Android Application Development

JFokus 2011

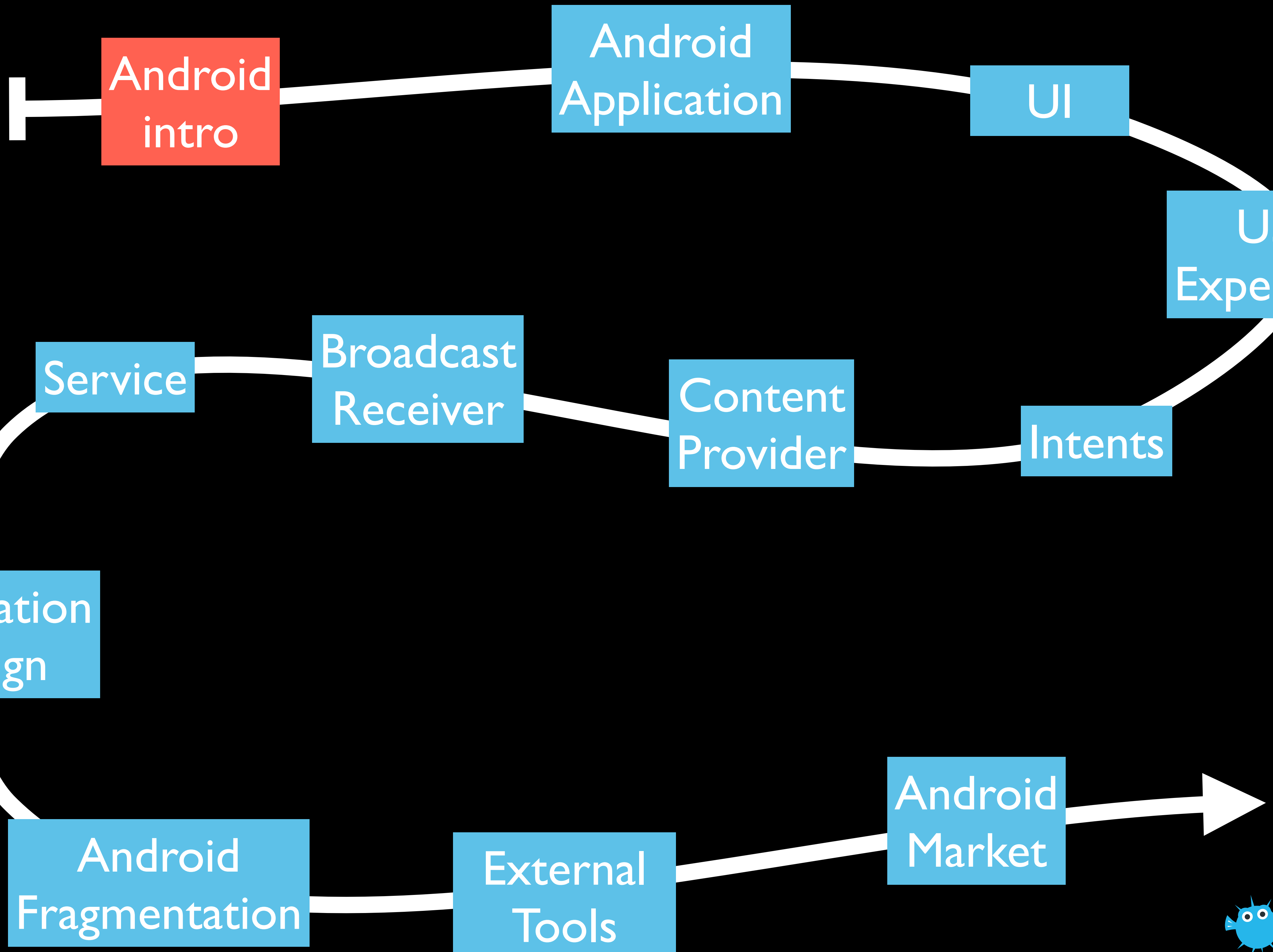
Anders Göransson

www.jayway.com
blog.jayway.com

About You

- Android experience
    - None / Little
    - Moderate / Experienced

Agenda


Android intro
Android Application
UI
User Experience
Intents
Content Provider
Broadcast Receiver
Service
Application Design
Android Fragmentation
External Tools
Android Market

Android History

- 2005 - Google acquires Android Inc.
- 2007 Sep - Mobile patents filed
- 2007 Nov - Open Handset Alliance
- 2008 Sep - Android 1.0
- 2008 Oct - G1 (HTC Dream)

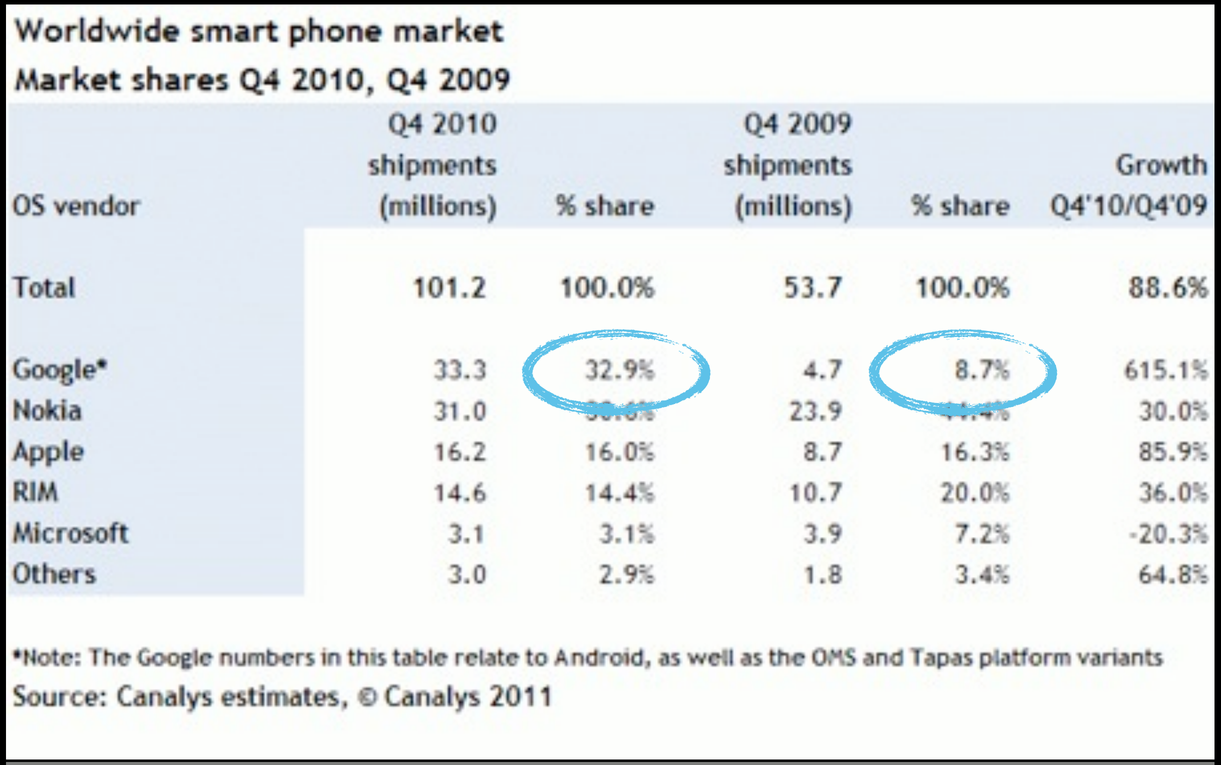
open handset alliance

JAYWAY

2011 - Why Develop for Android?

2011 - Why Develop for Android?

1. Market share. The future is Android!



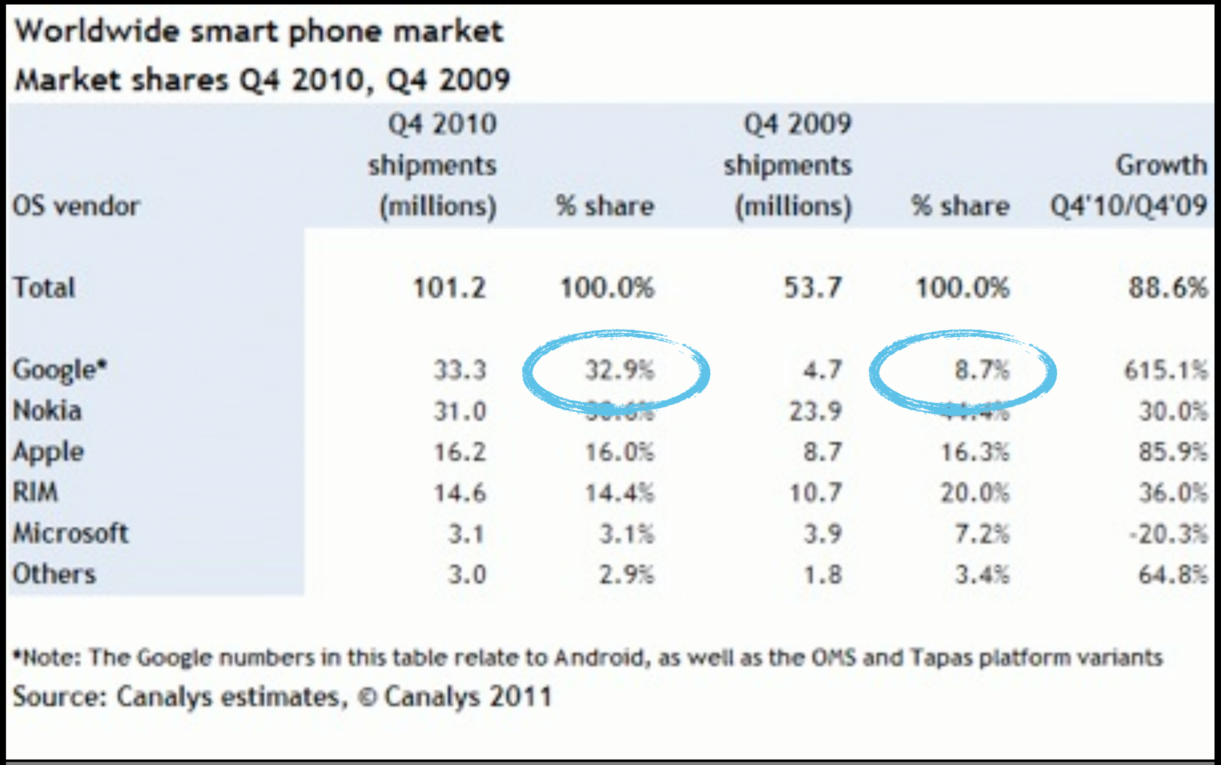
Worldwide smart phone market
Market shares Q4 2010, Q4 2009

OS vendor	Q4 2010 shipments (millions)	% share	Q4 2009 shipments (millions)	% share	Growth Q4'10/Q4'09
Total	101.2	100.0%	53.7	100.0%	88.6%
Google*	33.3	32.9%	4.7	8.7%	615.1%
Nokia	31.0	30.6%	23.9	44.4%	30.0%
Apple	16.2	16.0%	8.7	16.3%	85.9%
RIM	14.6	14.4%	10.7	20.0%	36.0%
Microsoft	3.1	3.1%	3.9	7.2%	-20.3%
Others	3.0	2.9%	1.8	3.4%	64.8%

*Note: The Google numbers in this table relate to Android, as well as the OMS and Tapas platform variants
Source: Canalys estimates, © Canalys 2011

2011 - Why Develop for Android?

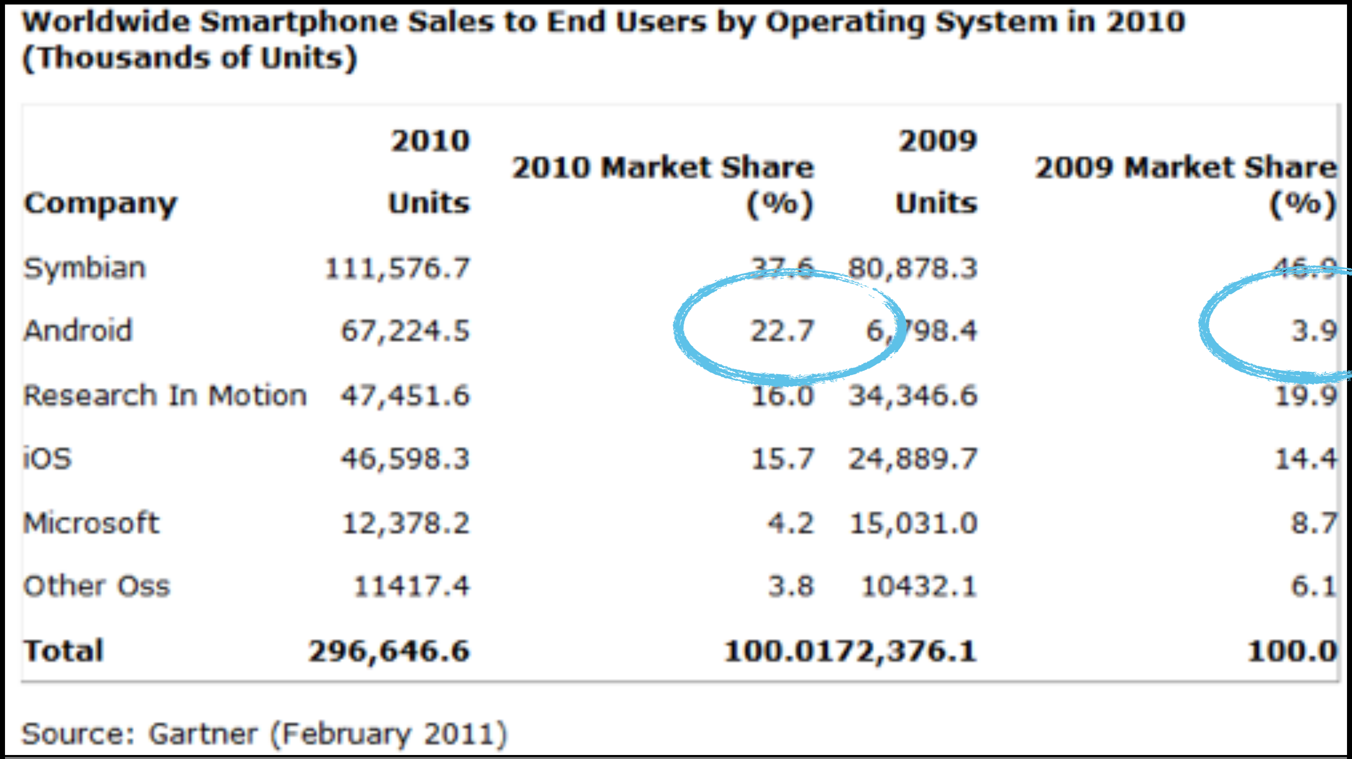
1. Market share. The future is Android!



Worldwide smart phone market
Market shares Q4 2010, Q4 2009

OS vendor	Q4 2010 shipments (millions)	% share	Q4 2009 shipments (millions)	% share	Growth Q4'10/Q4'09
Total	101.2	100.0%	53.7	100.0%	88.6%
Google*	33.3	32.9%	4.7	8.7%	615.1%
Nokia	31.0	30.6%	23.9	44.4%	30.0%
Apple	16.2	16.0%	8.7	16.3%	85.9%
RIM	14.6	14.4%	10.7	20.0%	36.0%
Microsoft	3.1	3.1%	3.9	7.2%	-20.3%
Others	3.0	2.9%	1.8	3.4%	64.8%

*Note: The Google numbers in this table relate to Android, as well as the OMS and Tapas platform variants
Source: Canalys estimates, © Canalys 2011



Worldwide Smartphone Sales to End Users by Operating System in 2010 (Thousands of Units)

Company	2010 Units	2010 Market Share (%)	2009 Units	2009 Market Share (%)
Symbian	111,576.7	37.6	80,878.3	46.9
Android	67,224.5	22.7	6,798.4	3.9
Research In Motion	47,451.6	16.0	34,346.6	19.9
iOS	46,598.3	15.7	24,889.7	14.4
Microsoft	12,378.2	4.2	15,031.0	8.7
Other Oss	11417.4	3.8	10432.1	6.1
Total	296,646.6	100.0	172,376.1	100.0

Source: Gartner (February 2011)

2011 - Why Develop for Android?

II. Android will be everywhere!

2011 - Why Develop for Android?

III. It’s open source, it’s free and it’s powerful

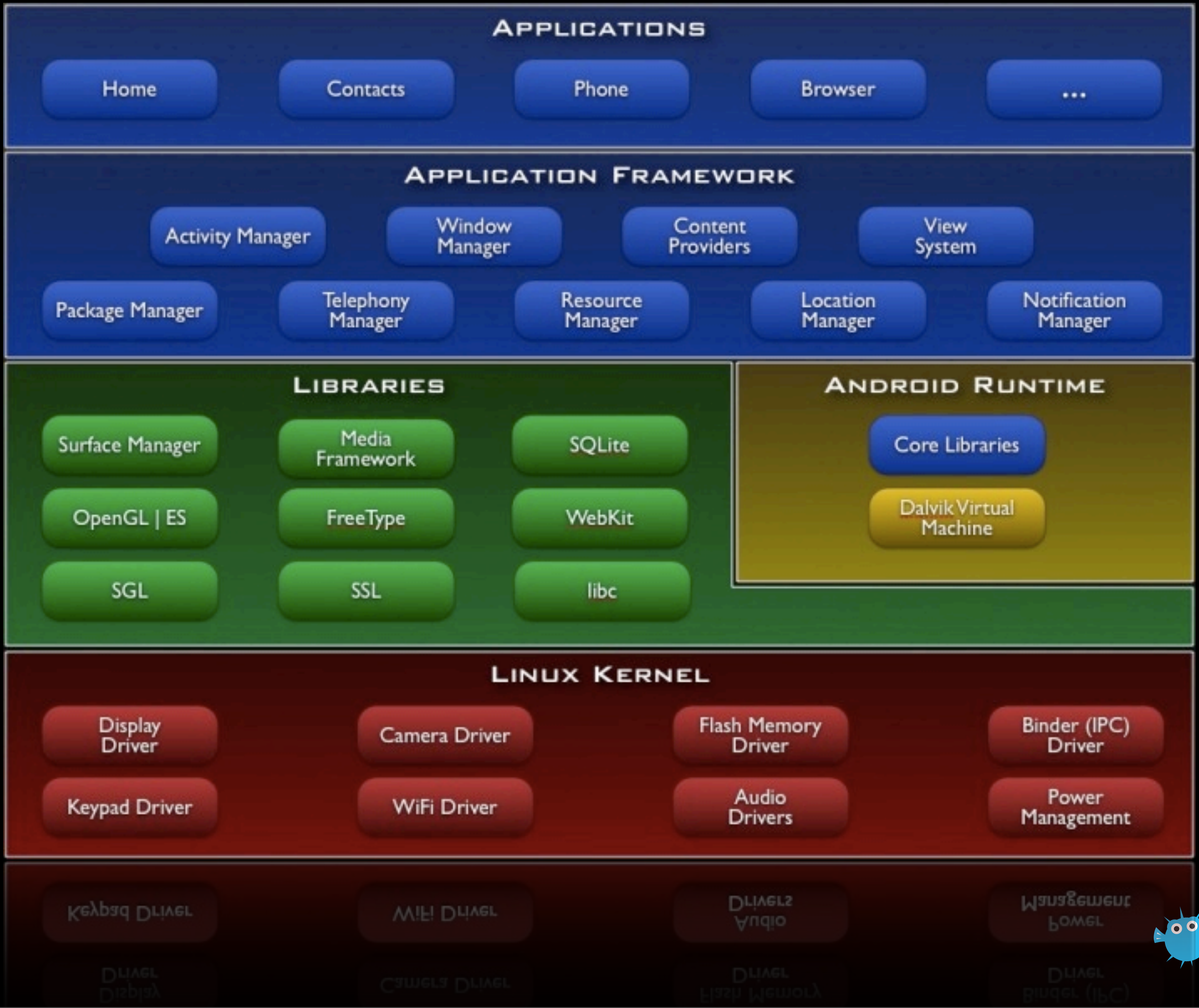
JAYWAY

2011 - Why Develop for Android?

IV. It's Java!

JAYWAY

Android Platform



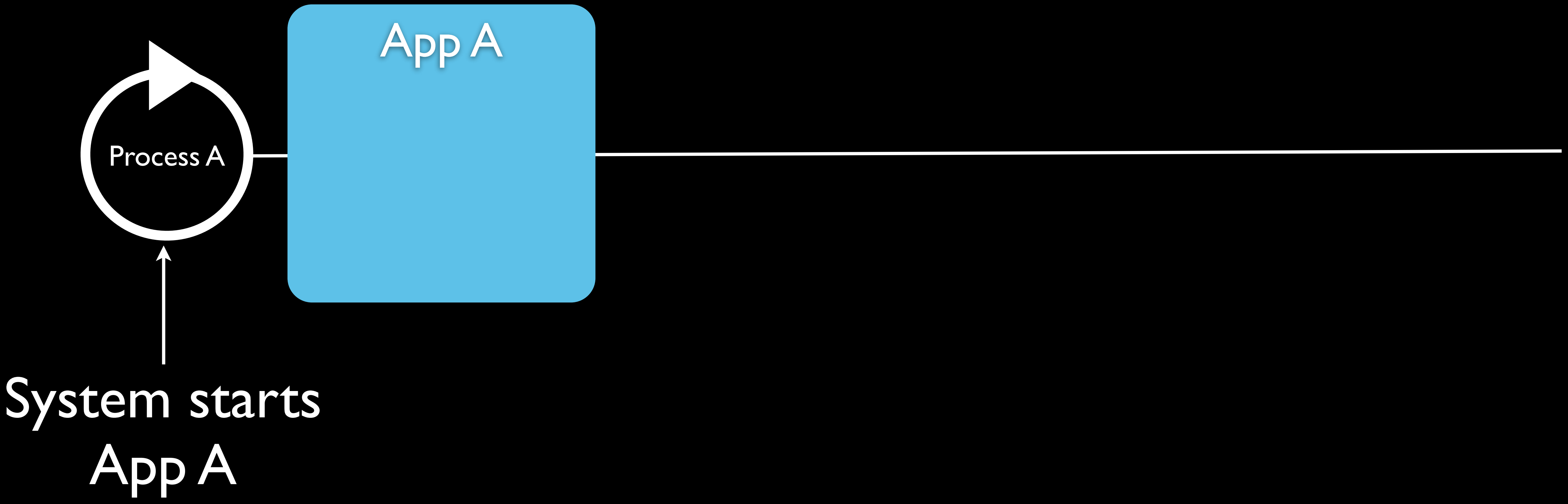
Linux Process

- Each process holds Virtual Machine
    - Application runs in a sandbox
- Applications run with distinct system identity
    - User Id
    - Isolated from other applications and platform
- Android contains an RPC mechanism for IPC

