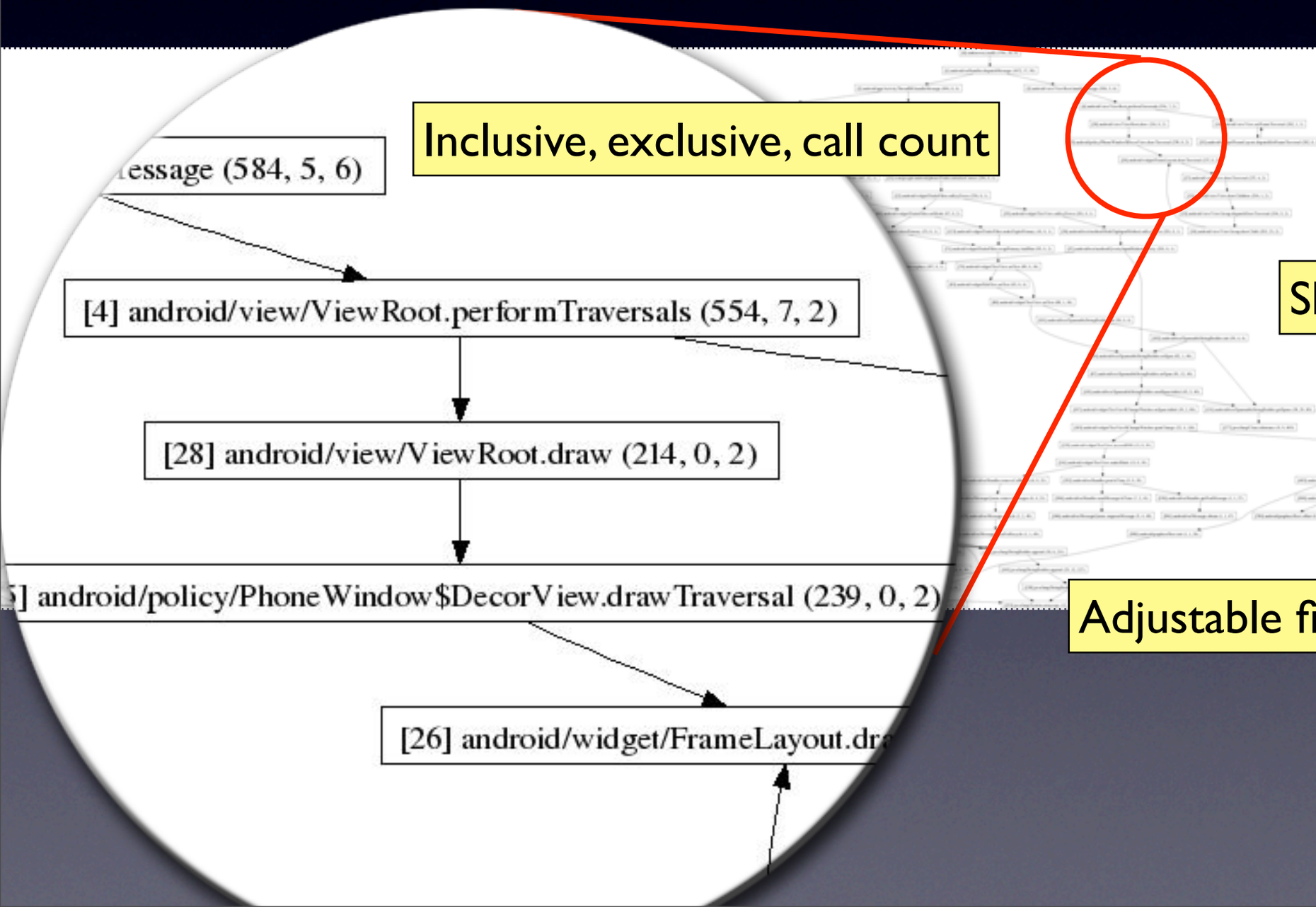
Per-thread Call Tree	Cpu time	% of Total
183 android/os/MessageQueue.next ()Landroid/os/Message;	0.128	0.0%
1 android/os/Handler.dispatchMessage (Landroid/os/Message;)V	1.991	0.4%
308 android/os/Message.recycle ()V	0.067	0.0%
183 android/os/MessageQueue.next ()Landroid/os/Message;	4.51	0.9%
1 android/os/Handler.dispatchMessage (Landroid/os/Message;)V	2.934	0.6%
308 android	0.066	0.0%
183 android	0.136	0.0%
1 android/os/Handler.dispatchMessage (Landroid/os/Message;)V	35.407	6.7%
2 android/app/ActivityThread$H.handleMessage (Landroid/os/Mess	35.377	6.7%
59 android/app/ActivityThread.access$1000 (Landroid/app/Act	35.335	6.7%
60 android/app/ActivityThread.handleStopActivity (Landroid,	35.248	6.7%
241 java/util/HashMap.get (Ljava/lang/Object;)Ljava/lar	0.218	0.0%
894 android/app/ActivityThread$StopInfo.<init> (Landr	0.058	0.0%
69 android/app/ActivityThread.performStopActivityInner	30.939	5.9%
86 android/app/ActivityThread.createThumbnailBitn	26.248	5.0%
596 android/graphics/Bitmap.createBitmap (IIZ)	0.438	0.1%
797 android/graphics/Bitmap.eraseColor (I)V	0.133	0.0%
711 android/graphics/Canvas.<init> (Landroid/	0.233	0.0%
88 android/app/Activity.onCreateThumbnail (La	25.318	4.8%
1033 android/app/Activity.onCreateDescription ()Lja	0.012	0.0%
220 android/app/Activity.performStop ()V	4.612	0.9%
296 android/view/View.setVisibility (I)V	1.1	0.2%
883 android/app/ActivityManagerNative.getDefault ()Lan	0.016	0.0%
282 android/app/ActivityManagerProxy.activityStopped (	2.751	0.5%
308 android/os/Message.recycle ()V	0.064	0.0%
183 android/os/MessageQueue		0.0%
1 android/os/Handler.dispatch		0.0%
308 android/os/Message.recycle ()V	0.054	0.0%
183 android/os/MessageQueue.next ()Landroid/os/Message;	0.195	0.0%
1 android/os/Handler.dispatchMessage (Landroid/os/Message;)V	3.277	0.6%
308 android/os/Message.recycle ()V	0.063	0.0%
183 android/os/MessageQueue.next ()Landroid/os/Message;	0.41	0.1%
[17] Binder Thread #3	3.696	0.7%