JAYWAY

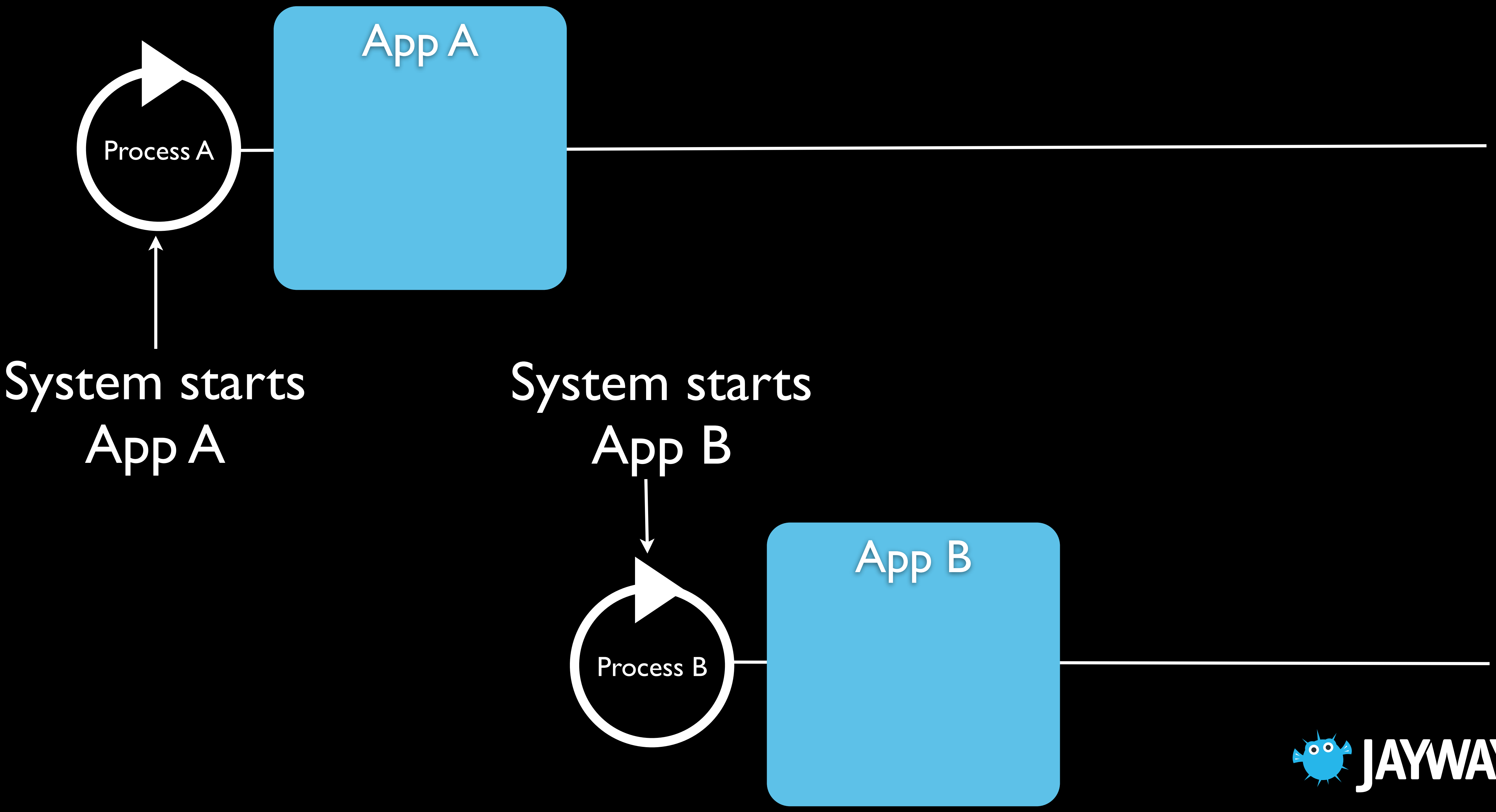
Linux Process

Default behavior



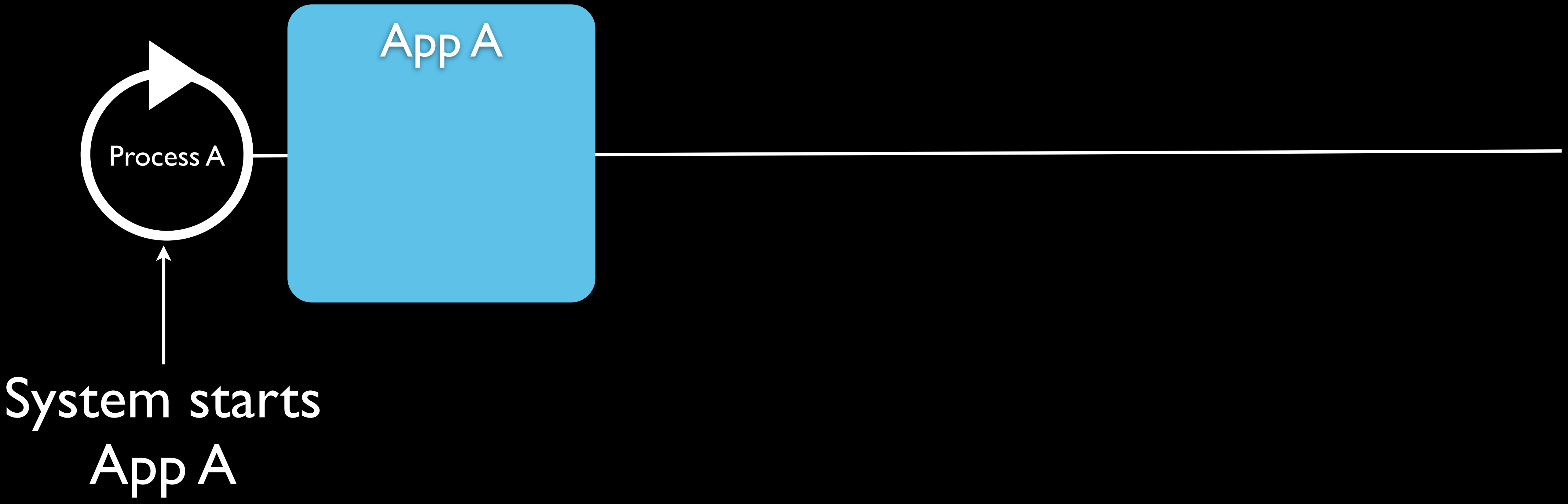
Linux Process

Default behavior



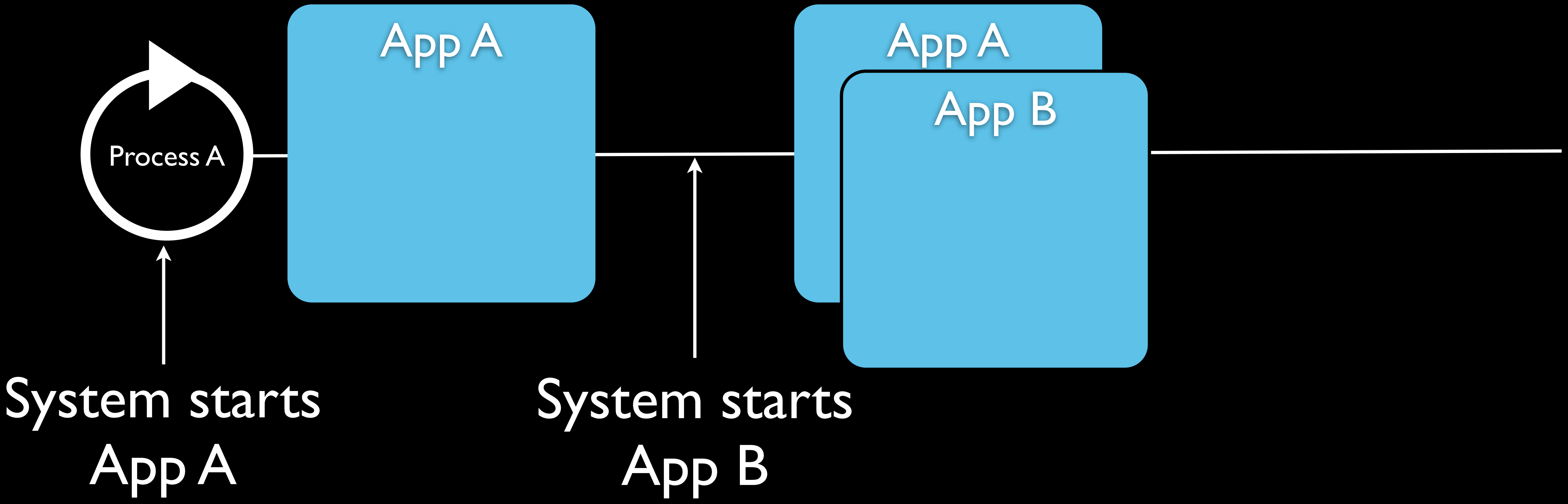
Linux Process

Configured behavior



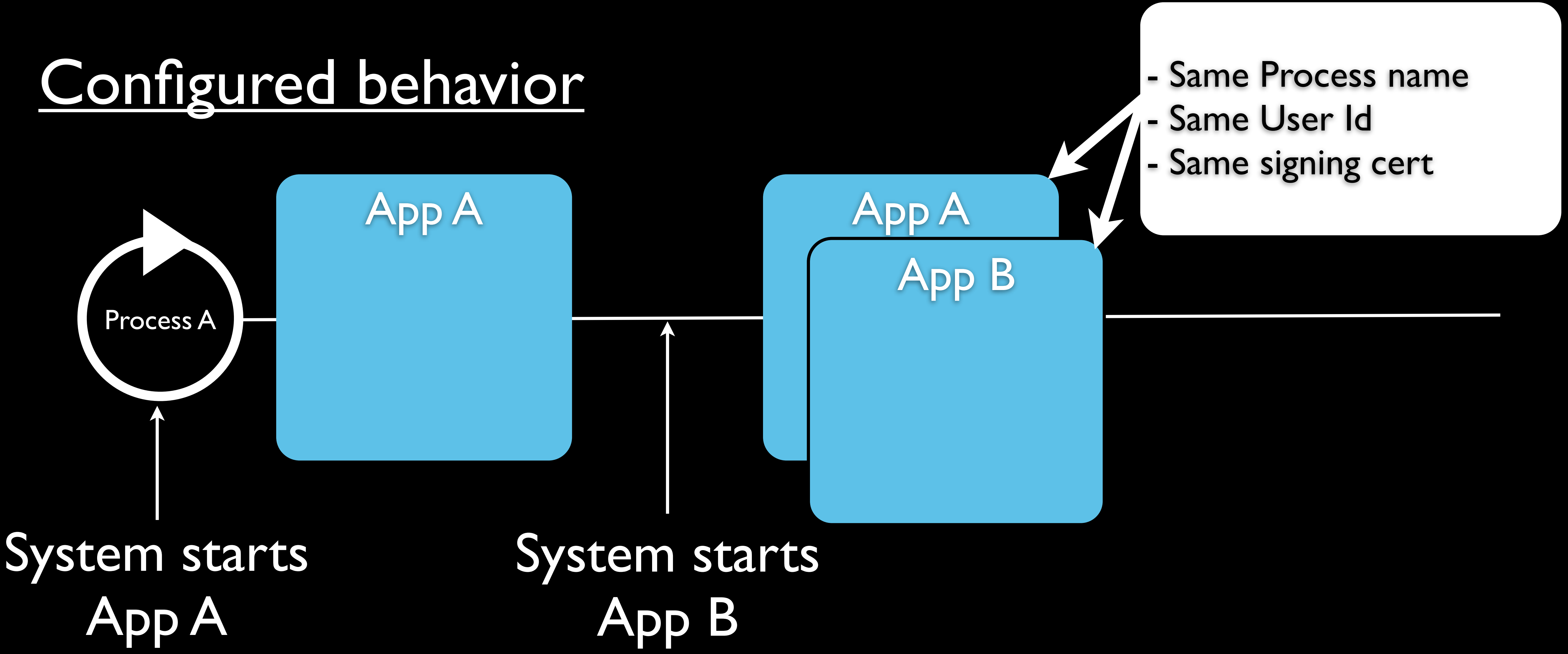
Linux Process

Configured behavior



Linux Process

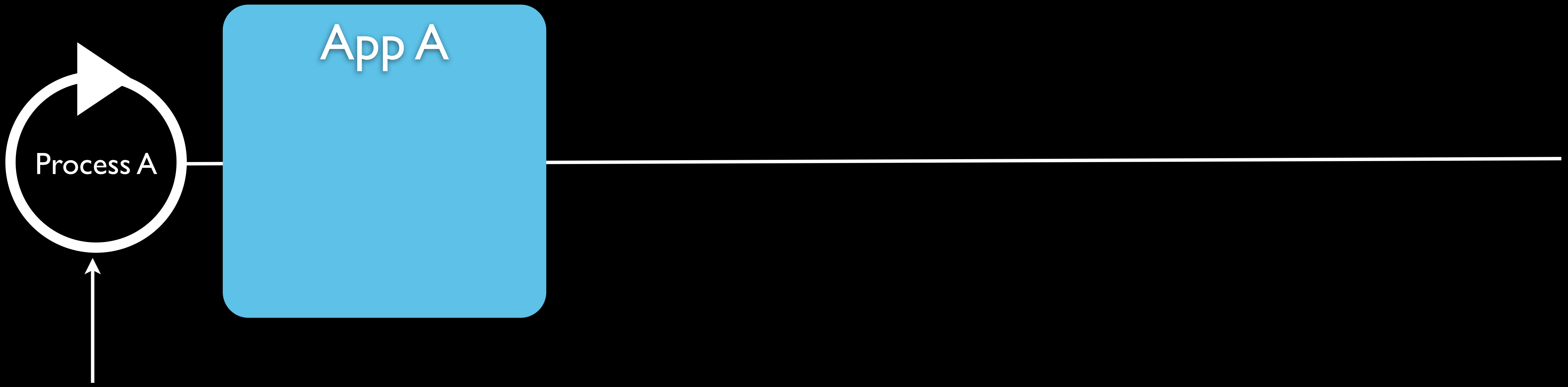
Configured behavior



JAYWAY

Linux Process

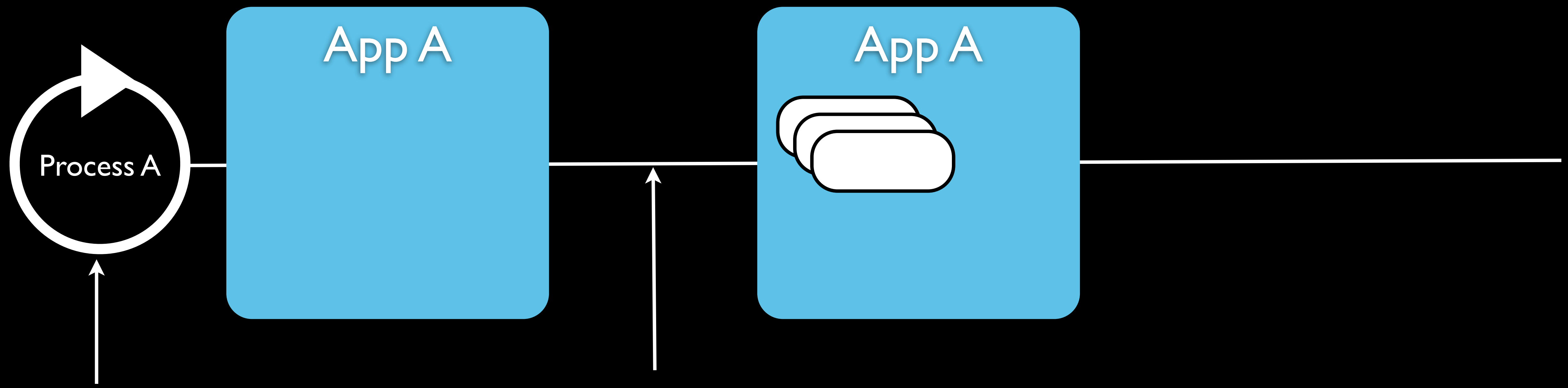
Configured behavior



System starts
App A

Linux Process

Configured behavior

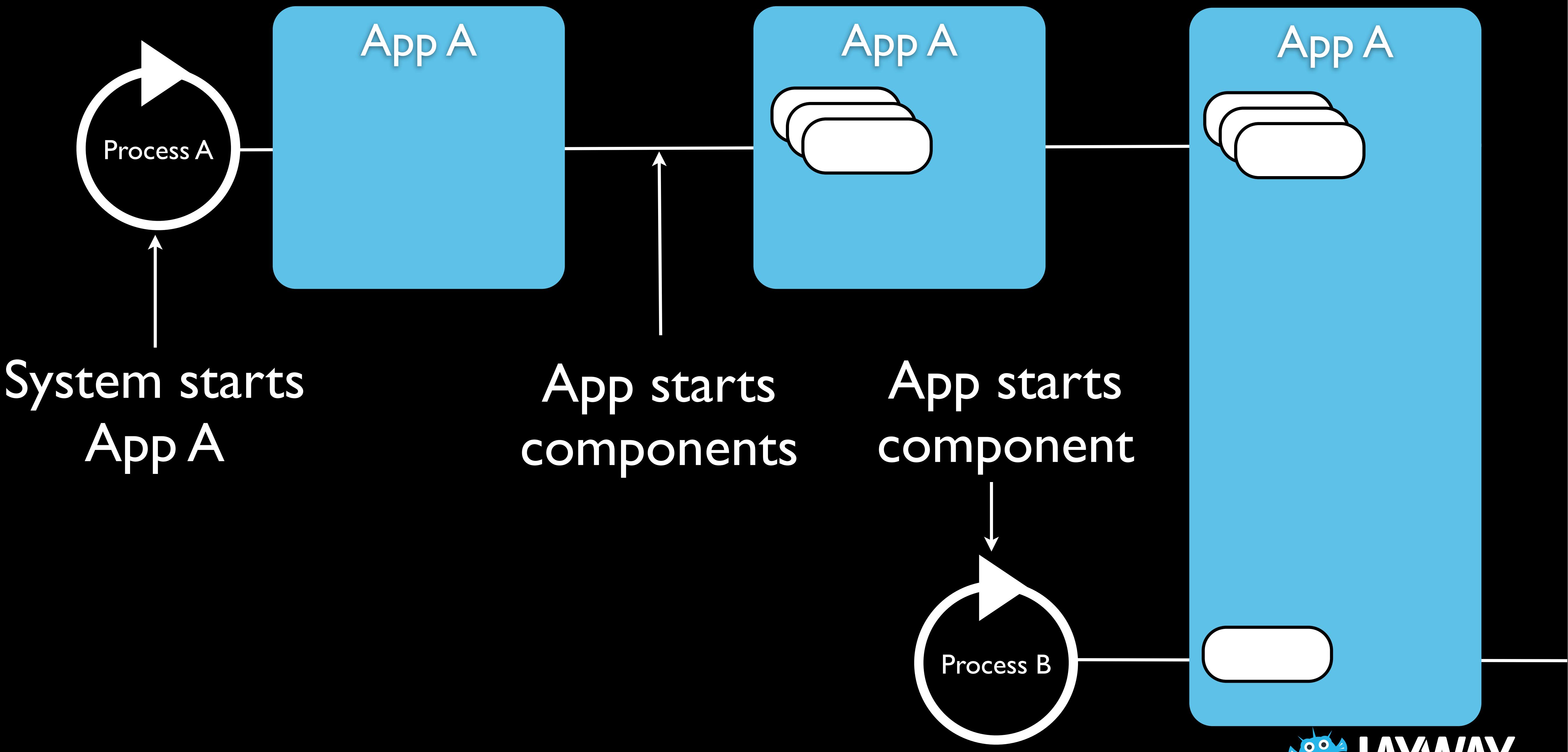


System starts App A

App starts components

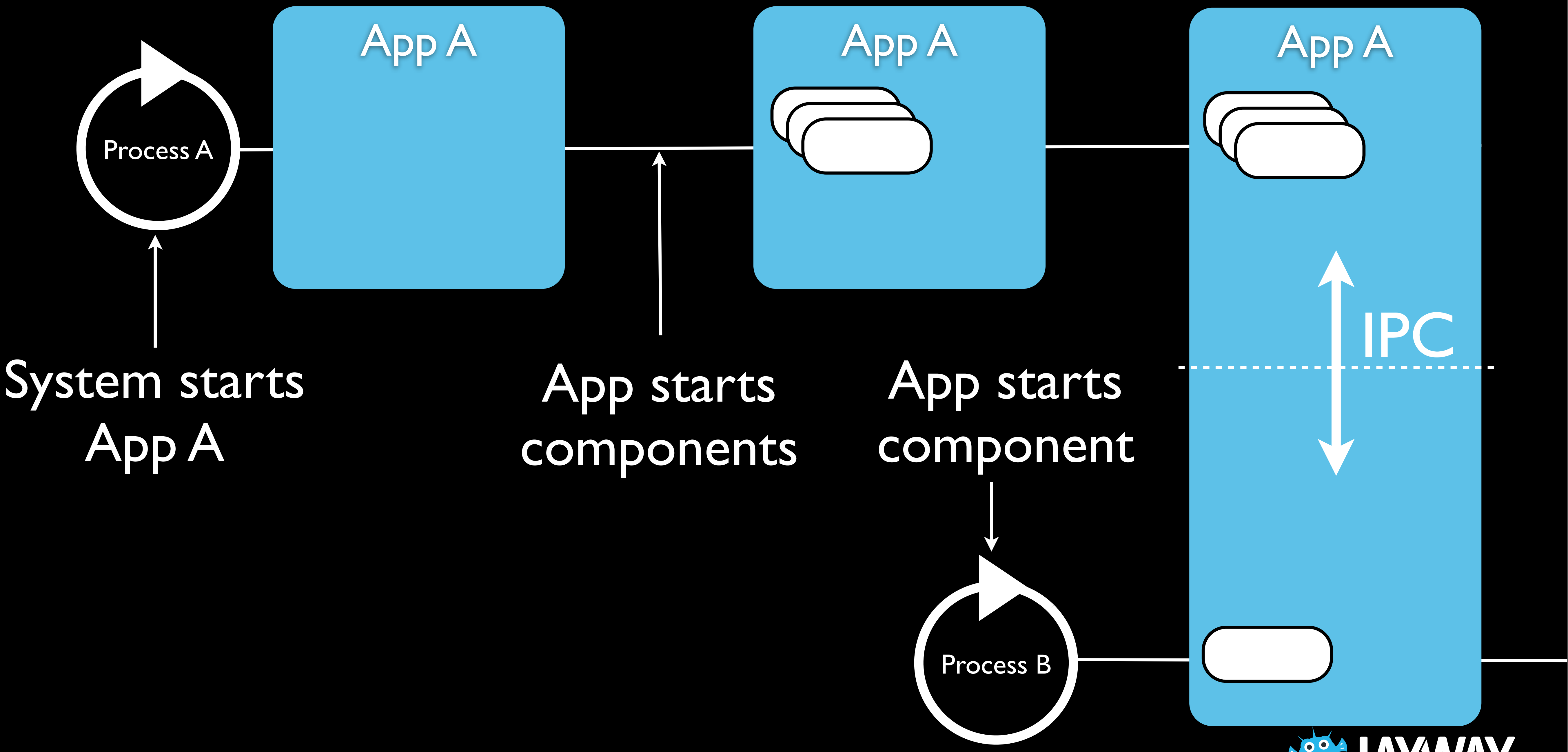
Linux Process

Configured behavior



Linux Process

Configured behavior

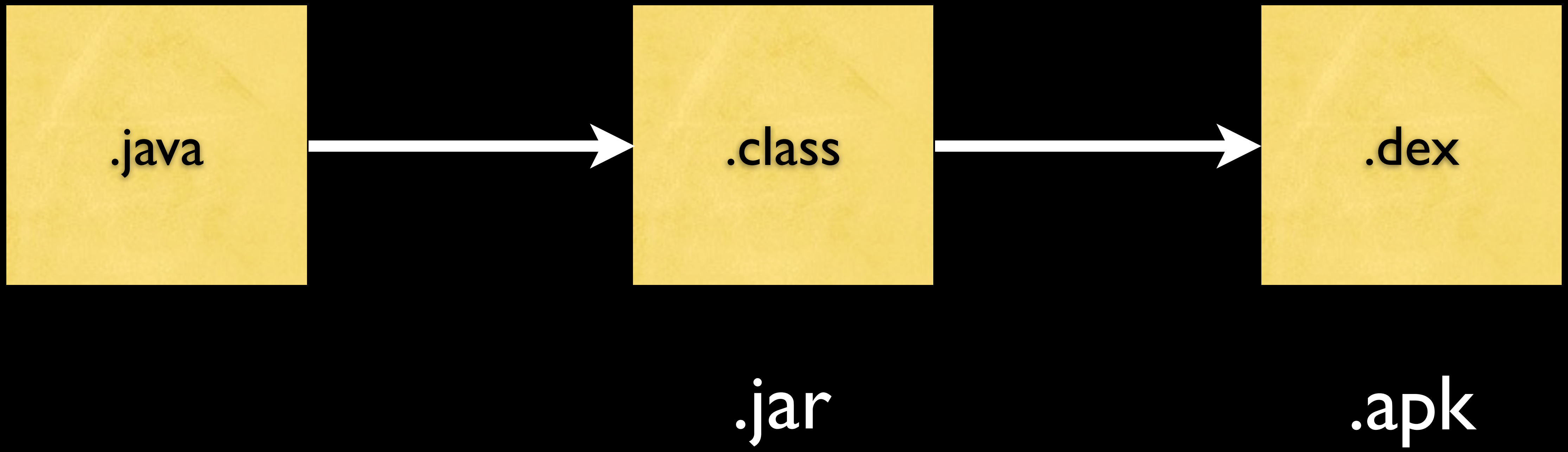


Android and Java

- Apache Harmony class libraries
- Java 5.0
- Dalvik Virtual Machine
  - Memory optimizations
  - One VM per application
  - .class-files -> .dex-files
  - 2.2/Froyo+ includes Just In Time-compilation
  - Register based (not stack based)