Shows complete call tree

Color codes critical path

Google
CONFIDENTIAL

JTracedump



Inclusive, exclusive, call count

Shows filtered call graph

Adjustable filter threshold

Google
CONFIDENTIAL

Memory Tools

- Leak checker instruments malloc
- Records stack for every allocation
- Information can be retrieved from device and viewed in DDMS

CONFIDENTIAL

DDMS: Leak checker

Dalvik Debug Monitor

ADB proxy (device) | Update Threads | Update Heap | Halt | GC

D	Port	Client Name
	adb 8000	system_process
	adb 8001	com.google.android.phone
	adb 8002	com.google.android.home
	adb 8003	<pre-initialized>
	adb 8004	com.google.android.gtalk

Info | Threads | Java Heap | Native Heap

Press 'Request Update' to retrieve latest data

Request Update

Can get heap snapshots on demand

< 1 / 154 > Total Memory: 2,252,410

List of allocation sites

Total	Count	Size	Library	Method	File
66,560	1	66,560	/android/lib/libjava_runtime.so	??	??
62,652	1	62,652	/android/lib/libdvm.so	dvmDexFileParse	dal
57,344	7	8,192	/android/lib/libdvm.so	allocThread	dal

Stack for each allocation

Address	Library	Method	File
5303bde4	/android/lib/libc.so	calloc	system/bionic/bionic
660384dc	/android/lib/libdvm.so	dvmDexFileParse	dalvik/vm/DexFile.c
660387d8	/android/lib/libdvm.so	dvmDexFileOpenFromF	dalvik/vm/DexFile.c
4a338	/android/lib/libdvm.so	dvmJarFileOpen	dalvik/vm/JarFile.c
646b0	/android/lib/libdvm.so	prepareCpe	dalvik/vm/oo/Class.
66064968	/android/lib/libdvm.so	dvmClassStartup	dalvik/vm/oo/Class.
6603ebd4	/android/lib/libdvm.so	dvmStartup	dalvik/vm/Init.c
66048090	/android/lib/libdvm.so	D_JNI_CreateJavaVM	dalvik/vm/Jni.c
5801e0f4	/android/lib/libjava_runtime.so	Autolock	include/utils/threads