Android Runtime
Core Libraries
Dalvik Virtual Machine

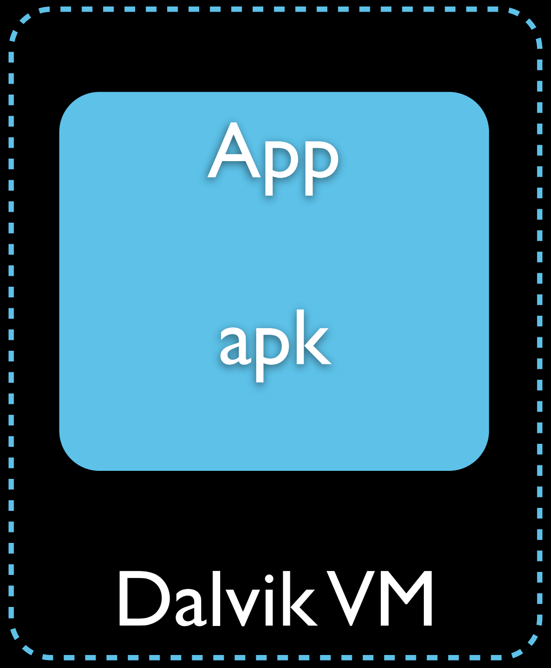
Building



Runtime

.dex
Resources
Manifest

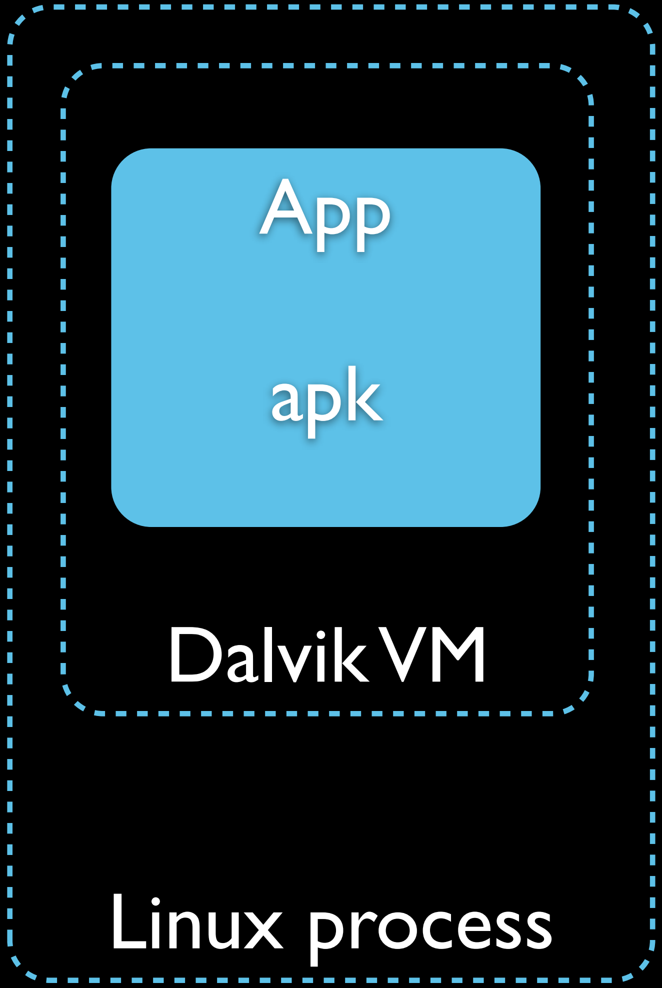
Runtime



.dex
Resources
Manifest

Optimized
Environment

Runtime



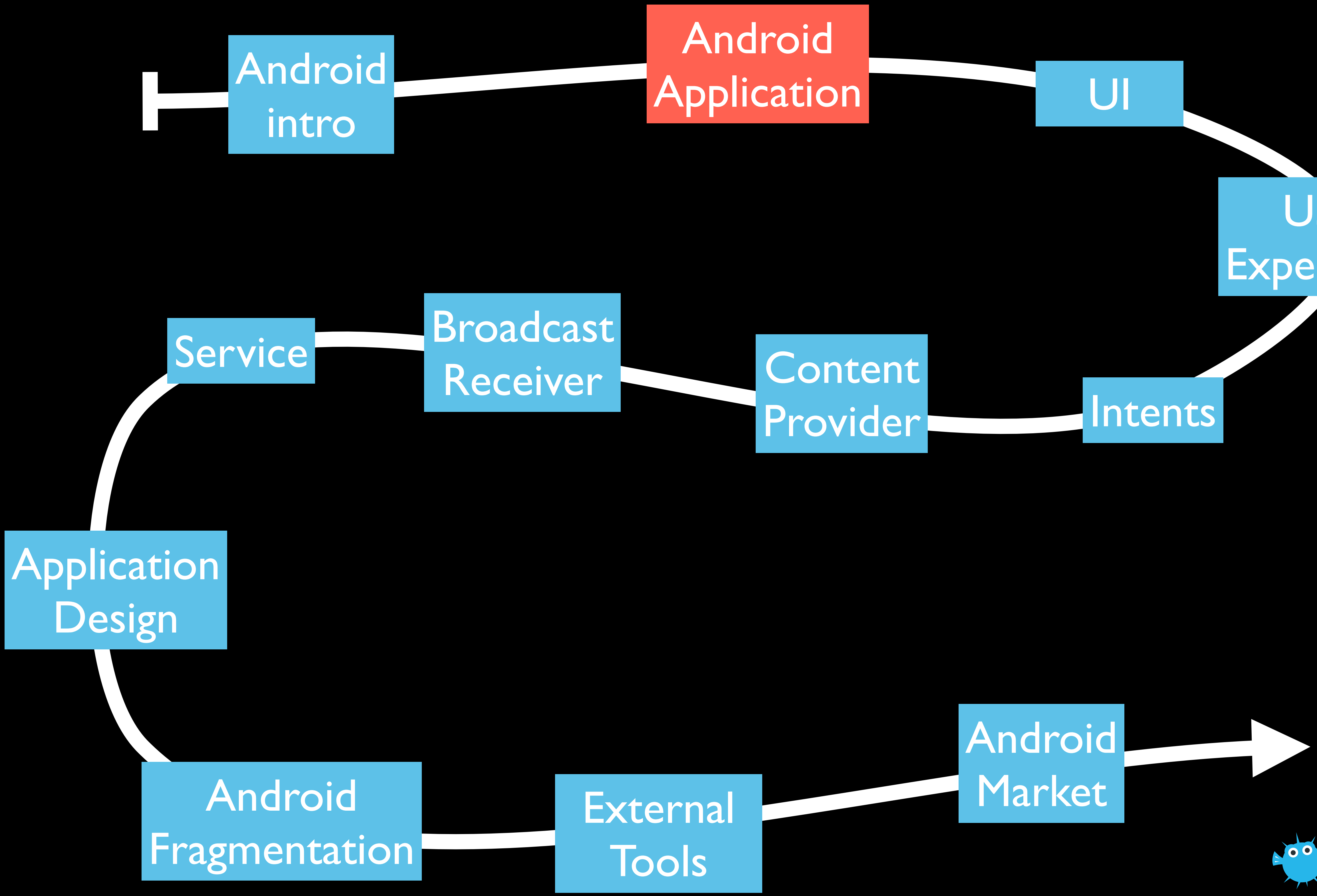
.dex
Resources
Manifest

Optimized
Environment

Linux properties
- Process Id
- User Id
- Thread model

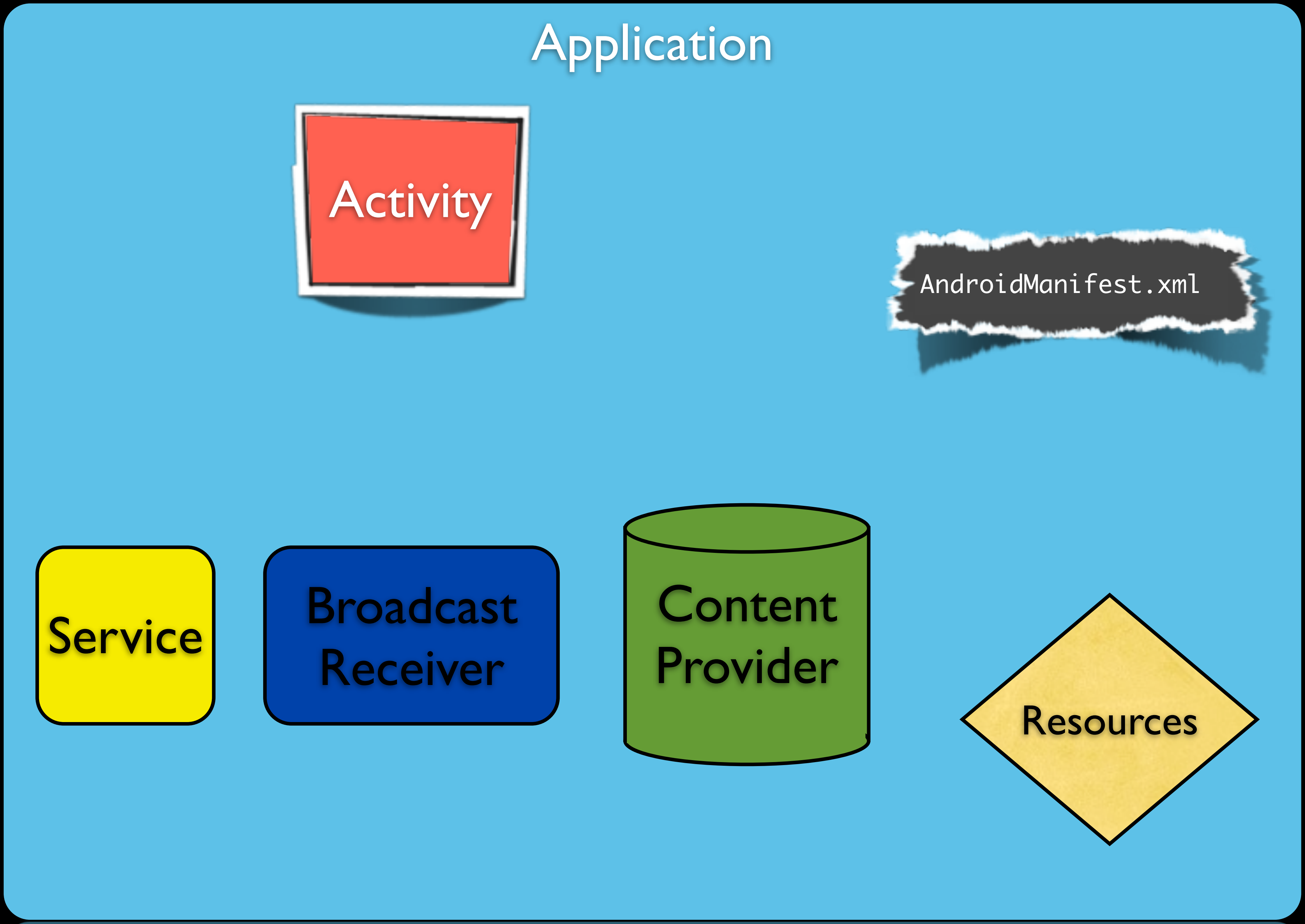
JAYWAY

Agenda



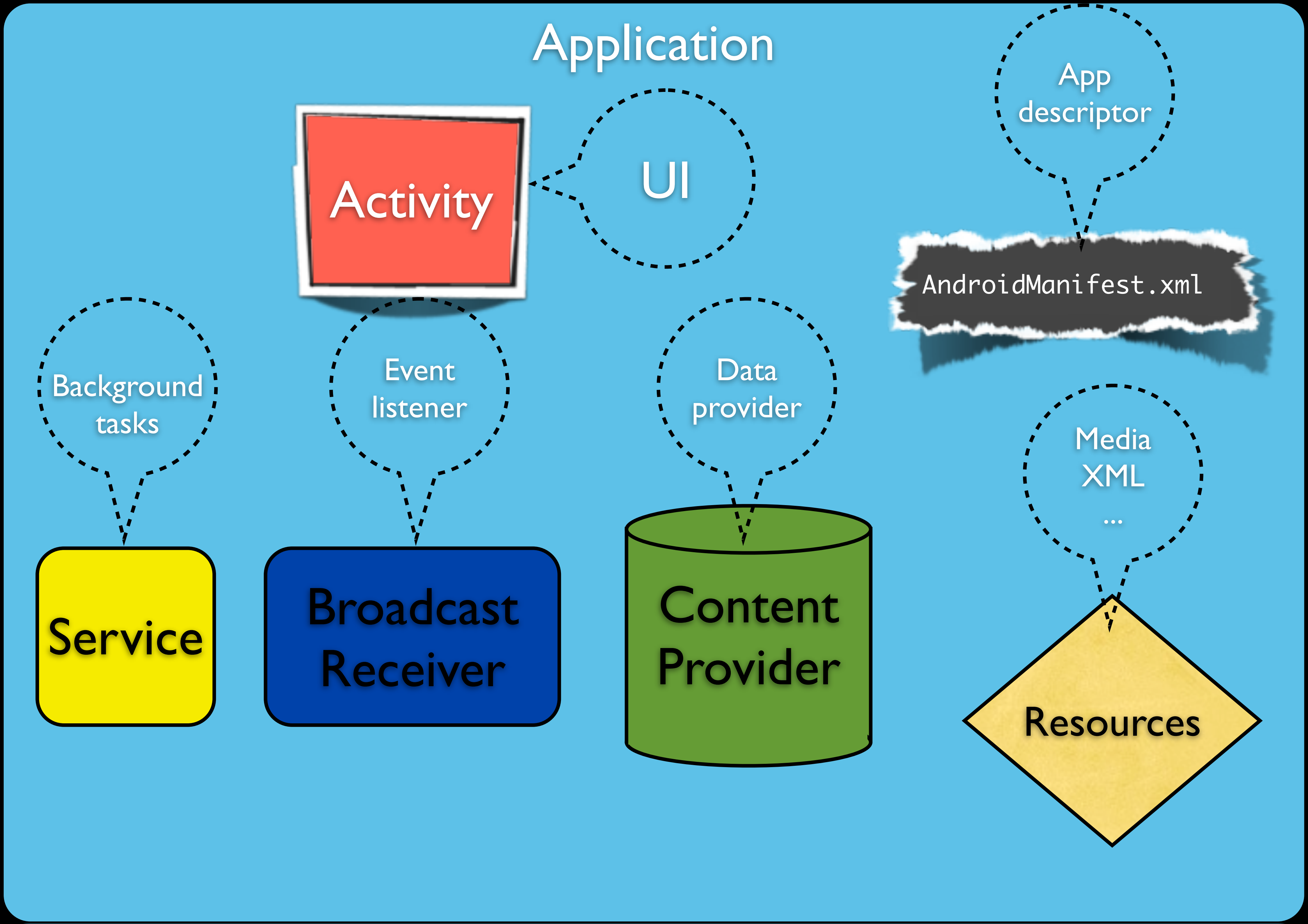
JAYWAY

Android Application

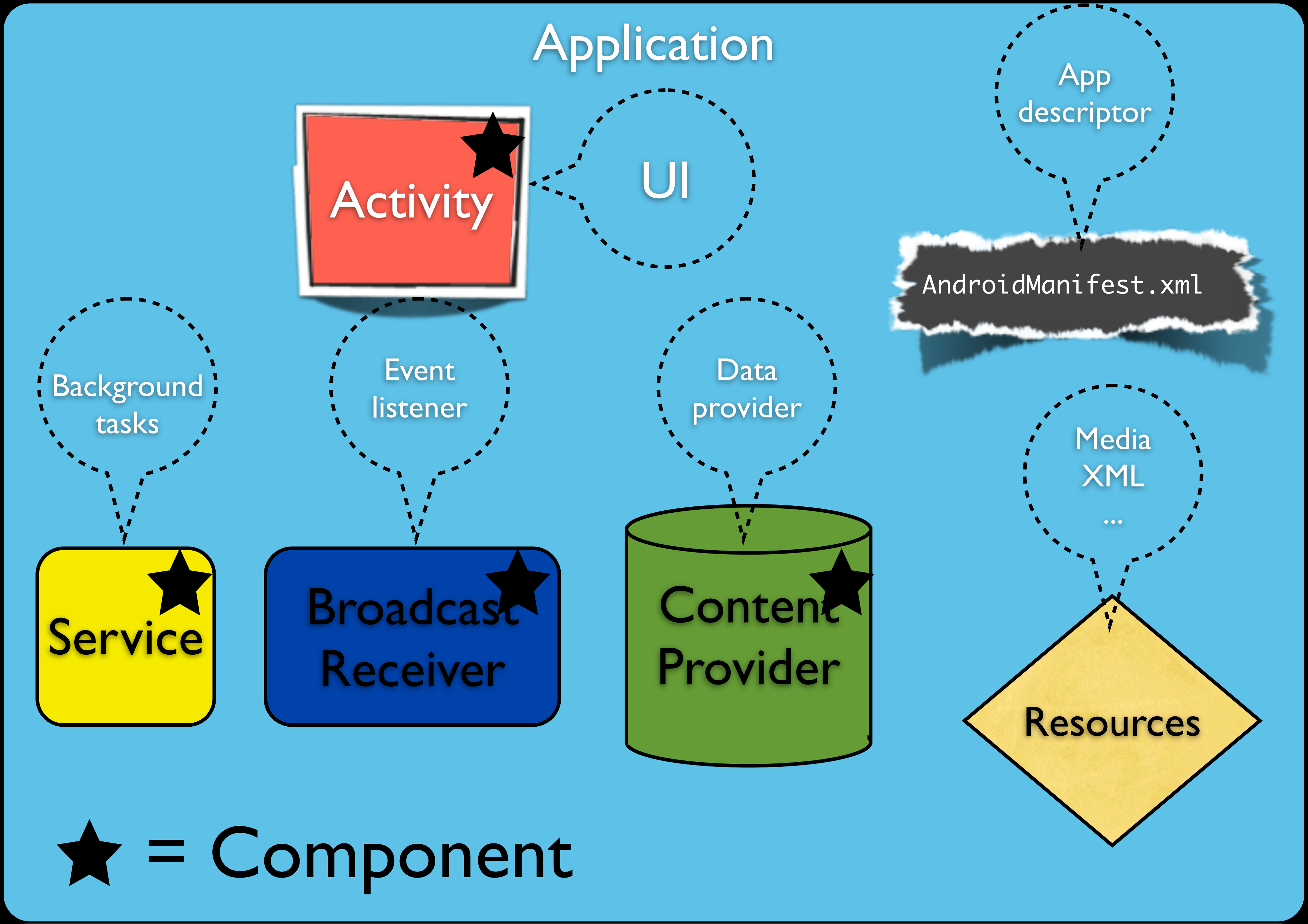


JAYWAY

Android Application

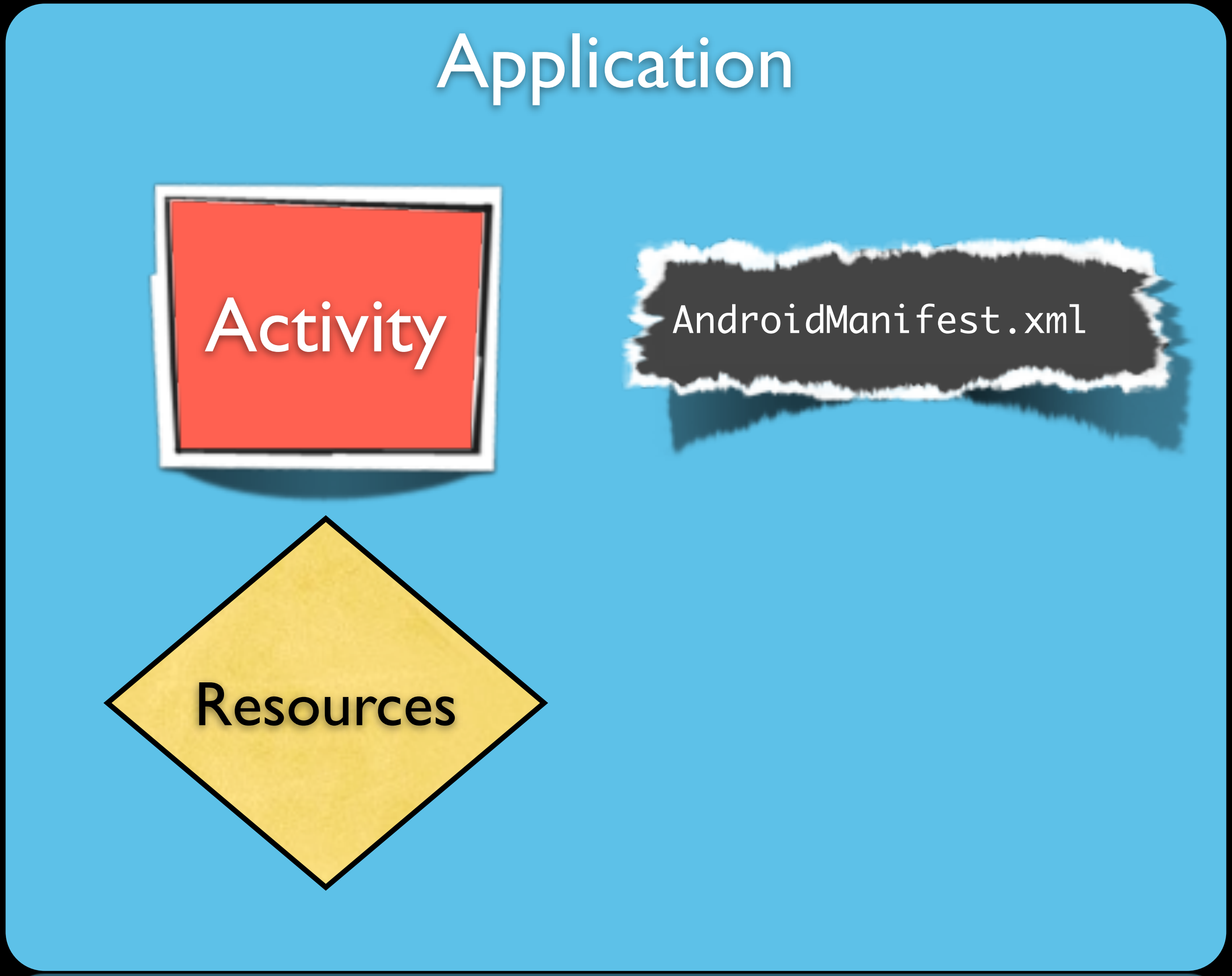


Android Application



JAYWAY

Minimum



Plain UI application

AndroidManifest.xml

```
<manifest>

    <uses-permission />
    <permission />

    <uses-sdk />
    <uses-configuration />
    <uses-feature />
    <supports-screens />

    <application>

        <activity>...</activity>

        <service>...</service>

        <receiver>...</receiver>

        <provider>...</provider>

    </application>

</manifest>
```

Security

Constraints

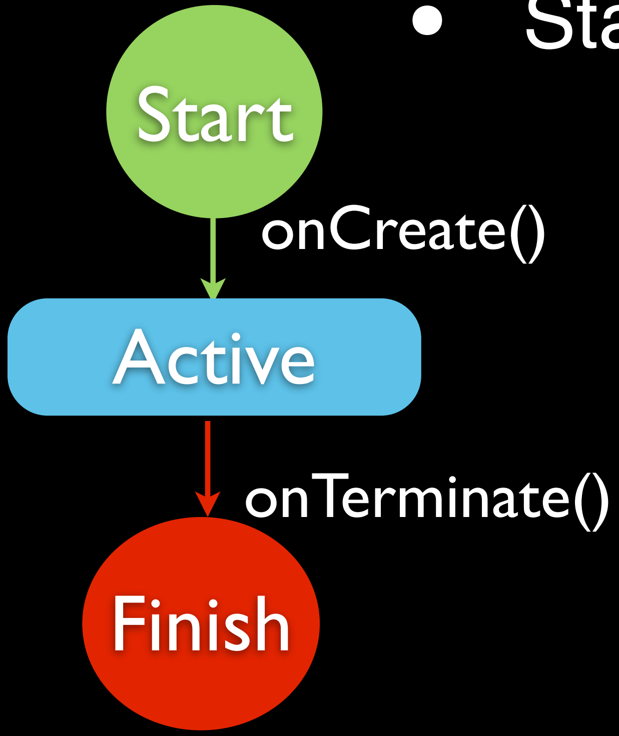
Components

JAYWAY

Application

- Base class for global application state
  - Subclass to define behavior
- Accessible from all components
- Started before all components

```
<manifest>
 <application>
    <activity>
    <service>
    <receiver>
    <provider>
 </application>
</manifest>
```



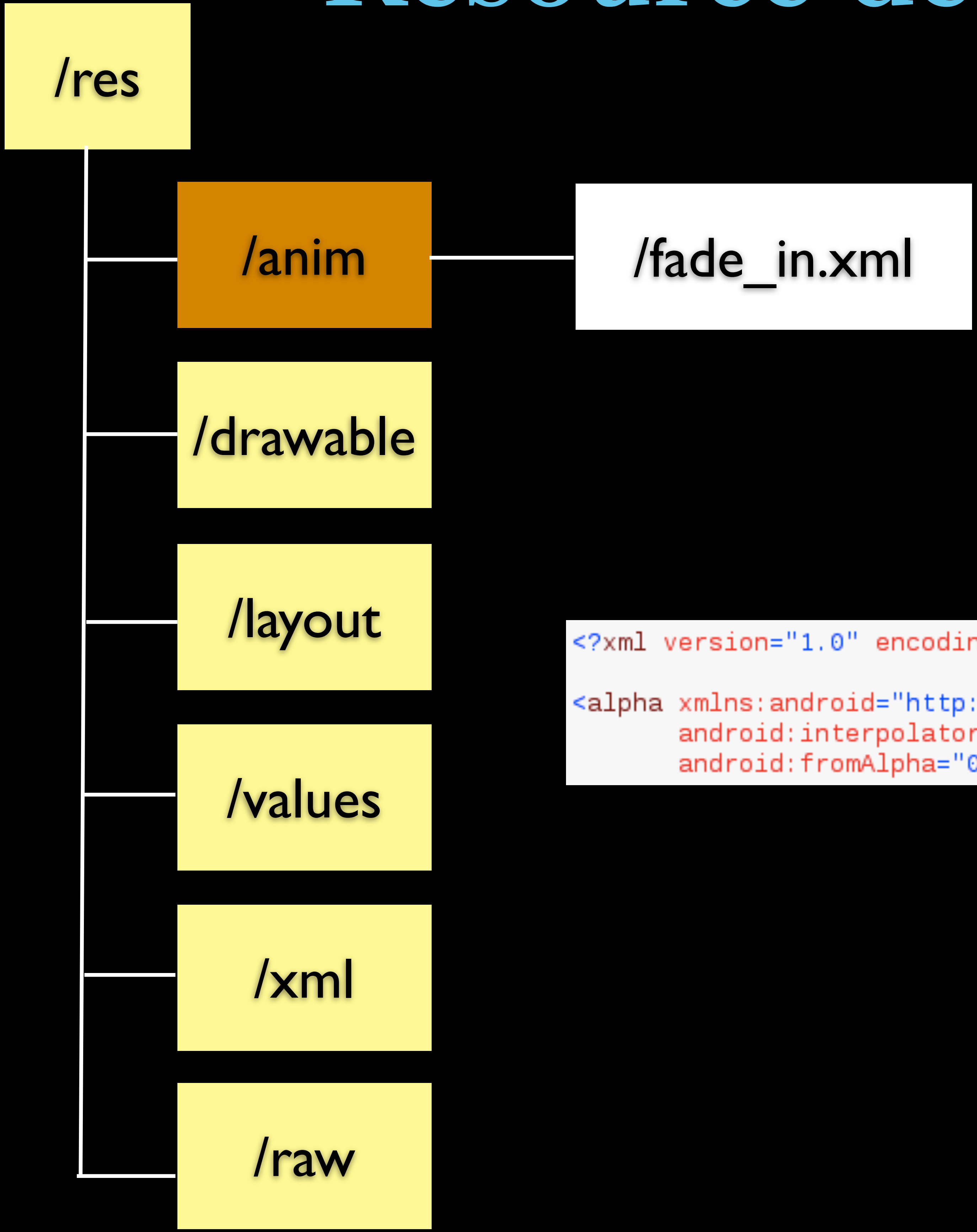
Application

- Notified when device configuration changes while the application is running
- Device configuration
  - Keyboard open/closed (HW + HW/SW)
  - User changes global setting
    - Locale, Font, etc.
- Orientation change
- MCC/MNC change

Resources

- Multimedia files
- Images
- XML -files
    - Layouts
    - Animations
    - Values (strings, dimensions, colors)
    - Menu-definitions
    - Preferences

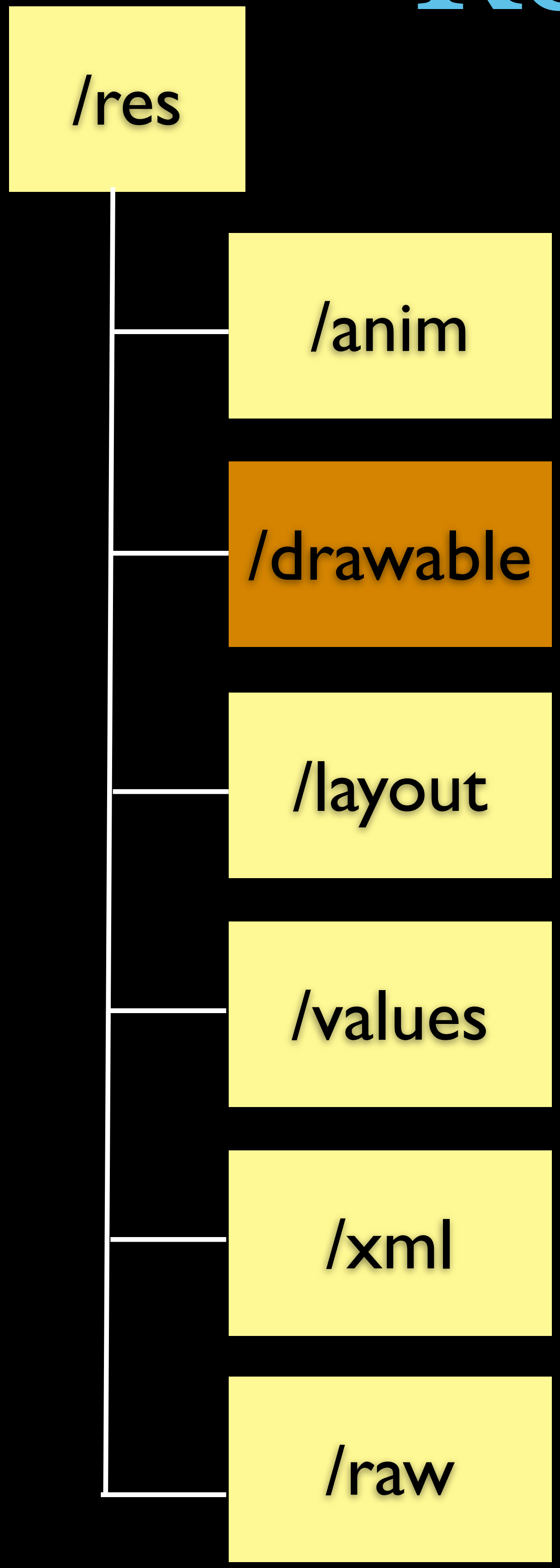
Resource definitions



```xml
<?xml version="1.0" encoding="utf-8"?>

<alpha xmlns:android="http://schemas.android.com/apk/res/android"
       android:interpolator="@android:anim/accelerate_interpolator"
       android:fromAlpha="0.0" android:toAlpha="1.0" android:duration="100" />
```

JAYWAY

Resource definitions



Bitmaps: *.png, *.jpg

Nine-patches: *.9.png

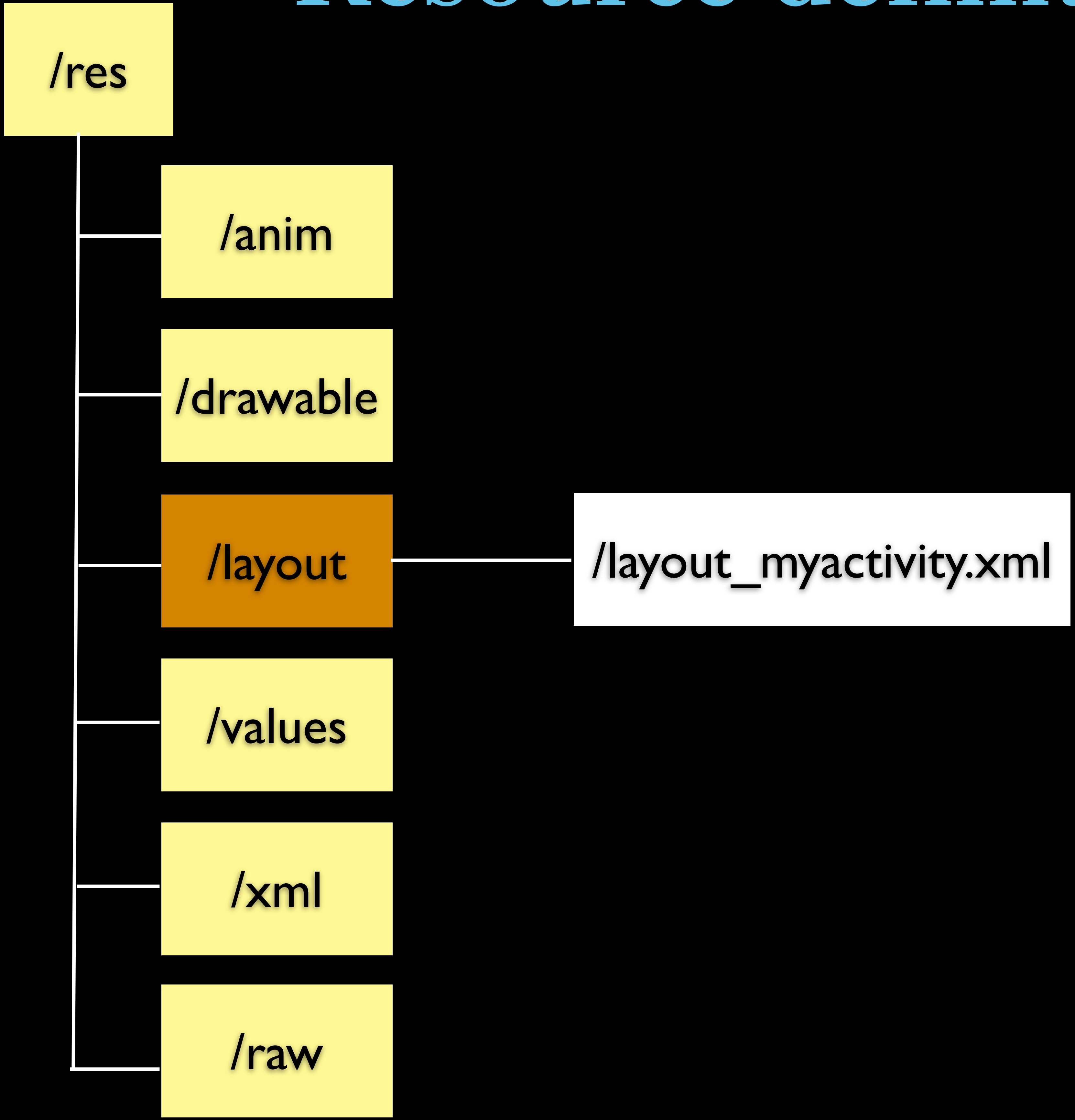
Shapes: Descriptive drawings commands, *.xml

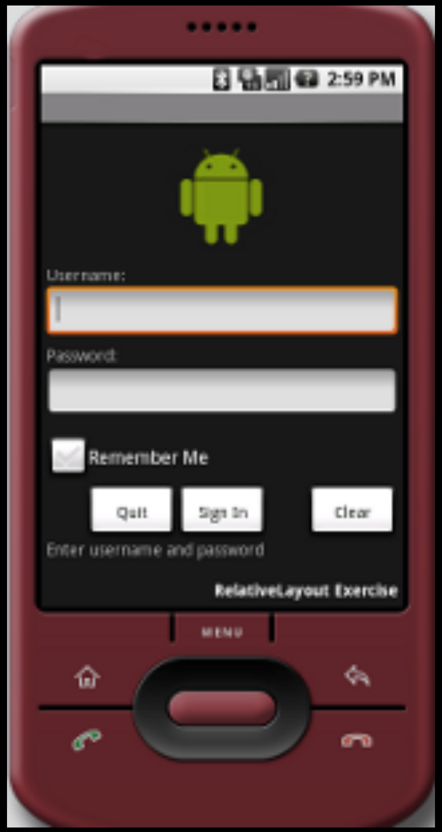
Layers: Multiple drawables compound to one, *.xml

States: Select one drawable in a list based on state, e.g. Button, *.xml

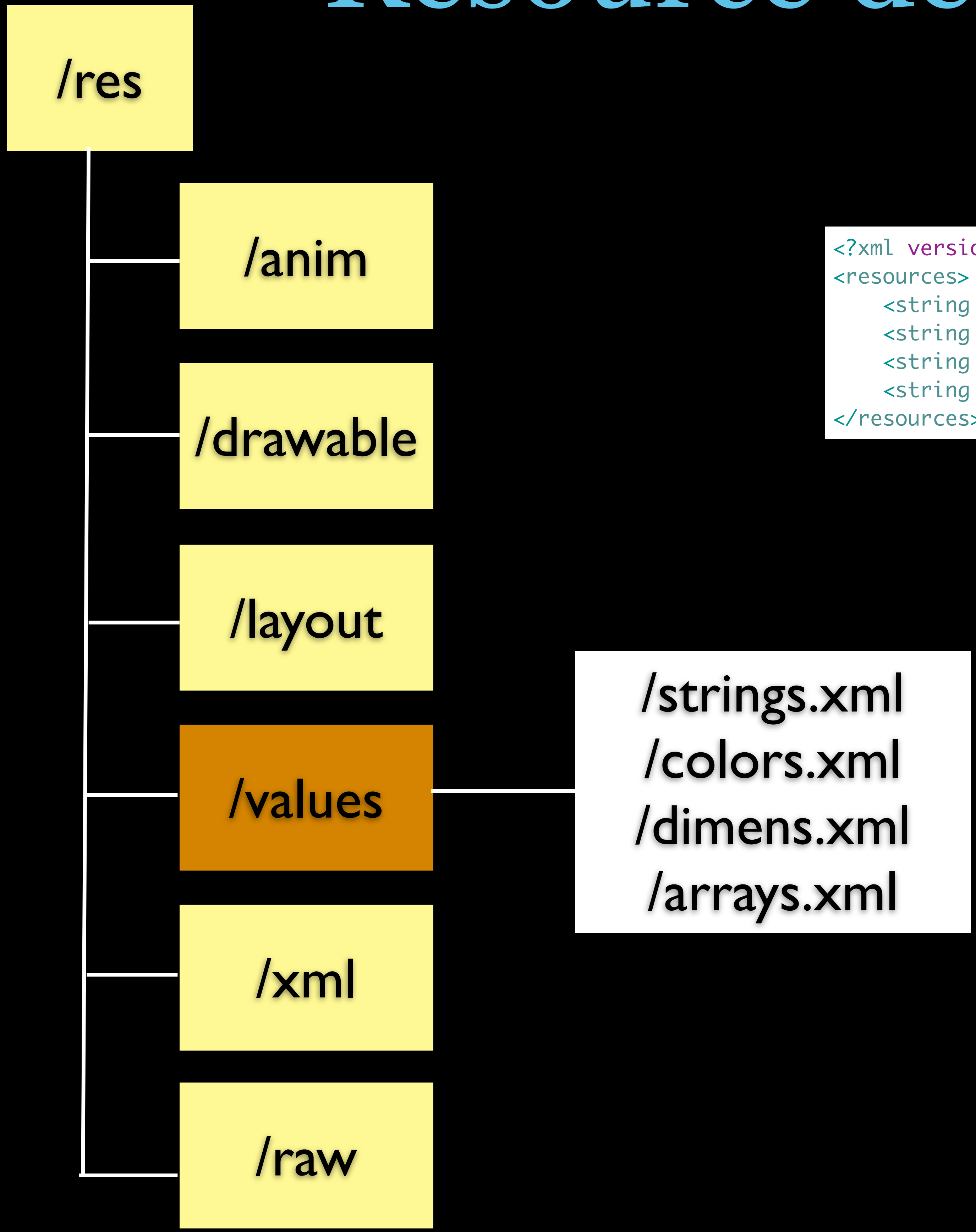
Levels: Select one drawable in a list based on level, e.g. Battery level, *.xml

Resource definitions



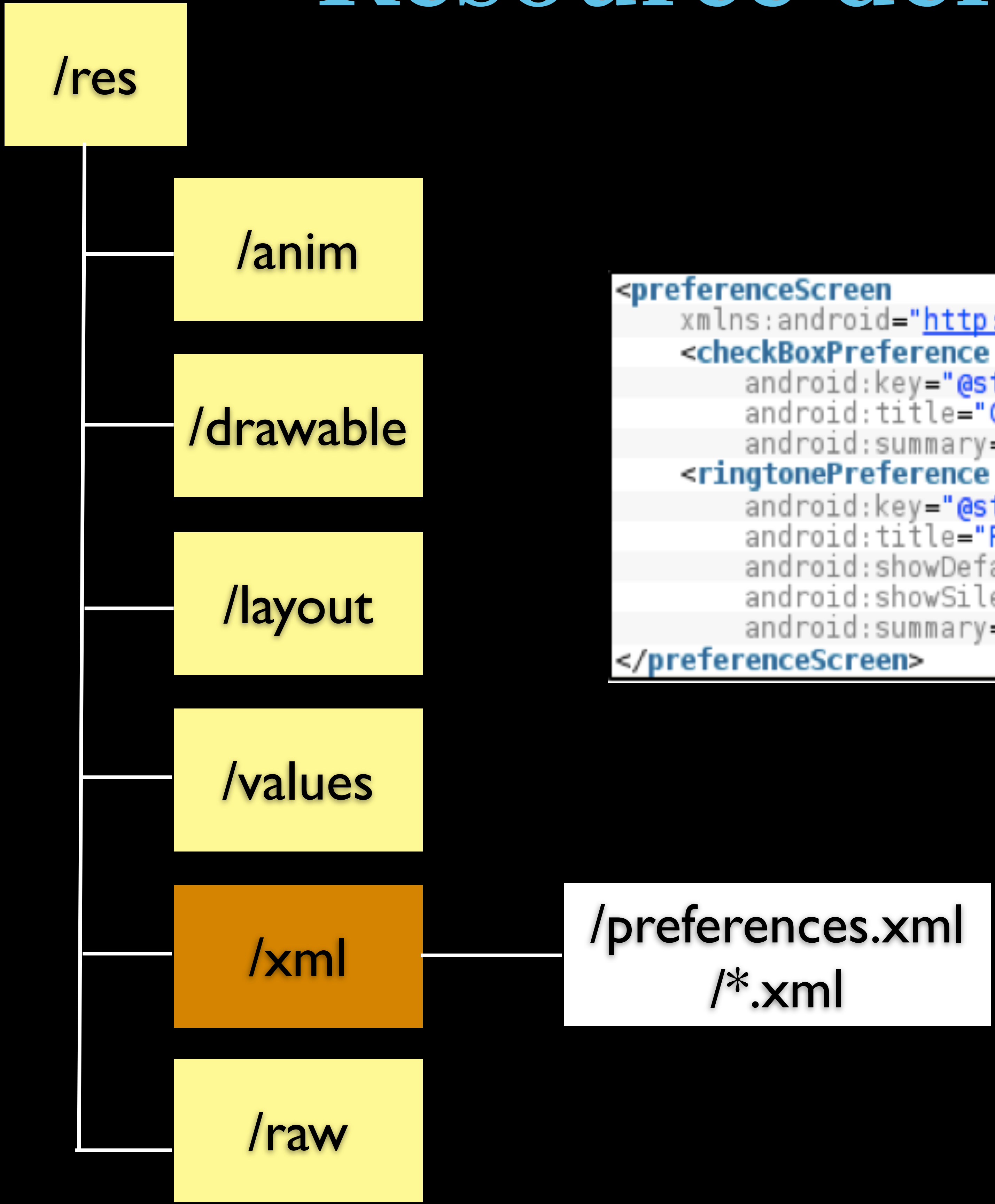


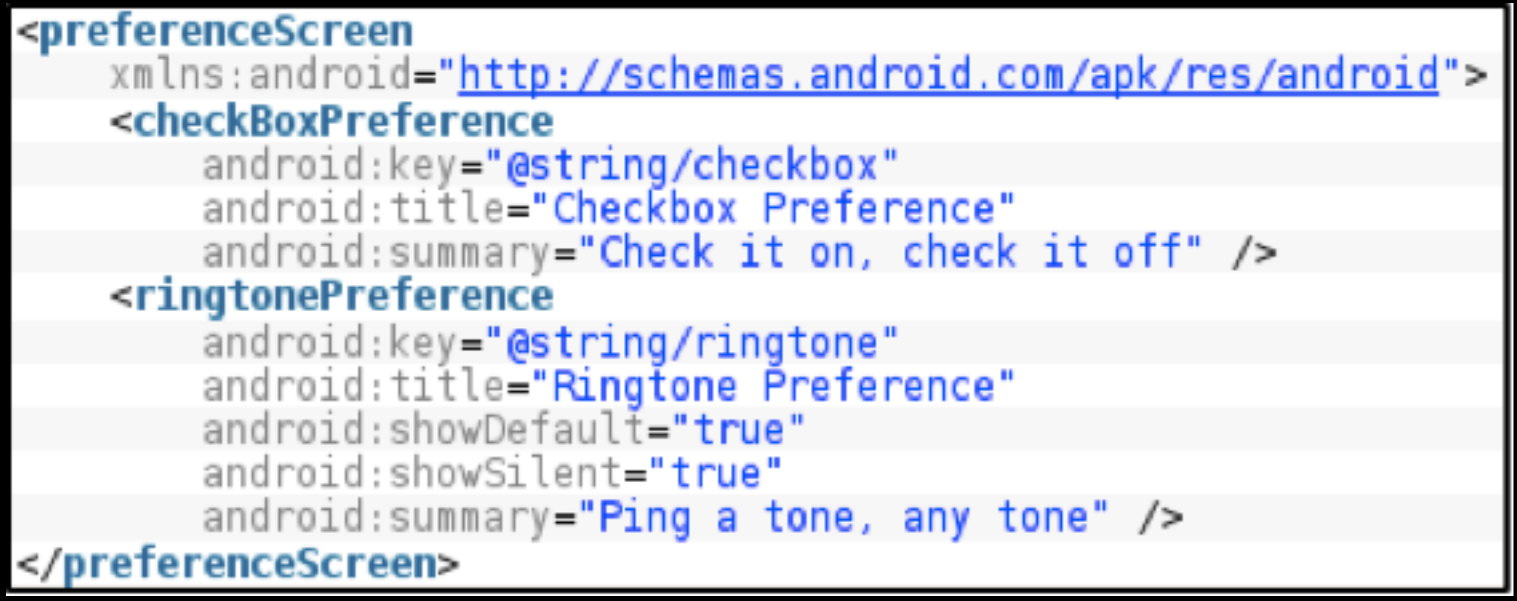
Resource definitions



```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">SpotifySearcher</string>
    <string name="artist">Artist</string>
    <string name="search">Search</string>
    <string name="search_result">Search result:</string>
</resources>
```

Resource definitions





```xml
<preferenceScreen
    xmlns:android="http://schemas.android.com/apk/res/android">
    <checkBoxPreference
        android:key="@string/checkbox"
        android:title="Checkbox Preference"
        android:summary="Check it on, check it off" />
    <ringtonePreference
        android:key="@string/ringtone"
        android:title="Ringtone Preference"
        android:showDefault="true"
        android:showSilent="true"
        android:summary="Ping a tone, any tone" />
</preferenceScreen>
```

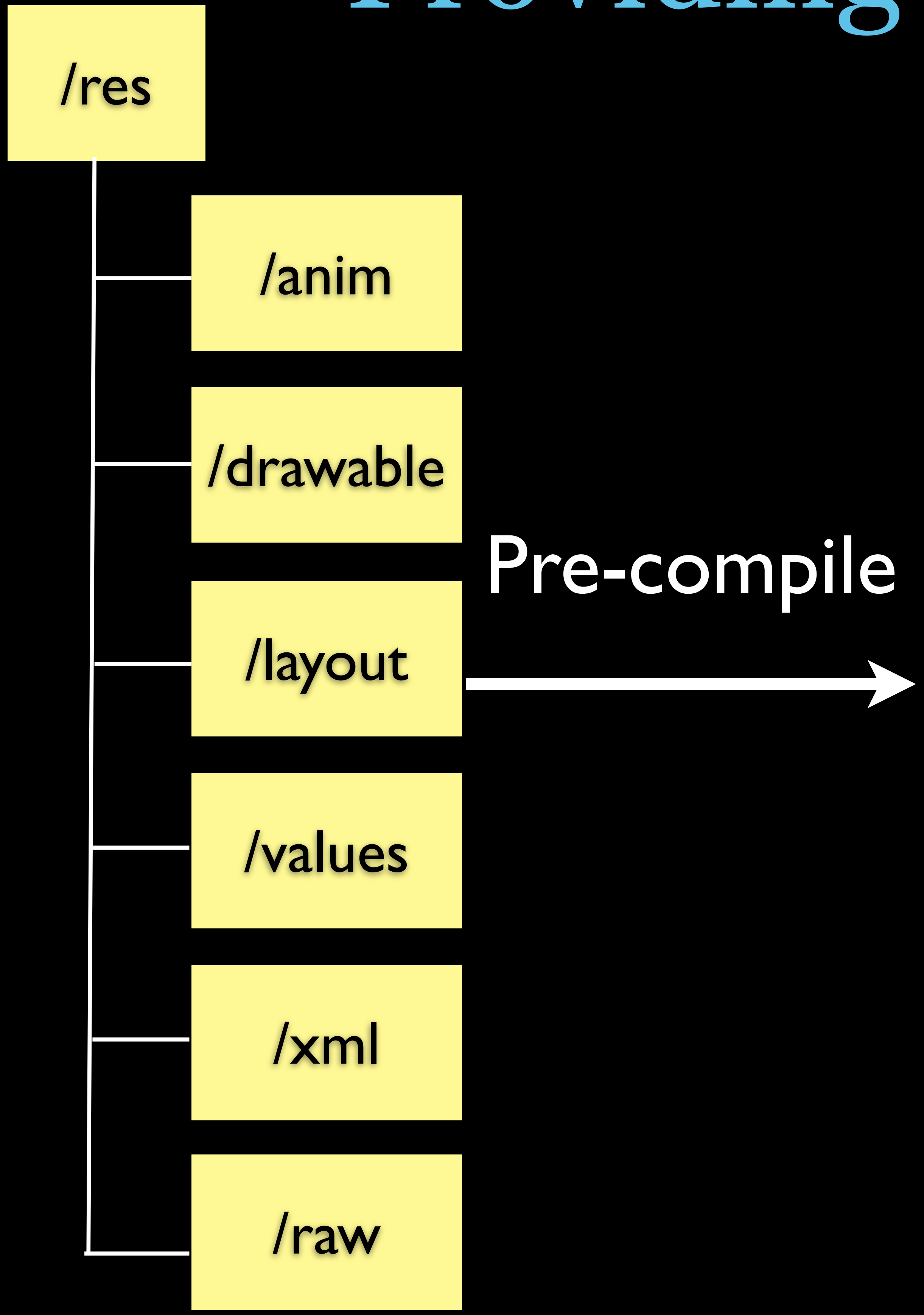
JAYWAY

Resource definitions

- /res
  - /anim
  - /drawable
  - /layout
  - /values
  - /xml
  - /raw — Arbitrary files

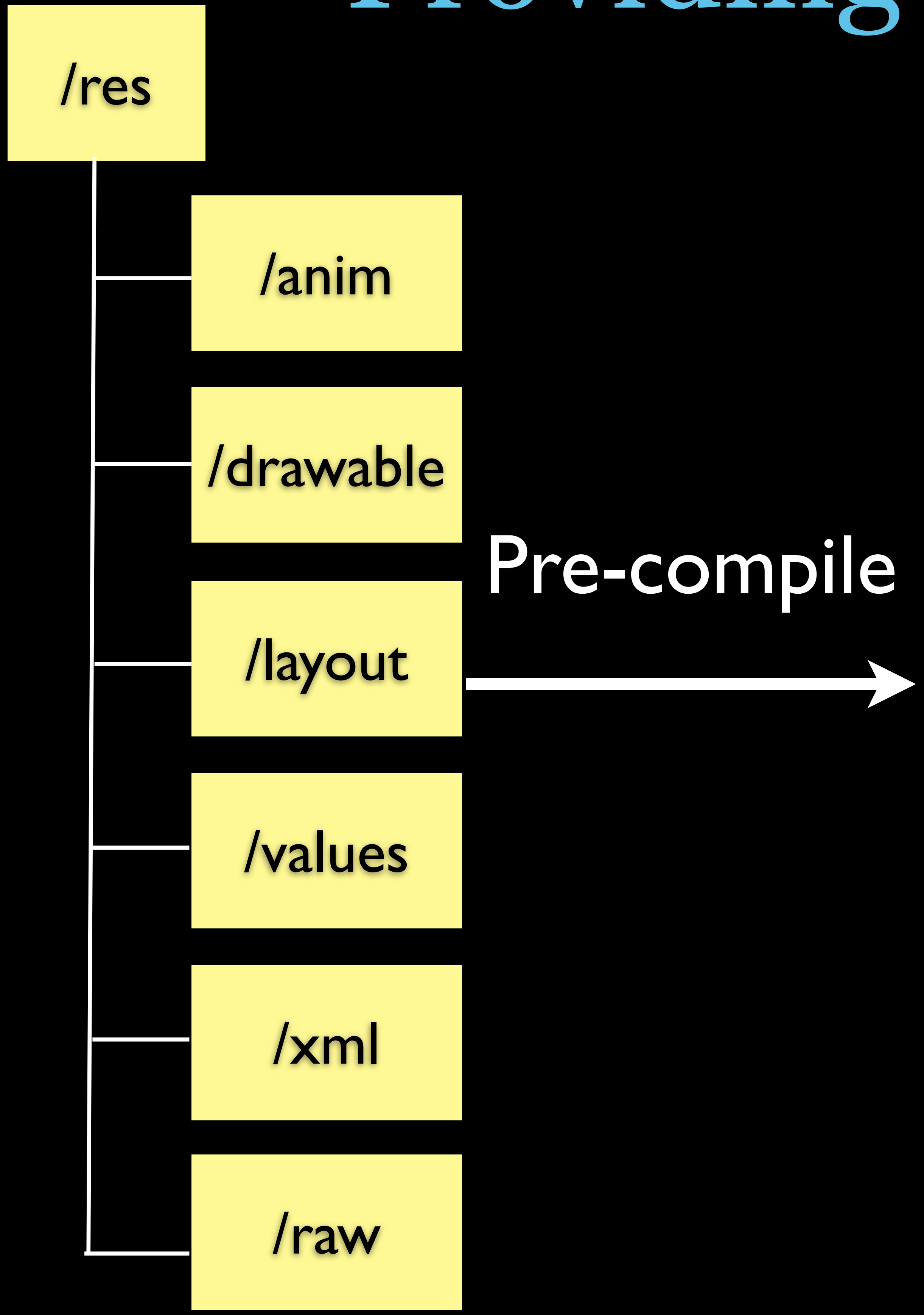
JAYWAY

Providing Resources



```
public final class R {
    public static final class attr {
    }
    public static final class color {
        public static final int dark_grey=0x7f040001;
        public static final int white=0x7f040000;
    }
    public static final class dimen {
        public static final int margin_vertical_default=0x7f050001;
        public static final int padding_default=0x7f050000;
        public static final int text_size_default=0x7f050002;
        public static final int text_size_large=0x7f050003;
    }
    public static final class drawable {
        public static final int icon=0x7f020000;
    }
    public static final class id {
        public static final int button_search=0x7f070001;
        public static final int edit_search_input=0x7f070000;
        public static final int text_search_result=0x7f070002;
    }
    public static final class layout {
        public static final int activity_search=0x7f030000;
        public static final int divider_line=0x7f030001;
    }
    public static final class string {
        public static final int app_name=0x7f060000;
        public static final int artist=0x7f060001;
        public static final int search=0x7f060002;
        public static final int search_result=0x7f060003;
    }
}
```