Idle

CONFIDENTIAL

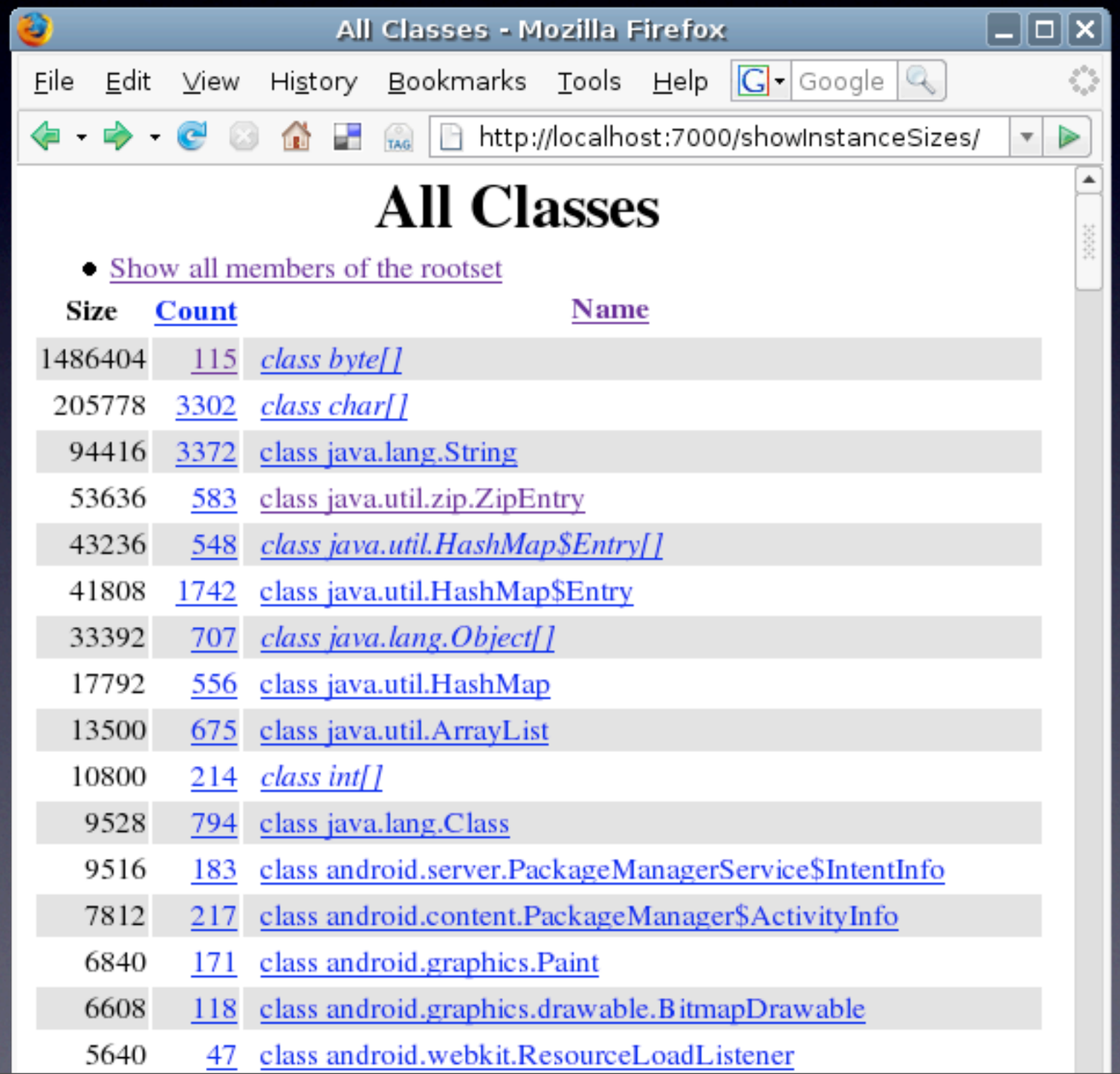
Heap Analysis Tool

- Provides a browsable snapshot of the entire VM heap
  - Object types
  - Object relationships
  - Memory usage
  - Differences over time