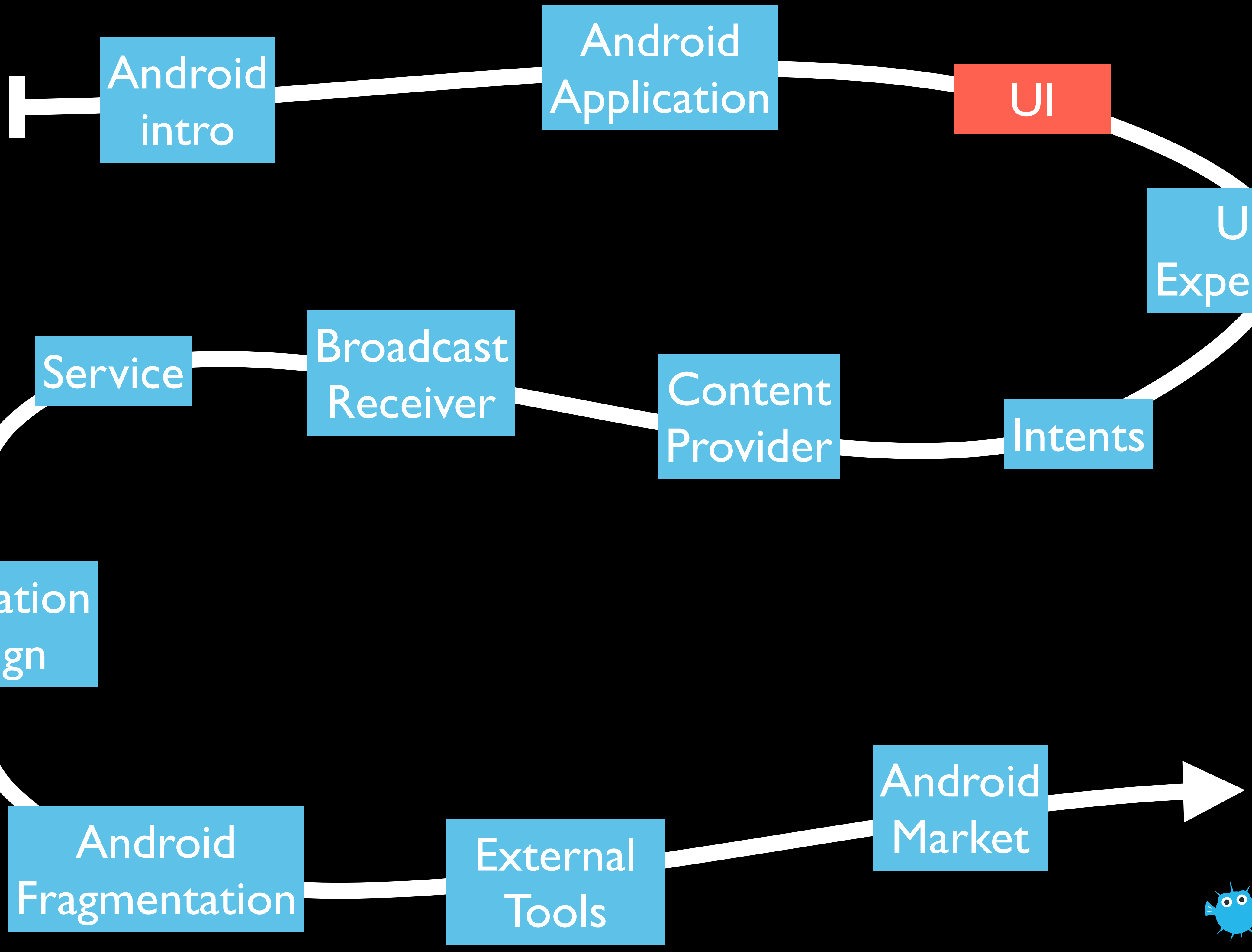
Providing Resources



R.java

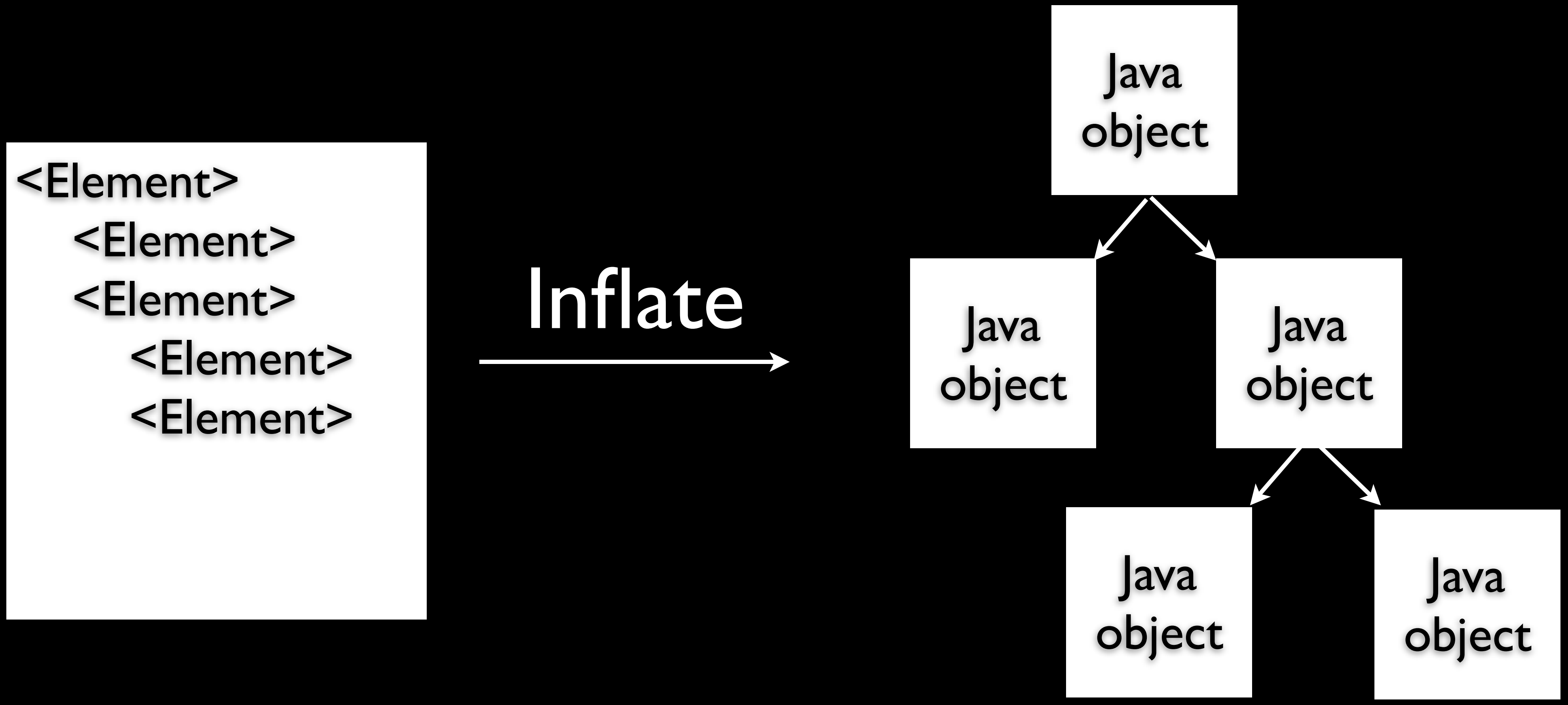
```
public final class R {
    public static final class attr {
    }
    public static final class color {
        public static final int dark_grey=0x7f040001;
        public static final int white=0x7f040000;
    }
    public static final class dimen {
        public static final int margin_vertical_default=0x7f050001;
        public static final int padding_default=0x7f050000;
        public static final int text_size_default=0x7f050002;
        public static final int text_size_large=0x7f050003;
    }
    public static final class drawable {
        public static final int icon=0x7f020000;
    }
    public static final class id {
        public static final int button_search=0x7f070001;
        public static final int edit_search_input=0x7f070000;
        public static final int text_search_result=0x7f070002;
    }
    public static final class layout {
        public static final int activity_search=0x7f030000;
        public static final int divider_line=0x7f030001;
    }
    public static final class string {
        public static final int app_name=0x7f060000;
        public static final int artist=0x7f060001;
        public static final int search=0x7f060002;
        public static final int search_result=0x7f060003;
    }
}
```

Agenda


Android intro
Android Application
UI
User Experience
Intents
Content Provider
Broadcast Receiver
Service
Application Design
Android Fragmentation
External Tools
Android Market

UI

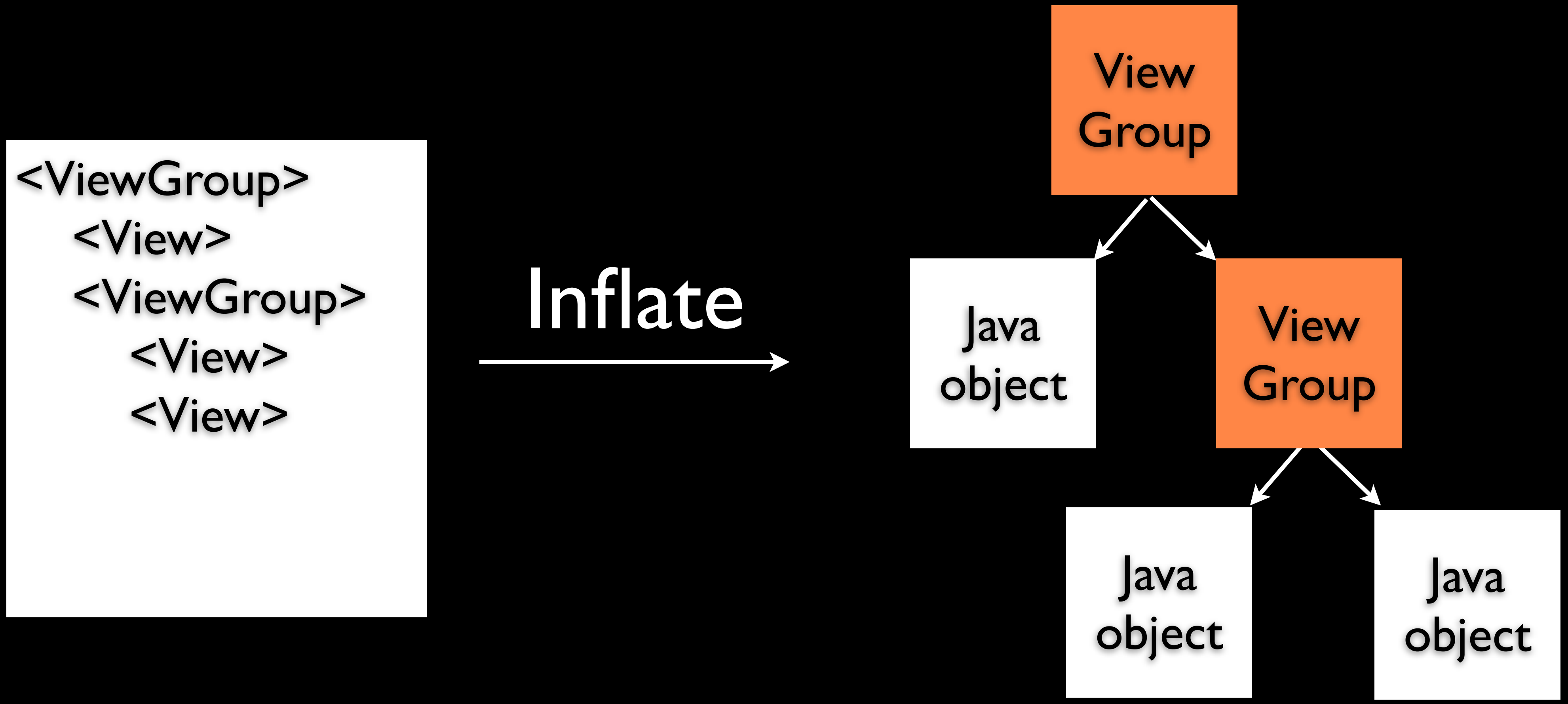
- Programmatic and Descriptive UI
  - Declare UI in XML description
  - Manipulate UI programmatically
- Separate UI presentation from UI behavior



JAYWAY

UI

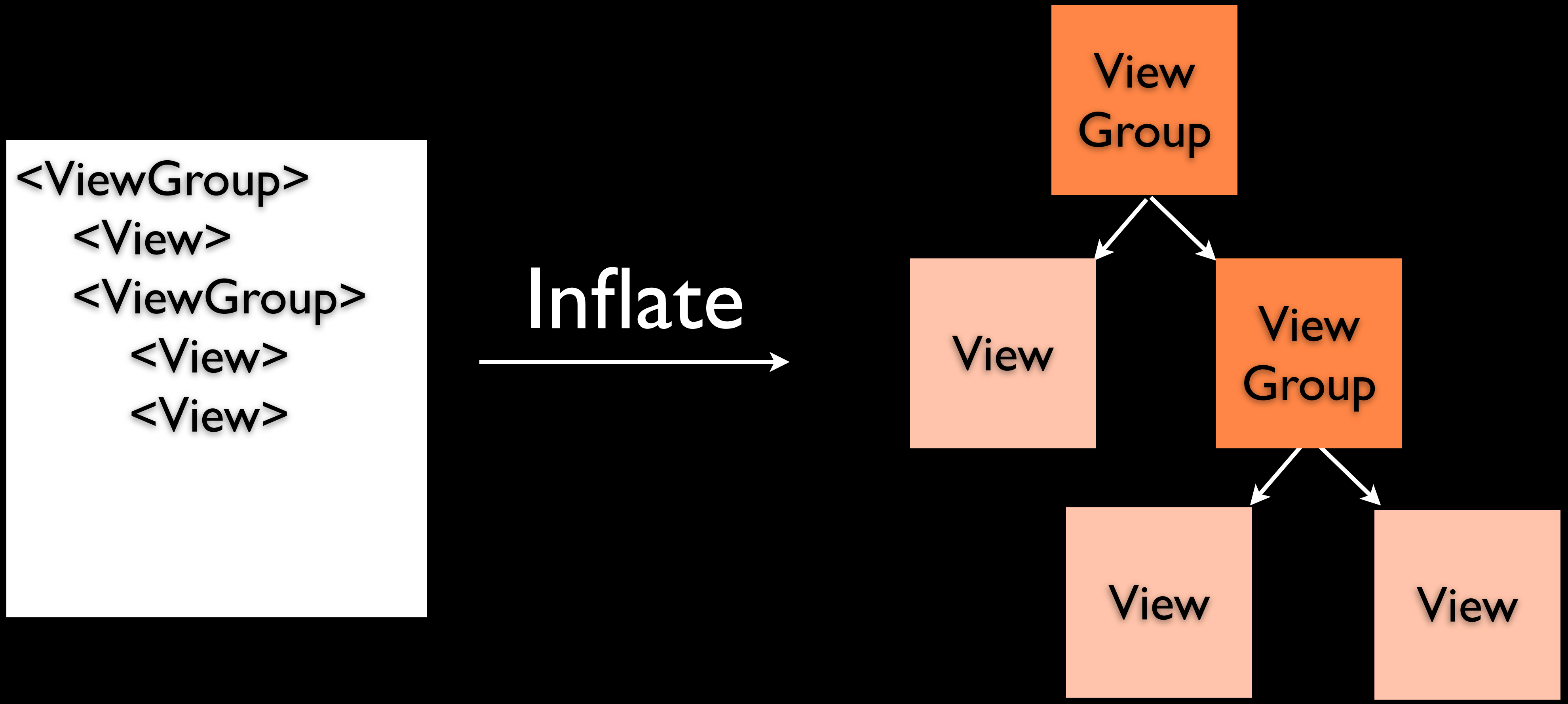
- Programmatic and Descriptive UI
  - Declare UI in XML description
  - Manipulate UI programmatically
- Separate UI presentation from UI behavior



JAYWAY

UI

- Programmatic and Descriptive UI
  - Declare UI in XML description
  - Manipulate UI programmatically
- Separate UI presentation from UI behavior

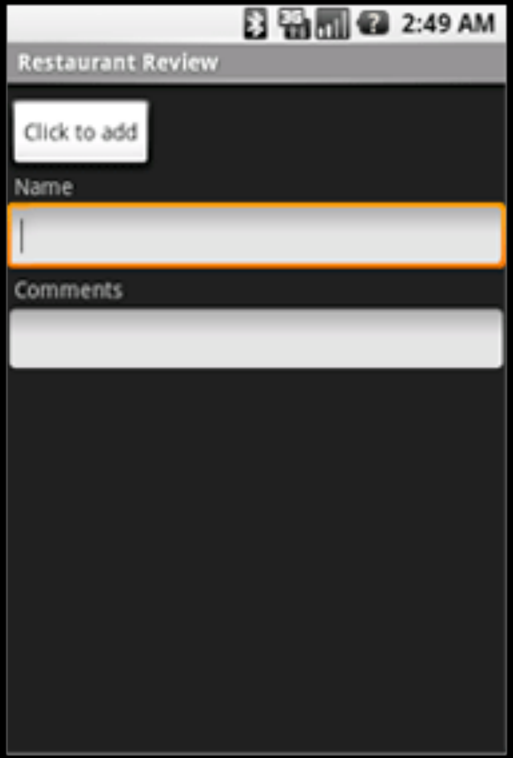


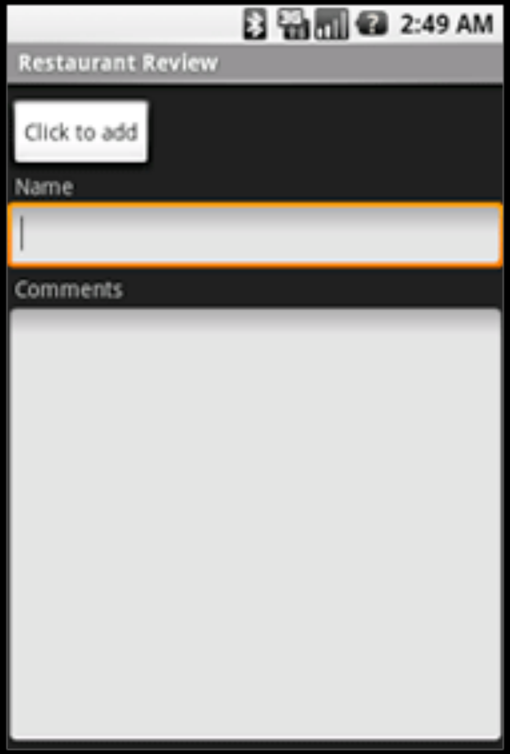
Layouts

- Implements ViewGroup
- Common layouts
  - LinearLayout
  - RelativeLayout
  - TableLayout

Layouts

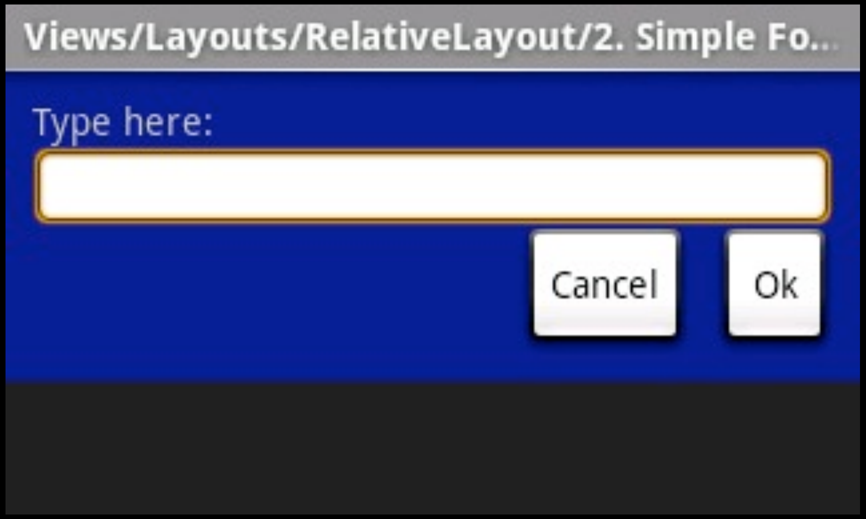
- Implements ViewGroup
- Common layouts
  - ★ LinearLayout
  - RelativeLayout
  - TableLayout





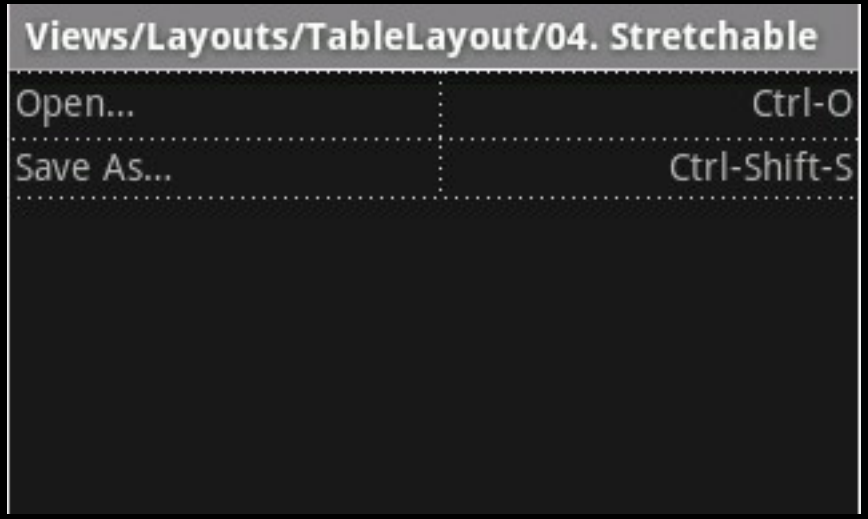
Layouts

- Implements ViewGroup
- Common layouts
  - LinearLayout
  - ★ RelativeLayout
  - TableLayout



Layouts

- Implements ViewGroup
- Common layouts
  - LinearLayout
  - RelativeLayout
  - ★ TableLayout



Views/Layouts/TableLayout/04. Stretchable

Open...	Ctrl-O
Save As...	Ctrl-Shift-S

Layouts

- Impl
- Com
  - L
  - R
  - ☆ Ta

yout/04. Stretchable

Ctrl-O

Ctrl-Shift-S

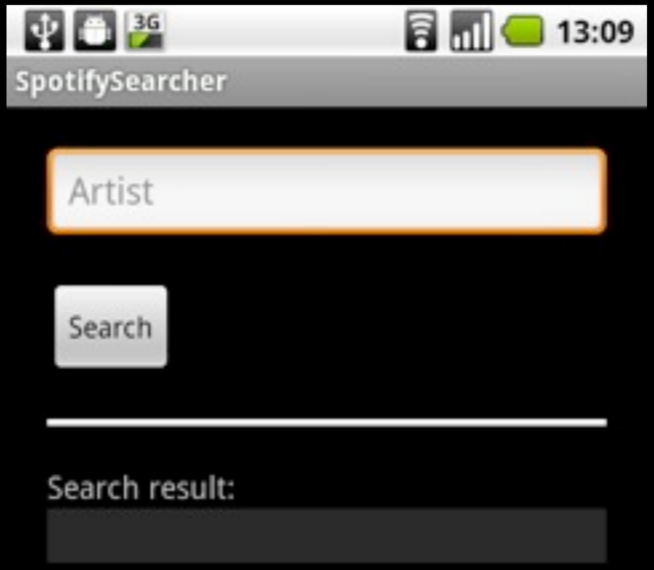
Consider Performance

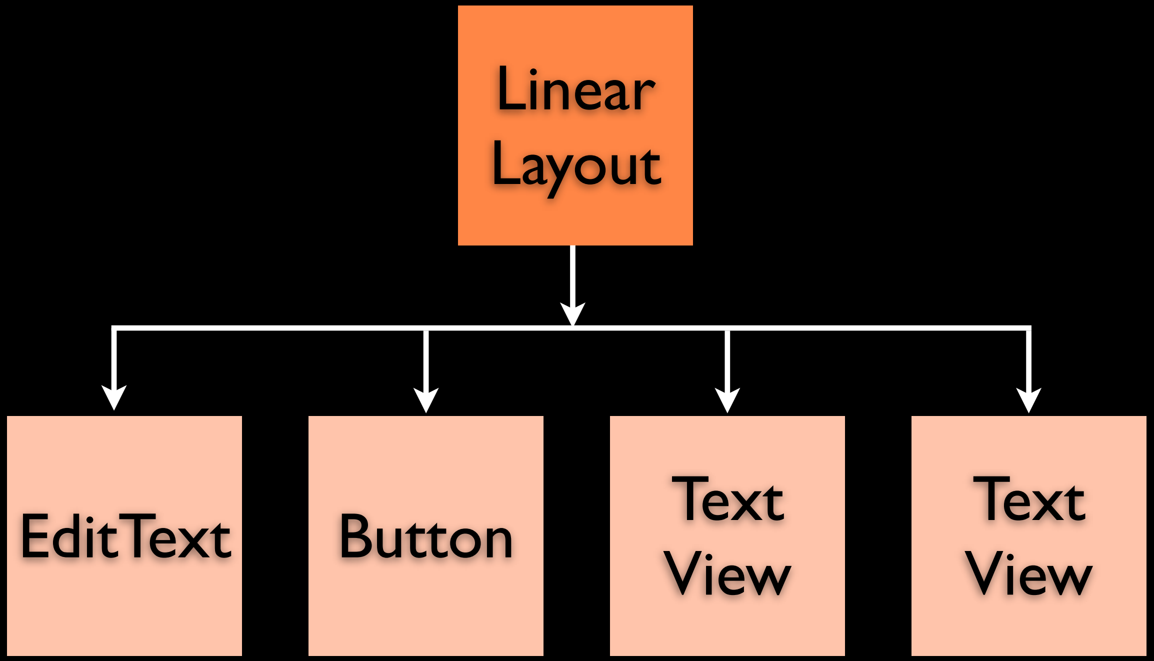
Inflating objects is expensive. Use as few elements as possible.

Use ‘layoutopt’

JAYWAY

Layout Example

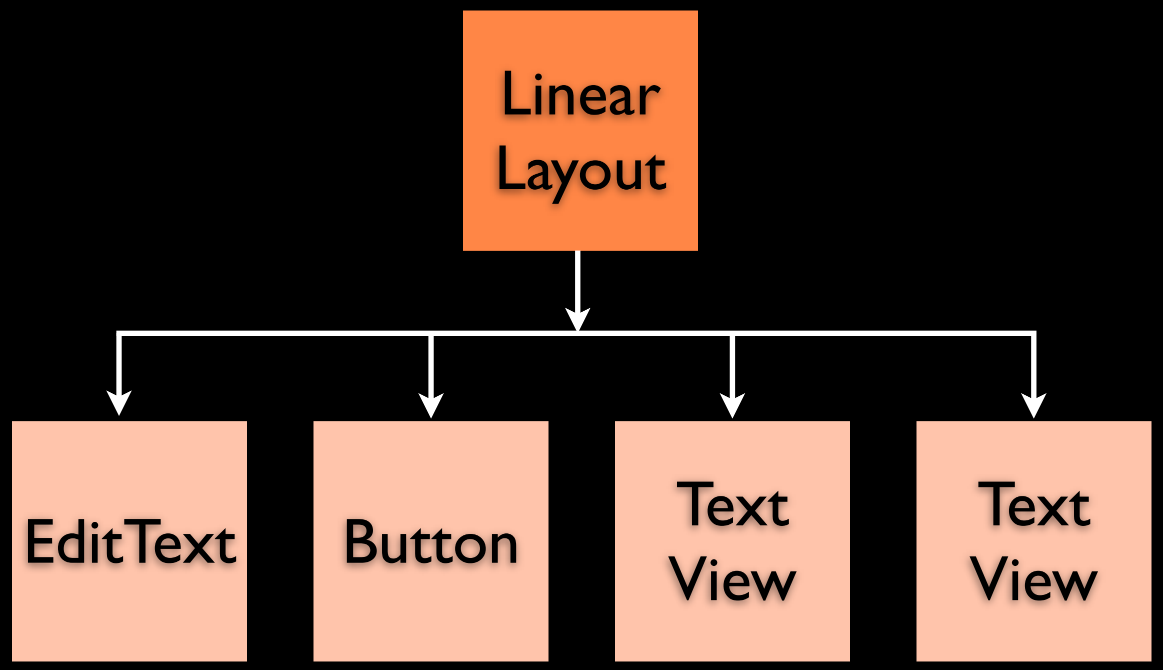




```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="@dimen/padding_default">
    <EditText
        android:id="@+id/edit_search_input"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="@string/artist" />
    <Button
        android:id="@+id/button_search"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="@dimen/margin_vertical_default"
        android:layout_marginTop="@dimen/margin_vertical_default"
        android:text="@string/search" />
    <include
        layout="@layout/divider_line" />
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="@dimen/margin_vertical_default"
        android:textSize="@dimen/text_size_default"
        android:text="@string/search_result"/>
    <TextView
        android:id="@+id/text_search_result"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="@color/dark_grey"
        android:textSize="@dimen/text_size_large"/>
</LinearLayout>
```

Drawing UI

- Top-down traversal
  - Measure
    - fill_parent, wrap_content, “fix value”
  - Layout
    - layout_*



JAYWAY

Use Resource Separation

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="@dimen/padding_default">
    <EditText
        android:id="@+id/edit_search_input"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="@string/artist" />
    <Button
        android:id="@+id/button_search"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="@dimen/margin_vertical_default"
        android:layout_marginTop="@dimen/margin_vertical_default"
        android:text="@string/search" />
    <include
        layout="@layout/divider_line" />
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="@dimen/margin_vertical_default"
        android:textSize="@dimen/text_size_default"
        android:text="@string/search_result"/>
    <TextView
        android:id="@+id/text_search_result"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="@color/dark_grey"
        android:textSize="@dimen/text_size_large"/>
</LinearLayout>
```

JAYWAY

Use Resource Separation

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="@dimen/padding_default">
    <EditText
        android:id="@+id/edit_search_input"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="@string/artist" />
    <Button
        android:id="@+id/button_search"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="@dimen/margin_vertical_default"
        android:layout_marginTop="@dimen/margin_vertical_default"
        android:text="@string/search" />
    <include
        layout="@layout/divider_line" />
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="@dimen/margin_vertical_default"
        android:textSize="@dimen/text_size_default"
        android:text="@string/search_result"/>
    <TextView
        android:id="@+id/text_search_result"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="@color/dark_grey"
        android:textSize="@dimen/text_size_large"/>
</LinearLayout>
```

colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
      <color name="white">#ffffff</color>
      <color name="dark_grey">#222222</color>
</resources>
```

JAYWAY

Use Resource Separation

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="@dimen/padding_default">
    <EditText
        android:id="@+id/edit_search_input"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="@string/artist" />
    <Button
        android:id="@+id/button_search"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="@dimen/margin_vertical_default"
        android:layout_marginTop="@dimen/margin_vertical_default"
        android:text="@string/search" />
    <include
        layout="@layout/divider_line" />
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="@dimen/margin_vertical_default"
        android:textSize="@dimen/text_size_default"
        android:text="@string/search_result"/>
    <TextView
        android:id="@+id/text_search_result"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="@color/dark_grey"
        android:textSize="@dimen/text_size_large"/>
</LinearLayout>
```

colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
      <color name="white">#ffffff</color>
      <color name="dark_grey">#222222</color>
</resources>
```

dimens.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
      <dimen name="padding_default">20dp</dimen>

      <dimen name="margin_vertical_default">20dp</dimen>

      <dimen name="text_size_default">15sp</dimen>
      <dimen name="text_size_large">20sp</dimen>
</resources>
```

JAYWAY

Use Resource Separation

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="@dimen/padding_default">
    <EditText
        android:id="@+id/edit_search_input"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="@string/artist" />
    <Button
        android:id="@+id/button_search"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="@dimen/margin_vertical_default"
        android:layout_marginTop="@dimen/margin_vertical_default"
        android:text="@string/search" />
    <include
        layout="@layout/divider_line" />
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="@dimen/margin_vertical_default"
        android:textSize="@dimen/text_size_default"
        android:text="@string/search_result"/>
    <TextView
        android:id="@+id/text_search_result"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="@color/dark_grey"
        android:textSize="@dimen/text_size_large"/>
</LinearLayout>
```

colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
     <color name="white">#ffffff</color>
     <color name="dark_grey">#222222</color>
</resources>
```

dimens.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
     <dimen name="padding_default">20dp</dimen>

     <dimen name="margin_vertical_default">20dp</dimen>

     <dimen name="text_size_default">15sp</dimen>
     <dimen name="text_size_large">20sp</dimen>
</resources>
```

strings.xml

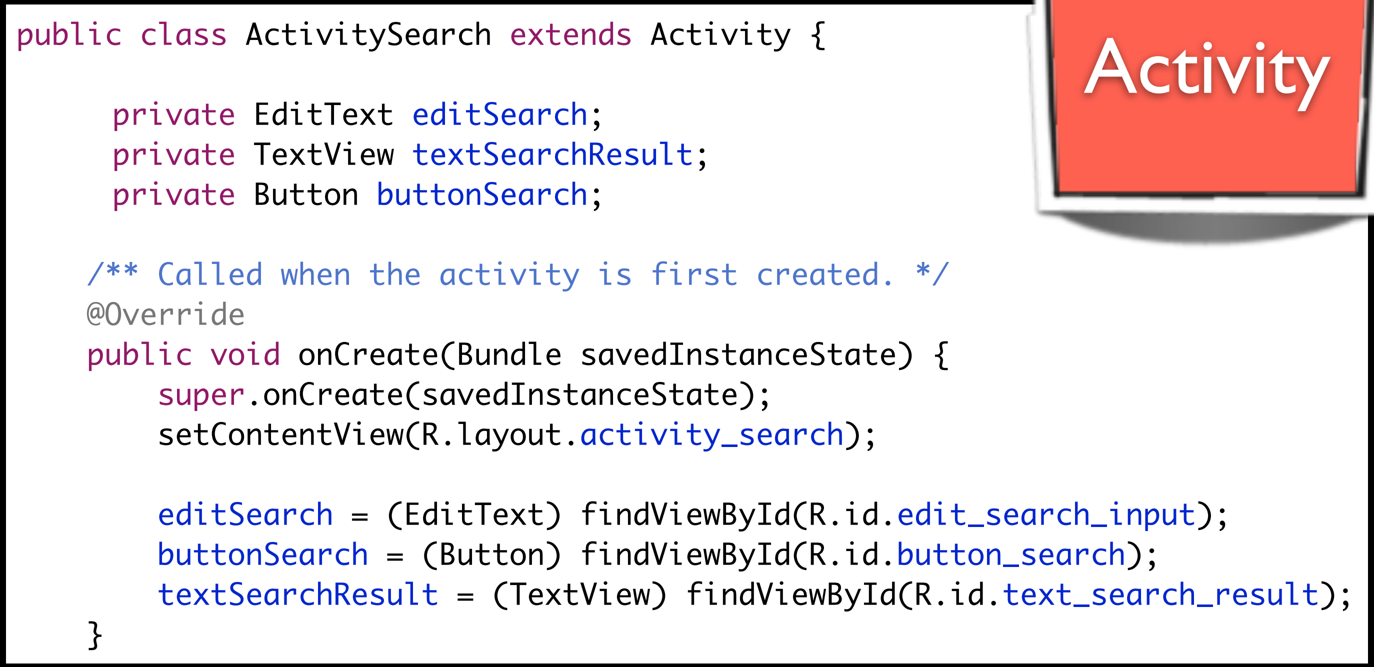
```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">SpotifySearcher</string>
    <string name="artist">Artist</string>
    <string name="search">Search</string>
    <string name="search_result">Search result:</string>
</resources>
```

JAYWAY

Activity

- Application component
- Subclassed by application
- User interface
  - Typically one full size screen
  - Inflates UI layouts
  - Key events
- Activity initialization
  - onCreate()

Accessing Resources



```
public class ActivitySearch extends Activity {

     private EditText editSearch;
     private TextView textSearchResult;
     private Button buttonSearch;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_search);

        editSearch = (EditText) findViewById(R.id.edit_search_input);
        buttonSearch = (Button) findViewById(R.id.button_search);
        textSearchResult = (TextView) findViewById(R.id.text_search_result);
    }
```

1. Inflate layout
2. Access by Id

Accessing Resources

Activity

```
public class ActivitySearch extends Activity {

     private EditText editSearch;
     private TextView textSearchResult;
     private Button buttonSearch;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_search);

        editSearch = (EditText) findViewById(R.id.edit_search_input);
        buttonSearch = (Button) findViewById(R.id.button_search);
        textSearchResult = (TextView) findViewById(R.id.text_search_result);
    }
```

R.java