Google
CONFIDENTIAL

Heap Analysis Tool

- Integrated with the Dalvik VM
- Uses standard, open-source Sun formats and tools



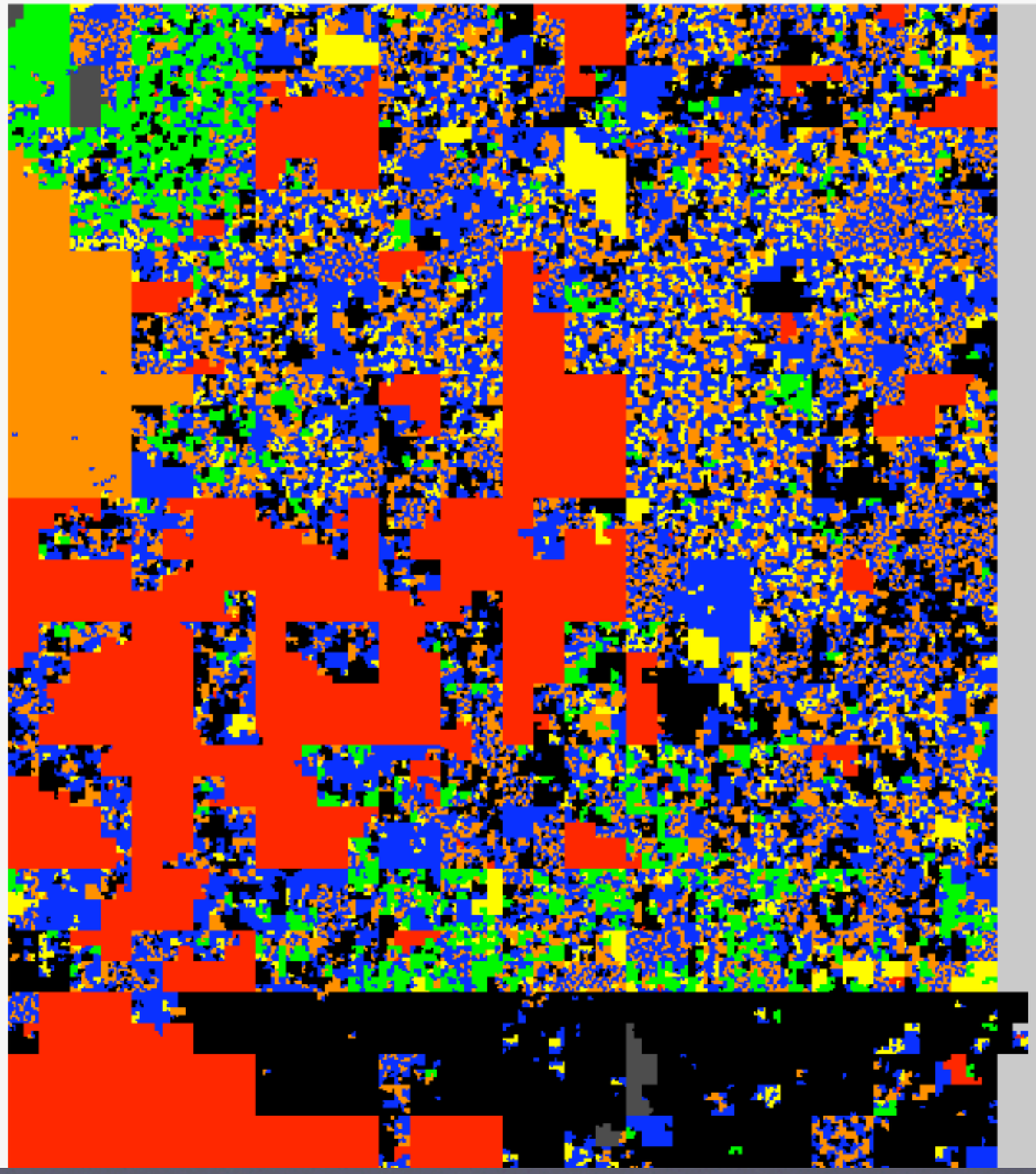
Size	Count	Name
1486404	115	*class byte[]*
205778	3302	*class char[]*
94416	3372	class java.lang.String
53636	583	class java.util.zip.ZipEntry
43236	548	*class java.util.HashMap$Entry[]*
41808	1742	class java.util.HashMap$Entry
33392	707	*class java.lang.Object[]*
17792	556	class java.util.HashMap
13500	675	class java.util.ArrayList
10800	214	*class int[]*
9528	794	class java.lang.Class
9516	183	class android.server.PackageManagerService$IntentInfo
7812	217	class android.content.PackageManager$ActivityInfo
6840	171	class android.graphics.Paint
6608	118	class android.graphics.drawable.BitmapDrawable
5640	47	class android.webkit.ResourceLoadListener