```
public final class R {
    public static final class attr {
    }
    public static final class color {
        public static final int dark_grey=0x7f040001;
        public static final int white=0x7f040000;
    }
    public static final class dimen {
        public static final int margin_vertical_default=0x7f050001;
        public static final int padding_default=0x7f050000;
        public static final int text_size_default=0x7f050002;
        public static final int text_size_large=0x7f050003;
    }
    public static final class drawable {
        public static final int icon=0x7f020000;
    }
    public static final class id {
        public static final int button_search=0x7f070001;
        public static final int edit_search_input=0x7f070000;
        public static final int text_search_result=0x7f070002;
    }
    public static final class layout {
        public static final int activity_search=0x7f030000;
        public static final int divider_line=0x7f030001;
    }
    public static final class string {
        public static final int app_name=0x7f060000;
        public static final int artist=0x7f060001;
        public static final int search=0x7f060002;
        public static final int search_result=0x7f060003;
    }
}
```

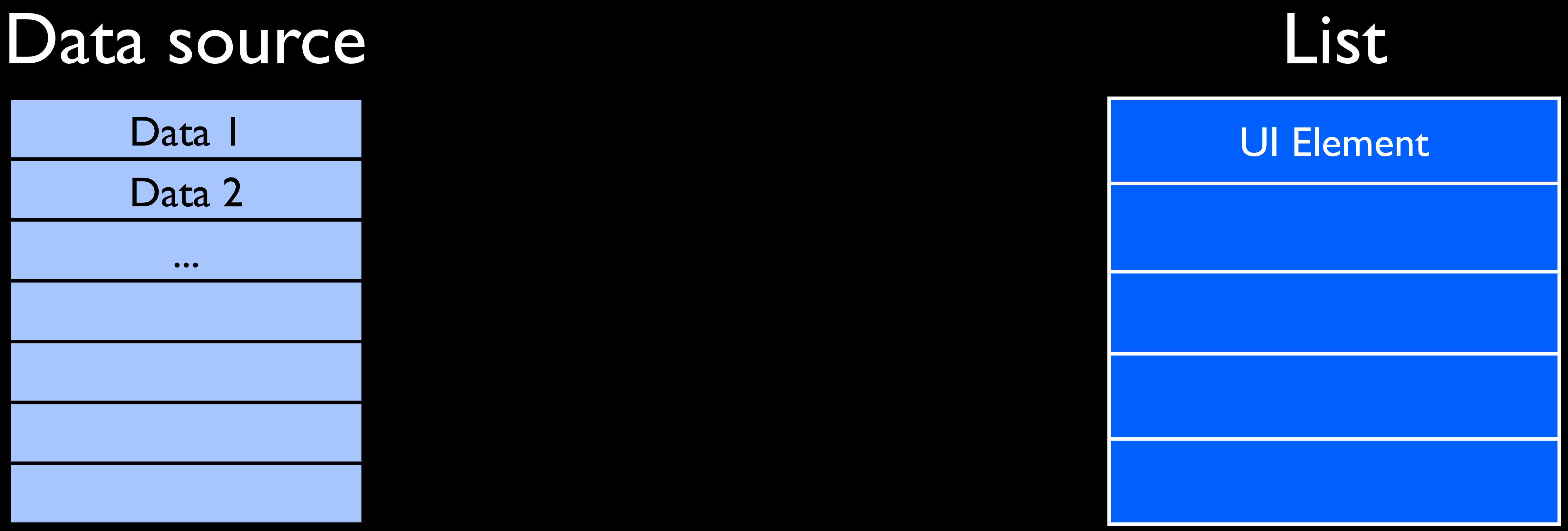
1. Inflate layout
2. Access by Id

JAYWAY

Lists

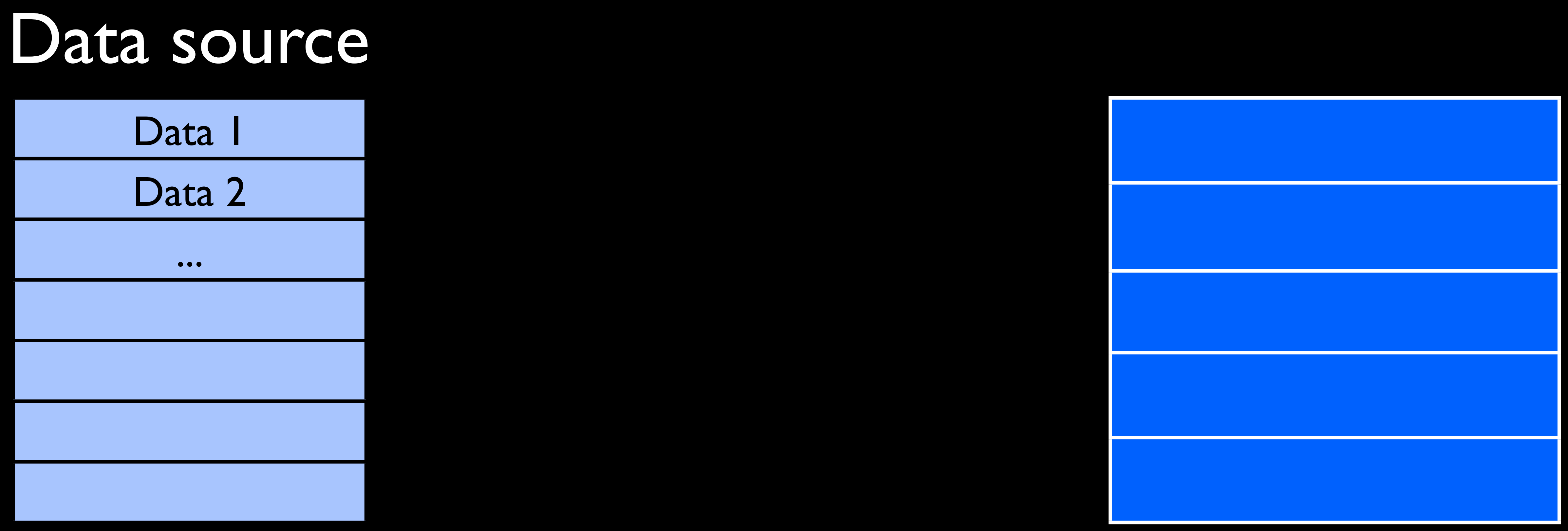
- Commonly used
- Needs to be efficient
  - Not consume too much memory
  - Fast scrolling
- Android-way of handling lists efficiently

The problem



Problem	Requirement
Arbitrary data source	Abstract mechanism
Arbitrary data type	Look and feel of each element
Dynamic data	Update list after insert, edit, remove
Lots of data requires a lot of memory	Effective caching

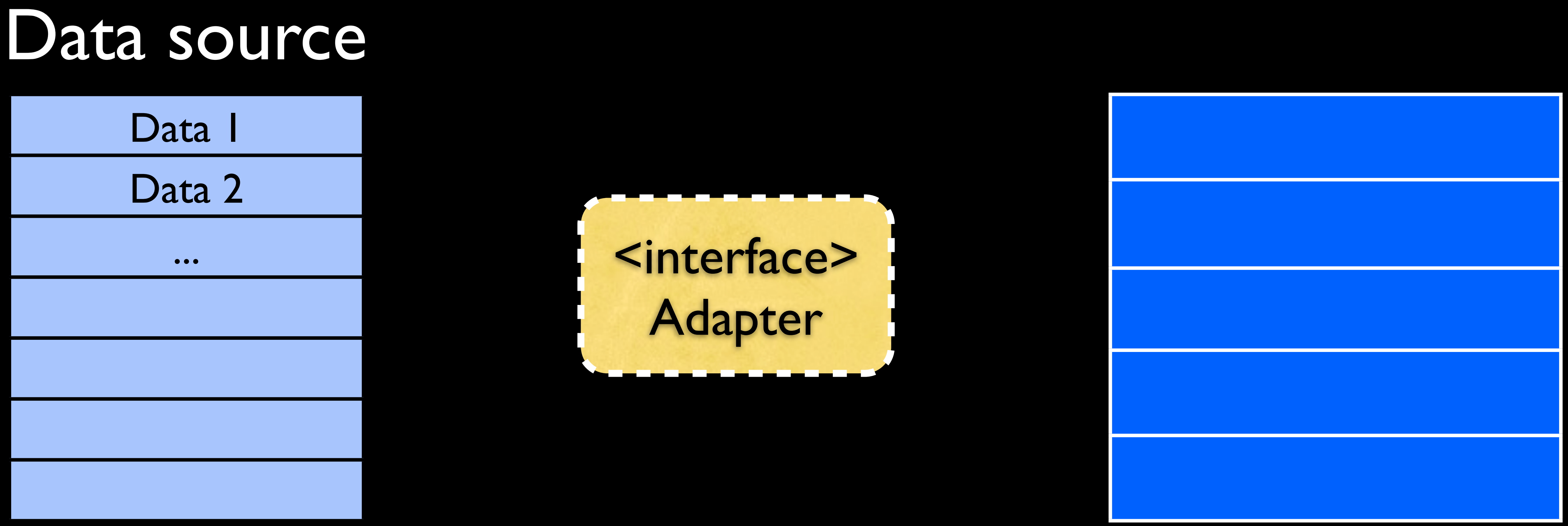
API



Problem	Requirement
Arbitrary data source	Abstract mechanism
Arbitrary data type	Look and feel of each element
Dynamic data	Update list after insert, edit, remove
Lots of data requires a lot of memory	Effective caching

JAYWAY

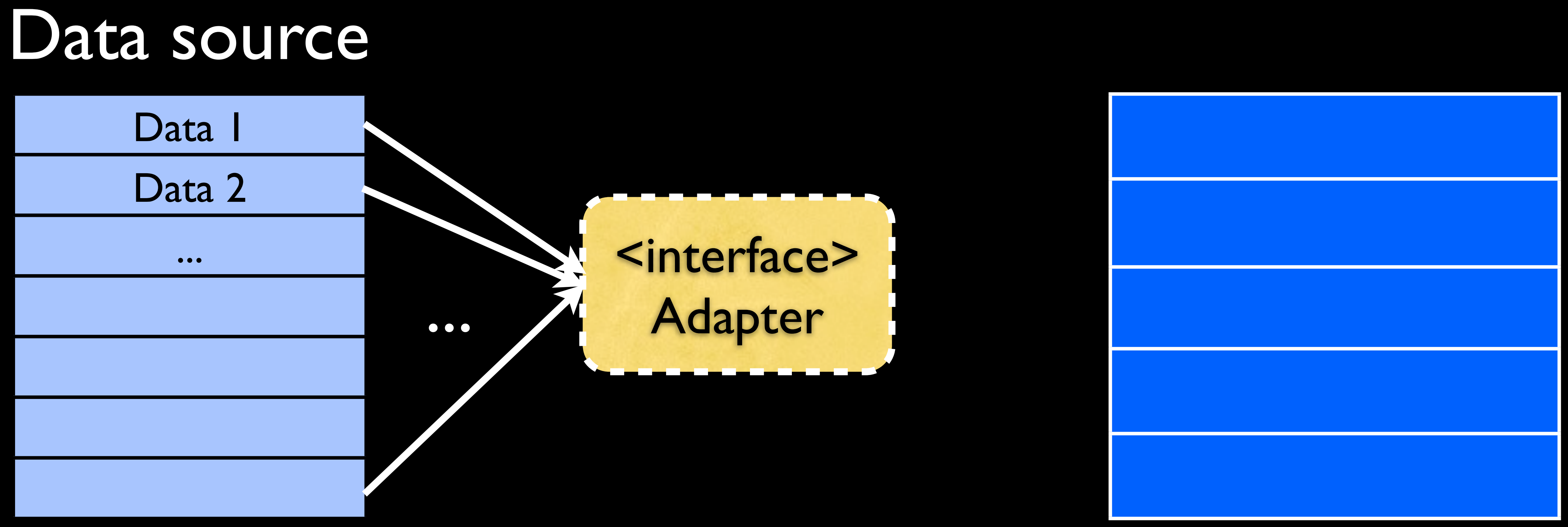
API



Problem	Requirement
Arbitrary data source	Abstract mechanism
Arbitrary data type	Look and feel of each element
Dynamic data	Update list after insert, edit, remove
Lots of data requires a lot of memory	Effective caching

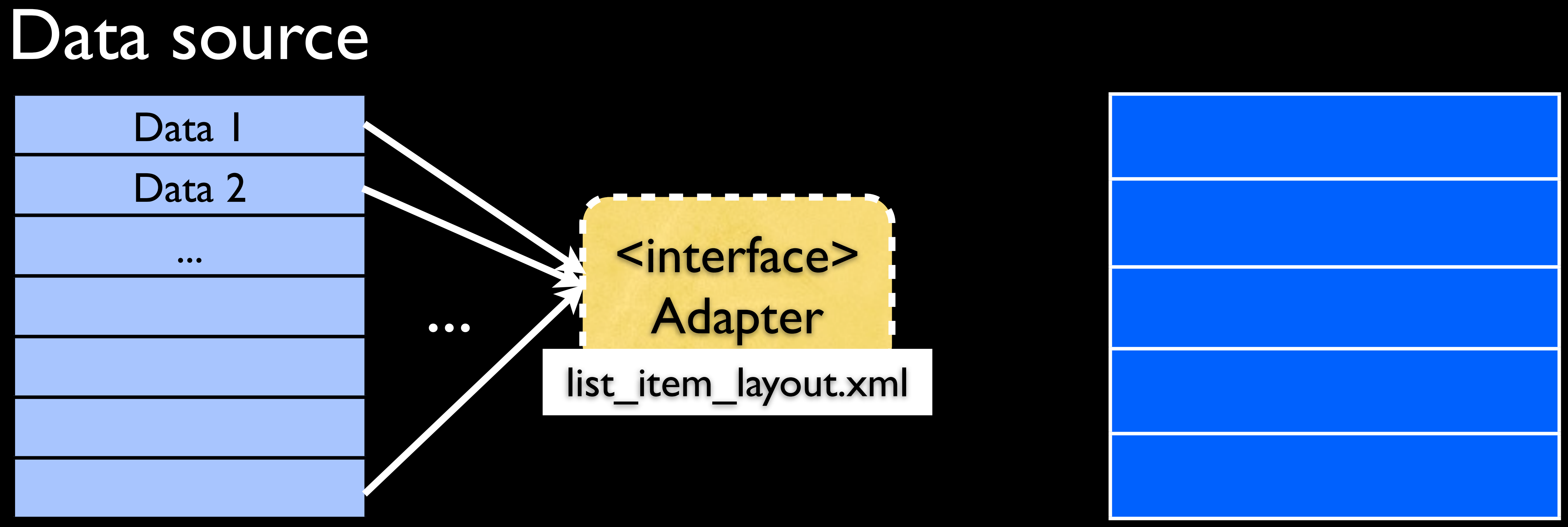
JAYWAY

API



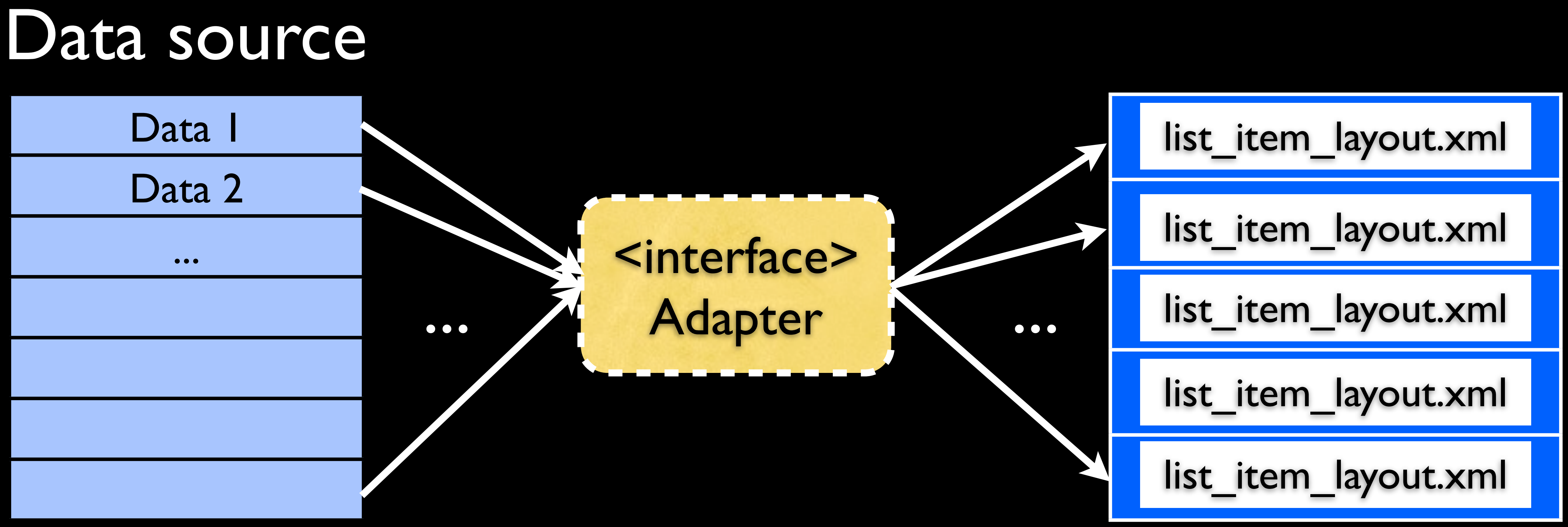
Problem	Requirement
Arbitrary data source	Abstract mechanism
Arbitrary data type	Look and feel of each element
Dynamic data	Update list after insert, edit, remove
Lots of data requires a lot of memory	Effective caching

API



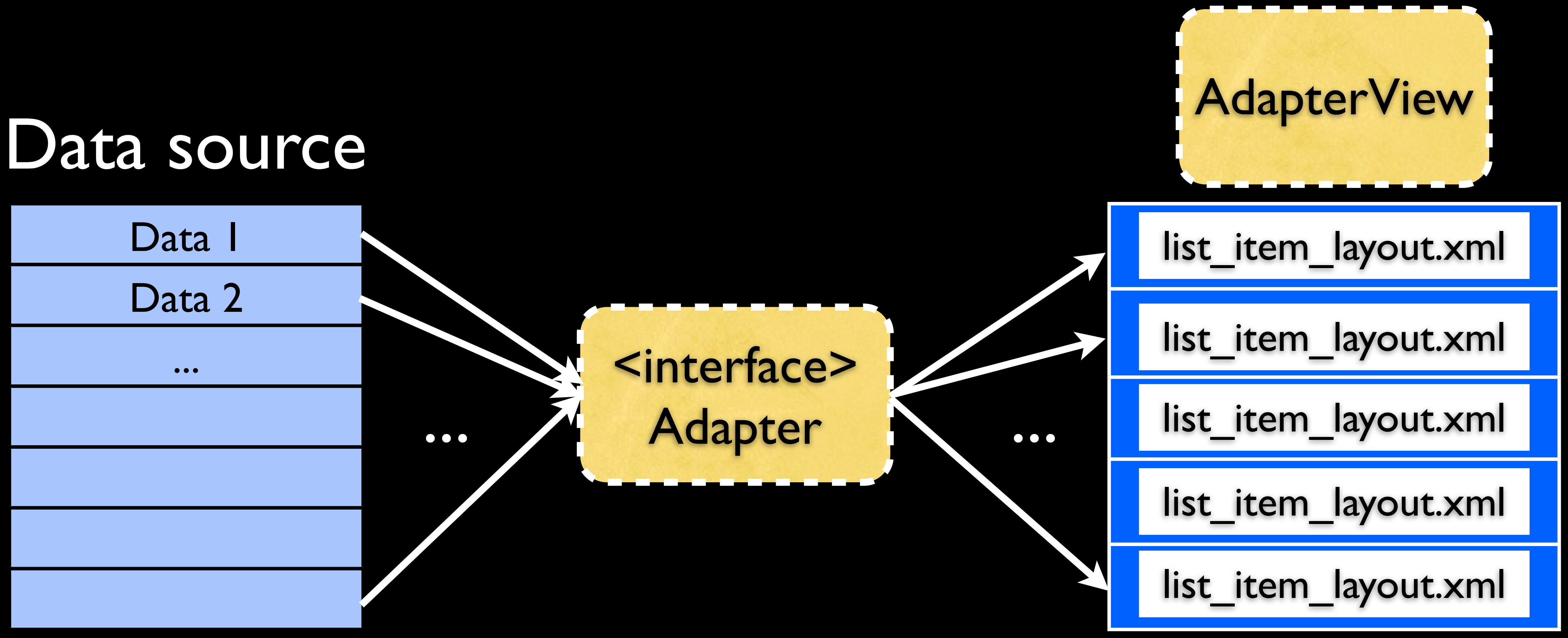
Problem	Requirement