DDMS Heap Viewer

- Provides runtime inspection of VM memory usage on a device
- Lets developers visually inspect memory layout, providing insight into memory usage and fragmentation
- Memory is presented as a Hilbert curve (a type of fractal) for a more useful visual representation than typical methods

Google
CONFIDENTIAL

ID	Heap Size	Allocated	Free	% Used	# Objects
1	4.066MB	2.890MB	1.176MB	71.08%	25,791



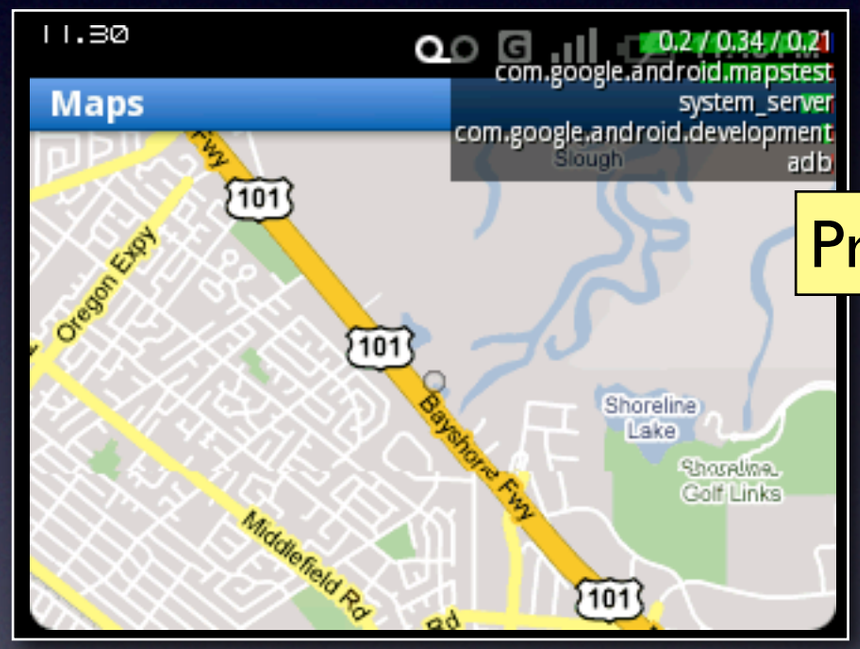
Key (1 pixel = 8 bytes):

- (heap expansion area)
- free
- data object
- class object
- 1-byte array (byte[], boolean[])
- 2-byte array (short[], char[])
- 4-byte array (object[], int[], float[])
- 8-byte array (long[], double[])
- unknown object
- non-Java object

Google
CONFIDENTIAL

Device Debug Display

Rendering frames per second



Process summary

Google
CONFIDENTIAL

Instrumentation

- Framework for driving application code with test code
  - Test code can invoke application code, simulate events, monitor results, catch exceptions, etc.
  - Test code can be downloaded
- Enables sophisticated automated testing, repeatable performance and memory analysis

Google
CONFIDENTIAL