Arbitrary data source	Abstract mechanism
Arbitrary data type	Look and feel of each element
Dynamic data	Update list after insert, edit, remove
Lots of data requires a lot of memory	Effective caching

API



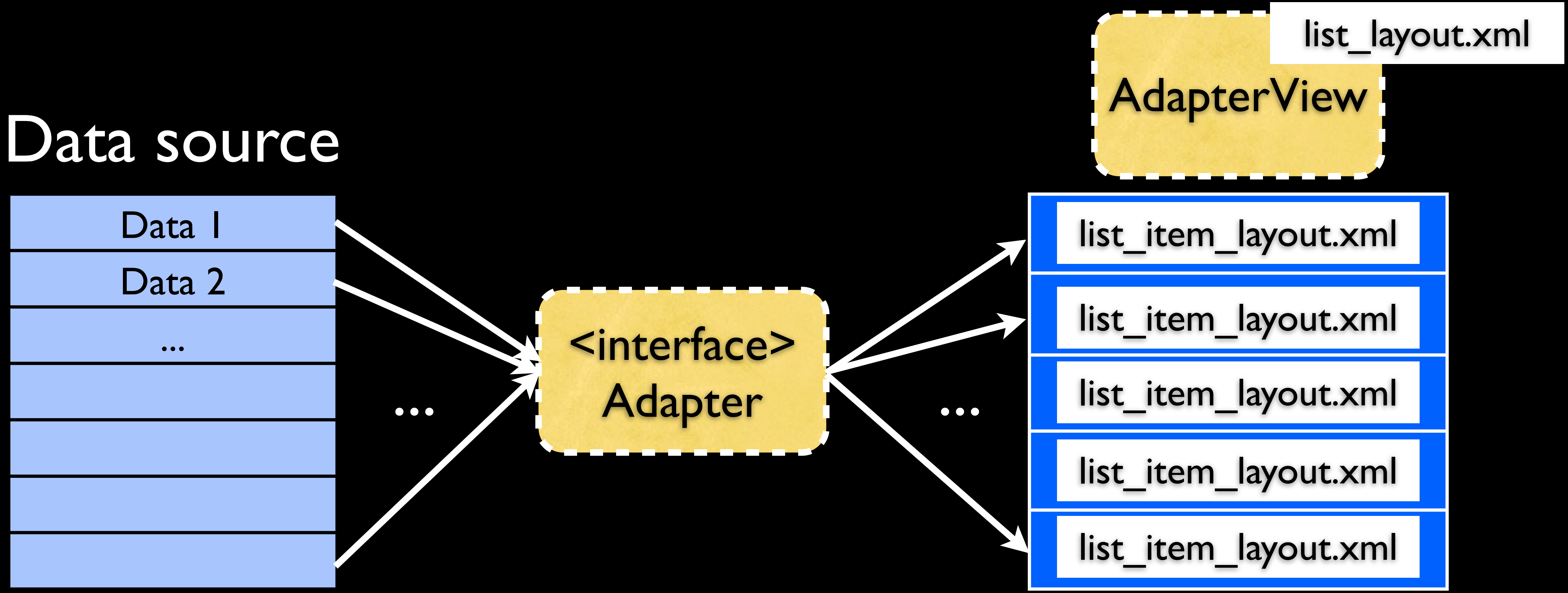
Problem	Requirement
Arbitrary data source	Abstract mechanism
Arbitrary data type	Look and feel of each element
Dynamic data	Update list after insert, edit, remove
Lots of data requires a lot of memory	Effective caching

API



Problem	Requirement
Arbitrary data source	Abstract mechanism
Arbitrary data type	Look and feel of each element
Dynamic data	Update list after insert, edit, remove
Lots of data requires a lot of memory	Effective caching

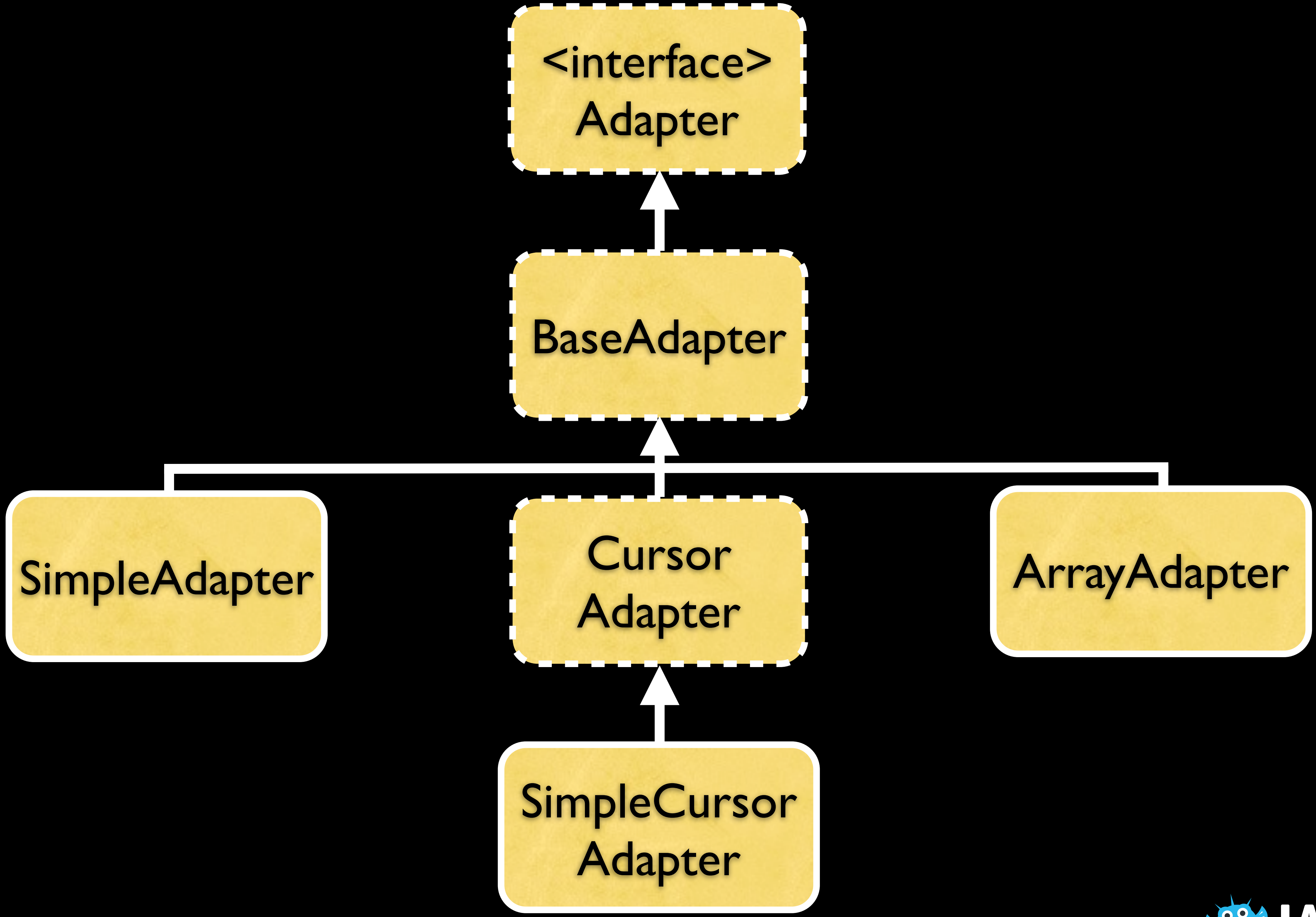
API



Problem	Requirement
Arbitrary data source	Abstract mechanism
Arbitrary data type	Look and feel of each element
Dynamic data	Update list after insert, edit, remove
Lots of data requires a lot of memory	Effective caching

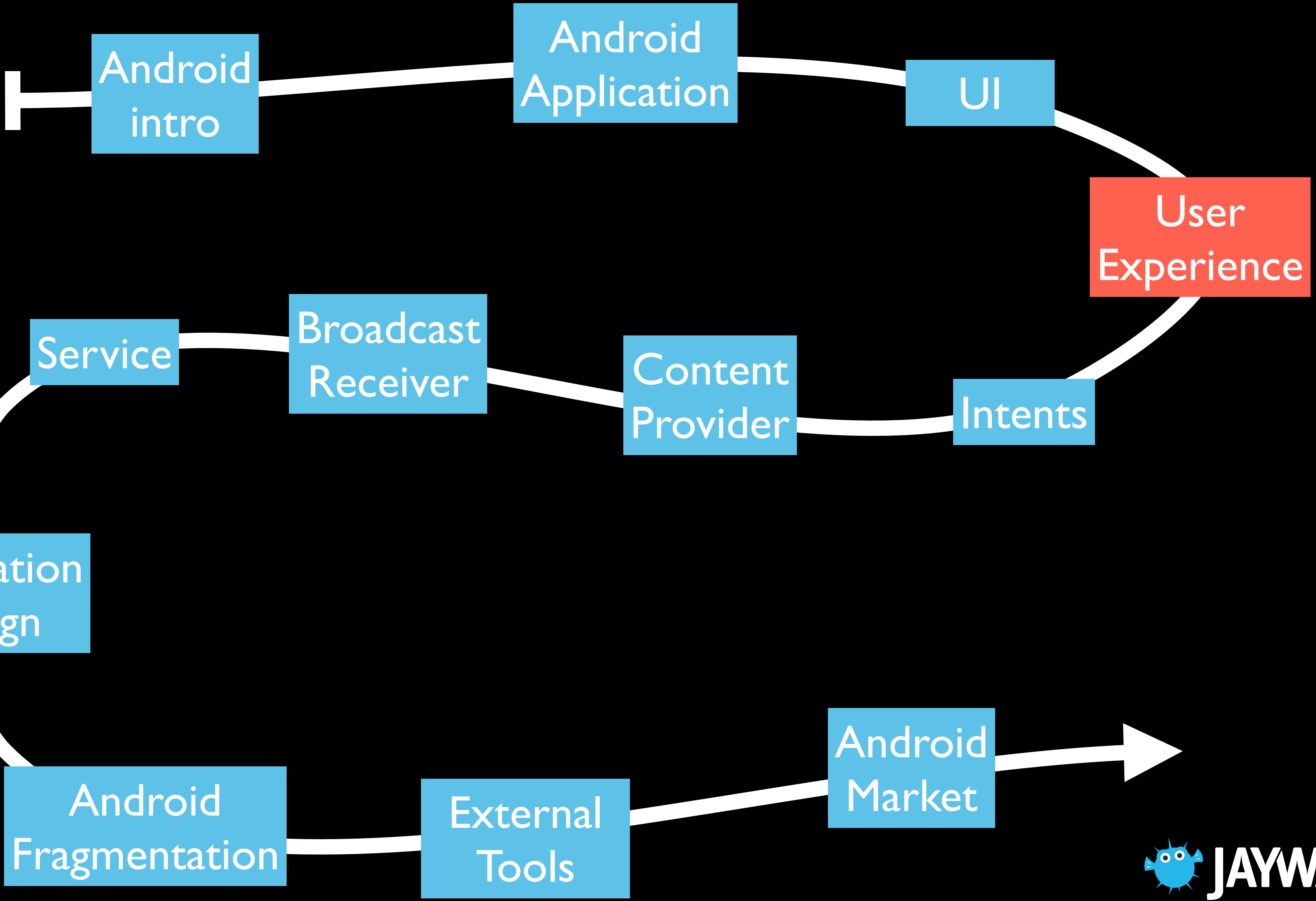
JAYWAY

Adapter



<interface>
Adapter
BaseAdapter
SimpleAdapter
Cursor
Adapter
ArrayAdapter
SimpleCursor
Adapter

Agenda



JAYWAY

User Experience

- Critical for application success
- UX
  - UI design
    - Employ a good designer
  - Deliver expected behavior
    - Master the Activity lifecycle
  - Responsive
    - Master the thread handling

JAYWAY

Lifecycle

States

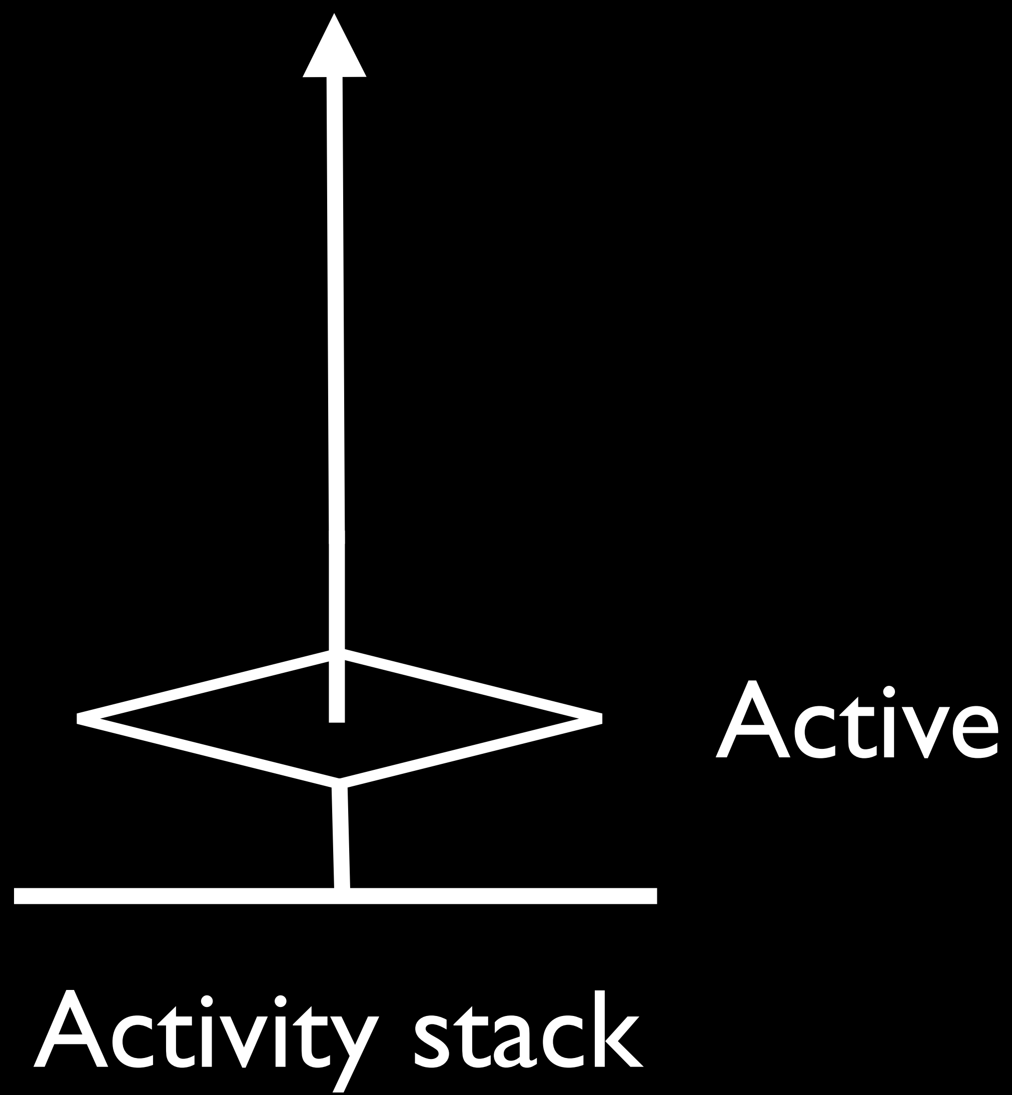
Active	Foreground
Paused	Partly obscured
Stopped	Completely obscured

JAYWAY

Lifecycle

States

Active	Foreground
Paused	Partly obscured
Stopped	Completely obscured

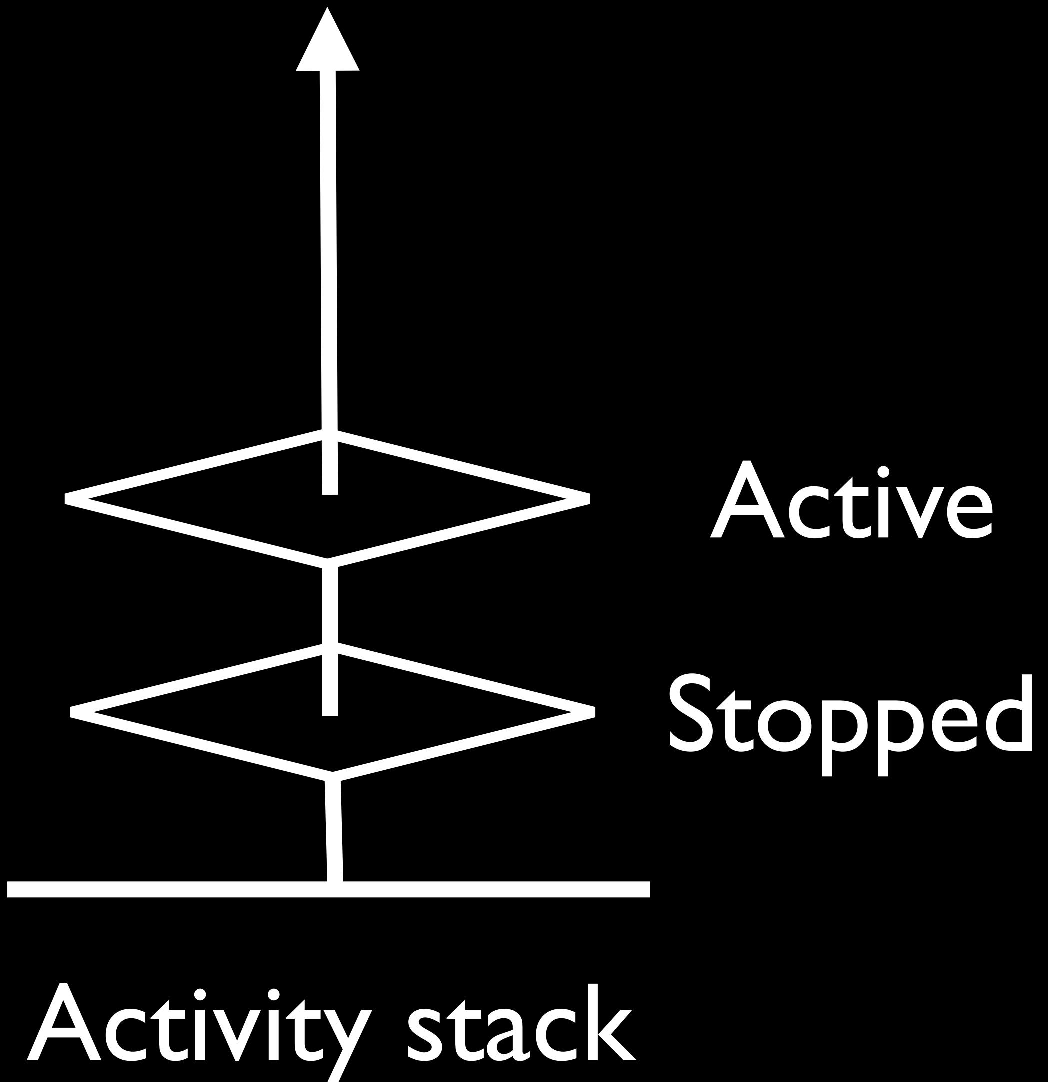


JAYWAY

Lifecycle

States

Active	Foreground
Paused	Partly obscured
Stopped	Completely obscured

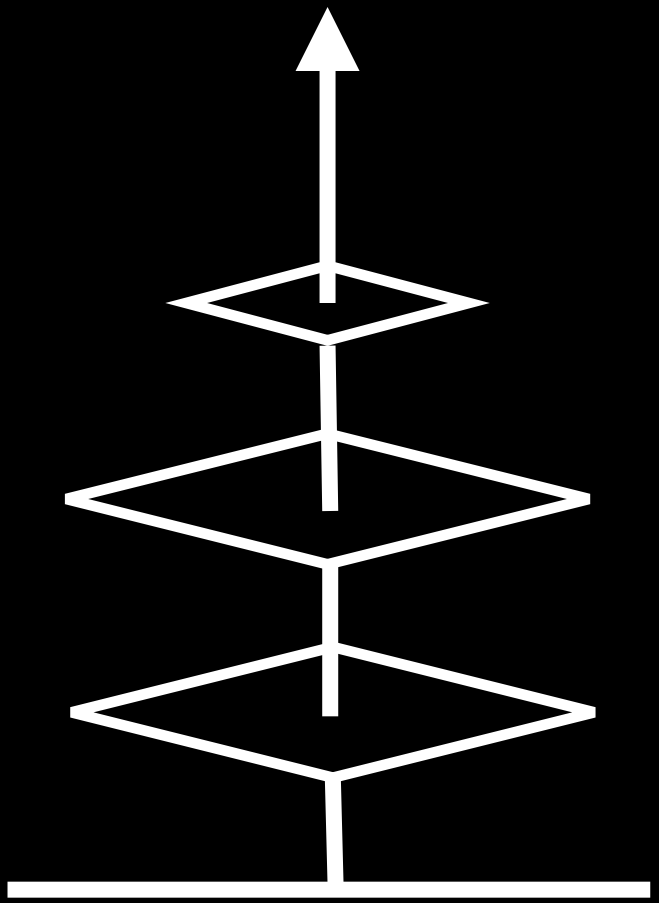


JAYWAY

Lifecycle

States

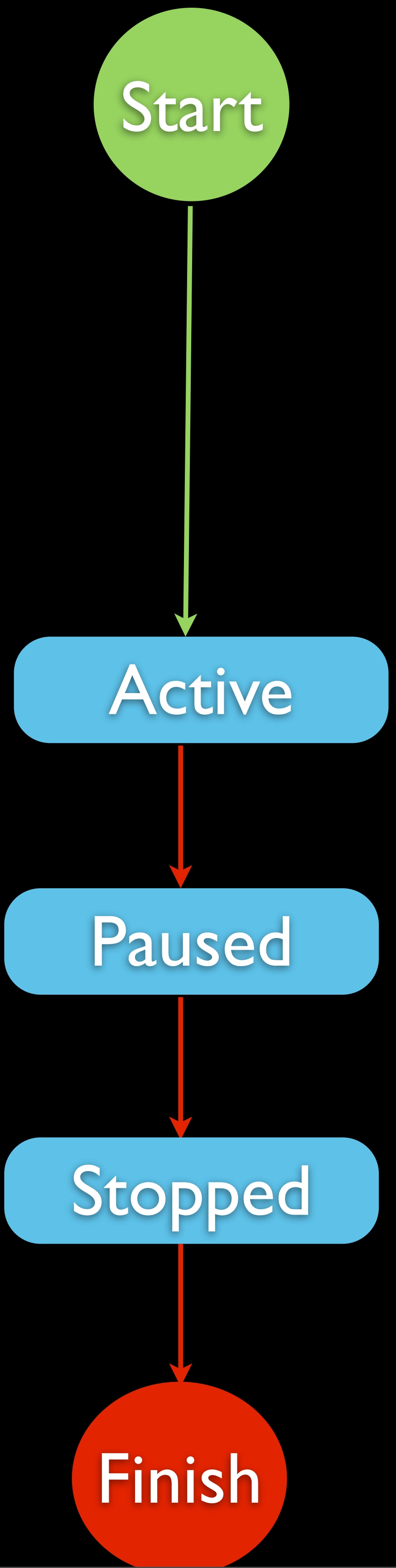
Active	Foreground
Paused	Partly obscured
Stopped	Completely obscured



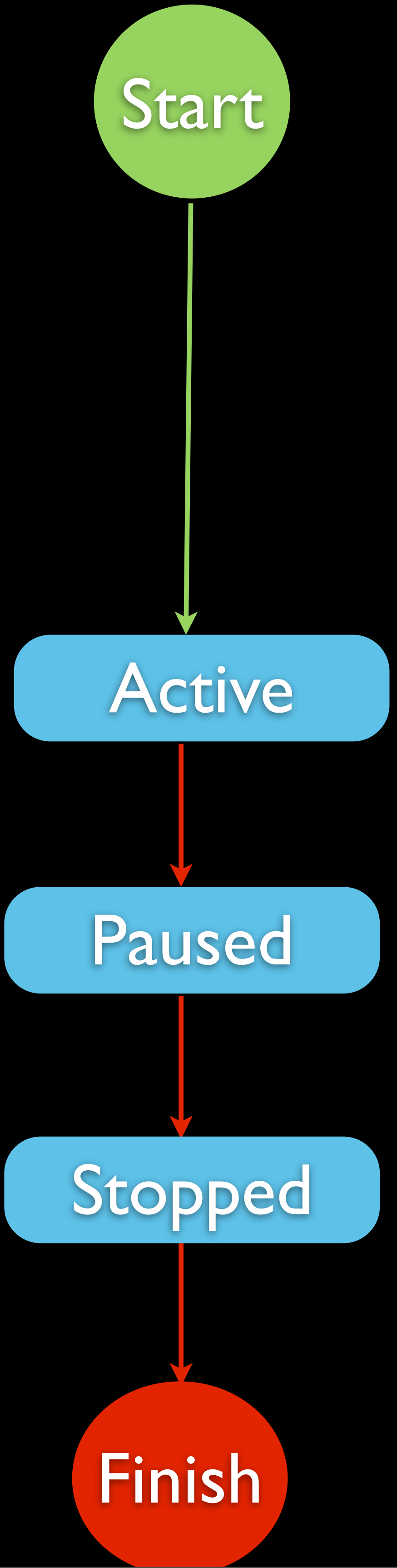
Activity stack

JAYWAY

Lifecycle

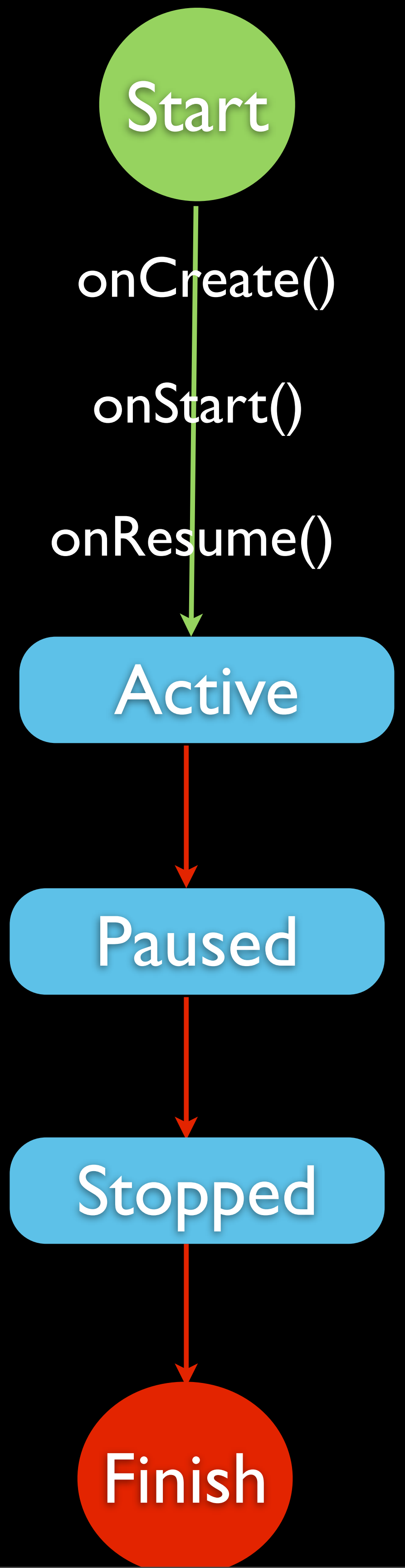


Lifecycle



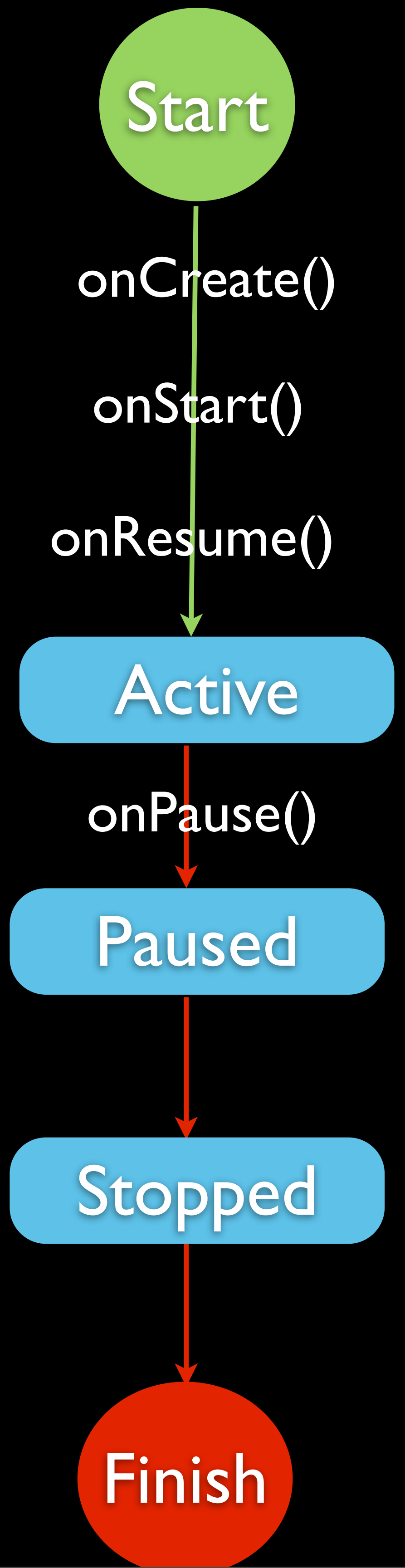
- System callbacks
- Override in subclass

Lifecycle



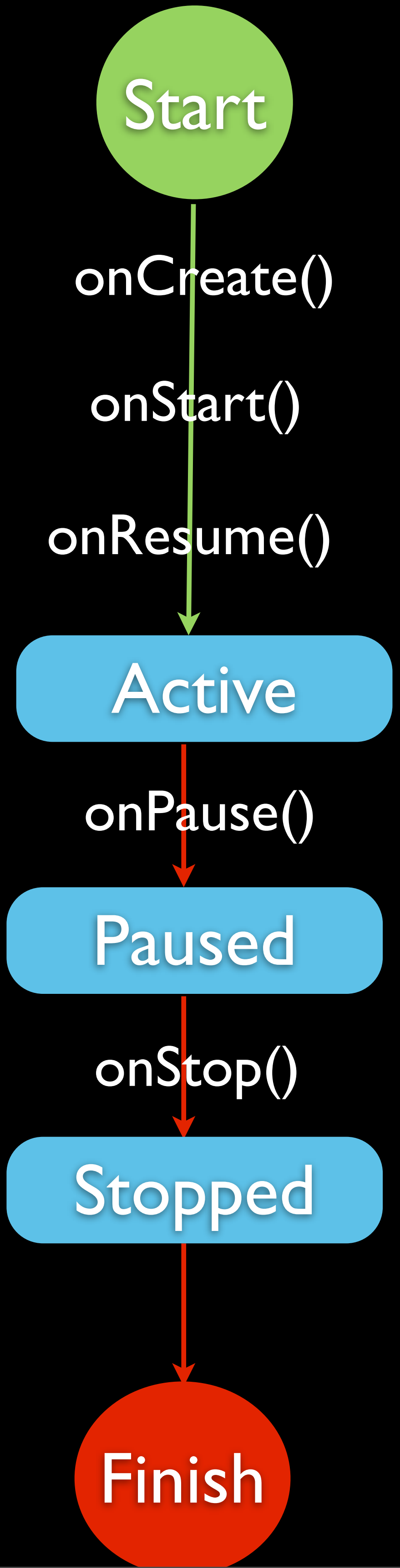
- System callbacks
- Override in subclass

Lifecycle



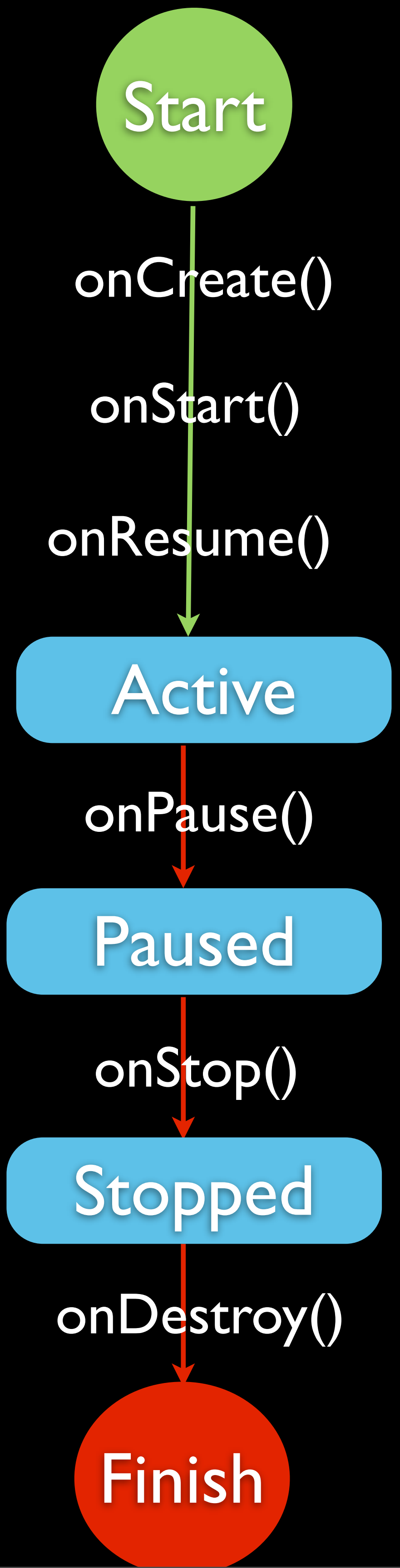
- System callbacks
- Override in subclass

Lifecycle



- System callbacks
- Override in subclass

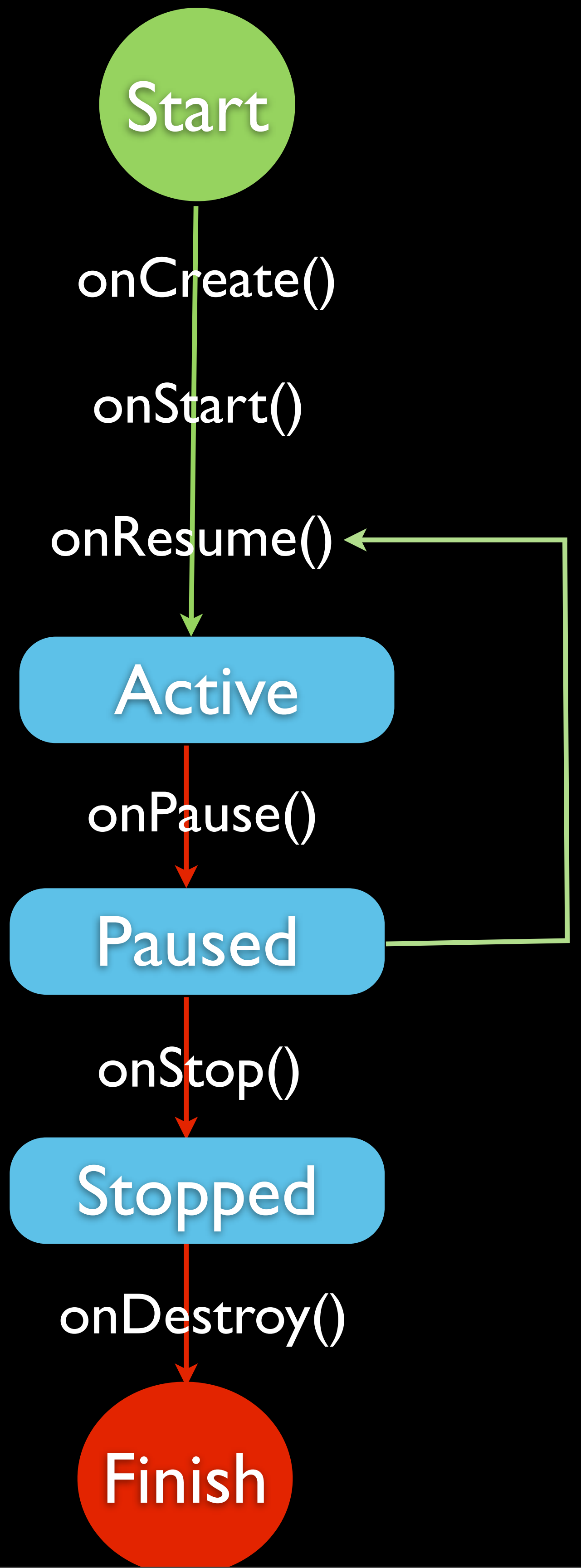
Lifecycle



- System callbacks
- Override in subclass

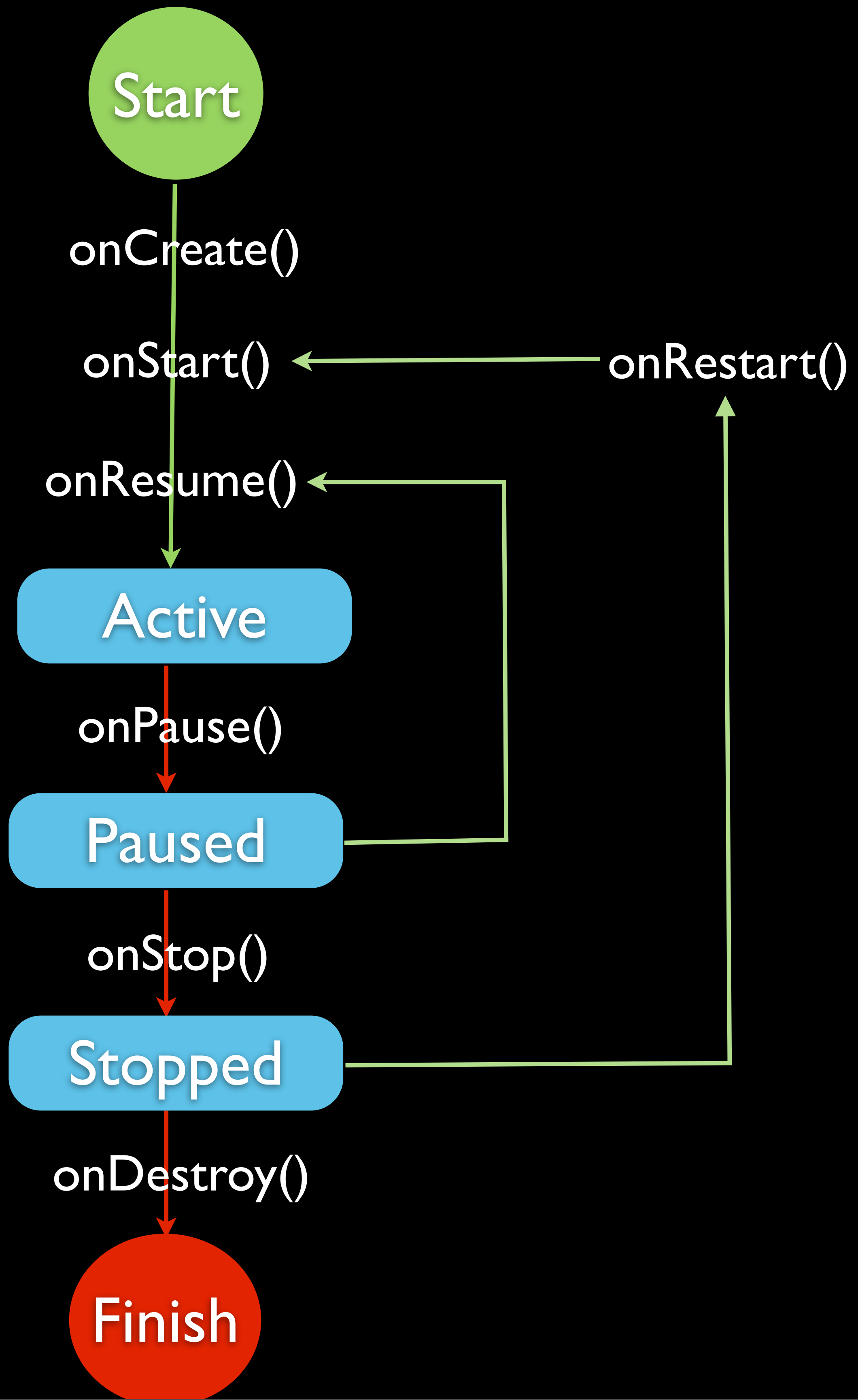
JAYWAY

Lifecycle



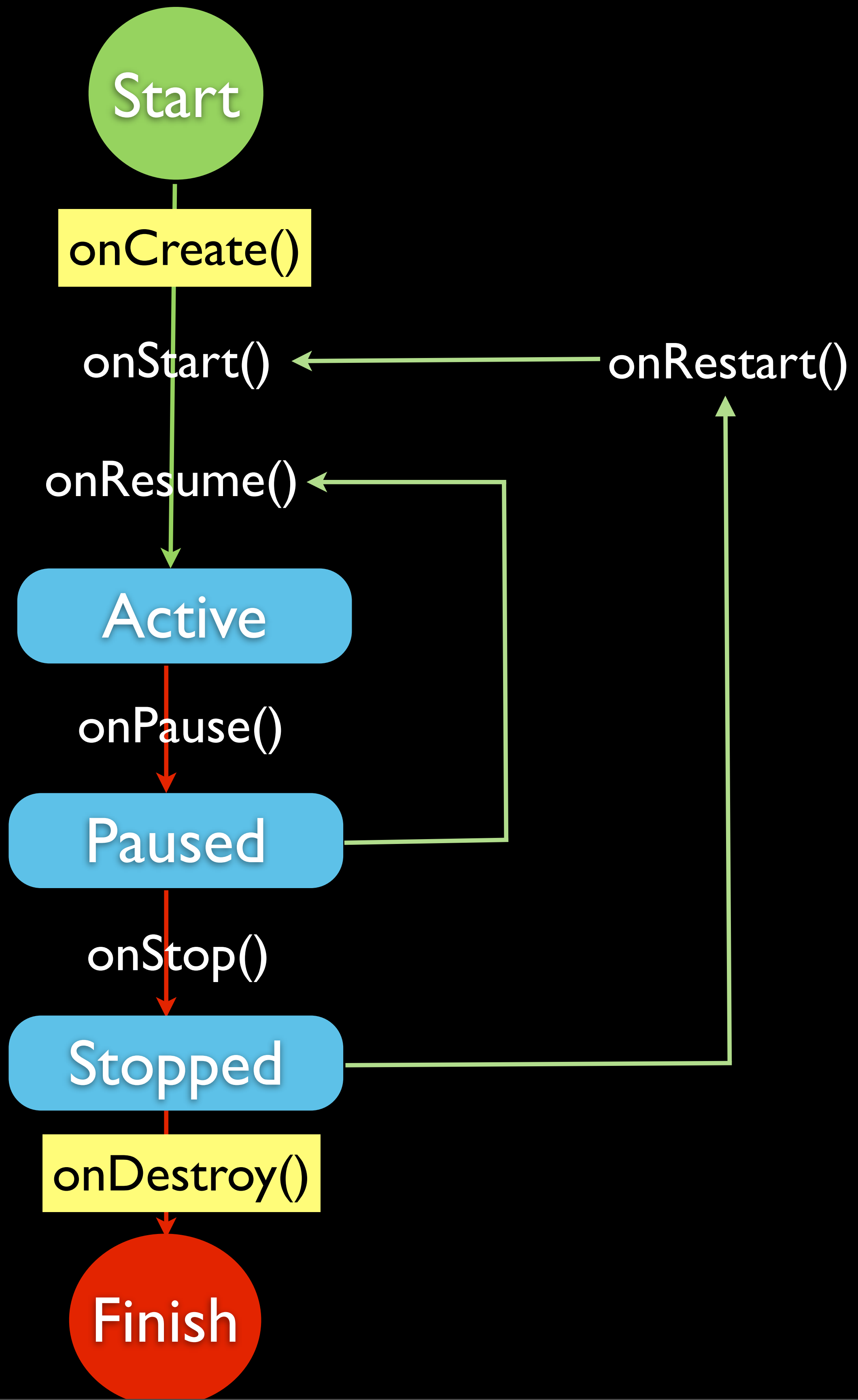
- System callbacks
- Override in subclass

Lifecycle



- System callbacks
- Override in subclass

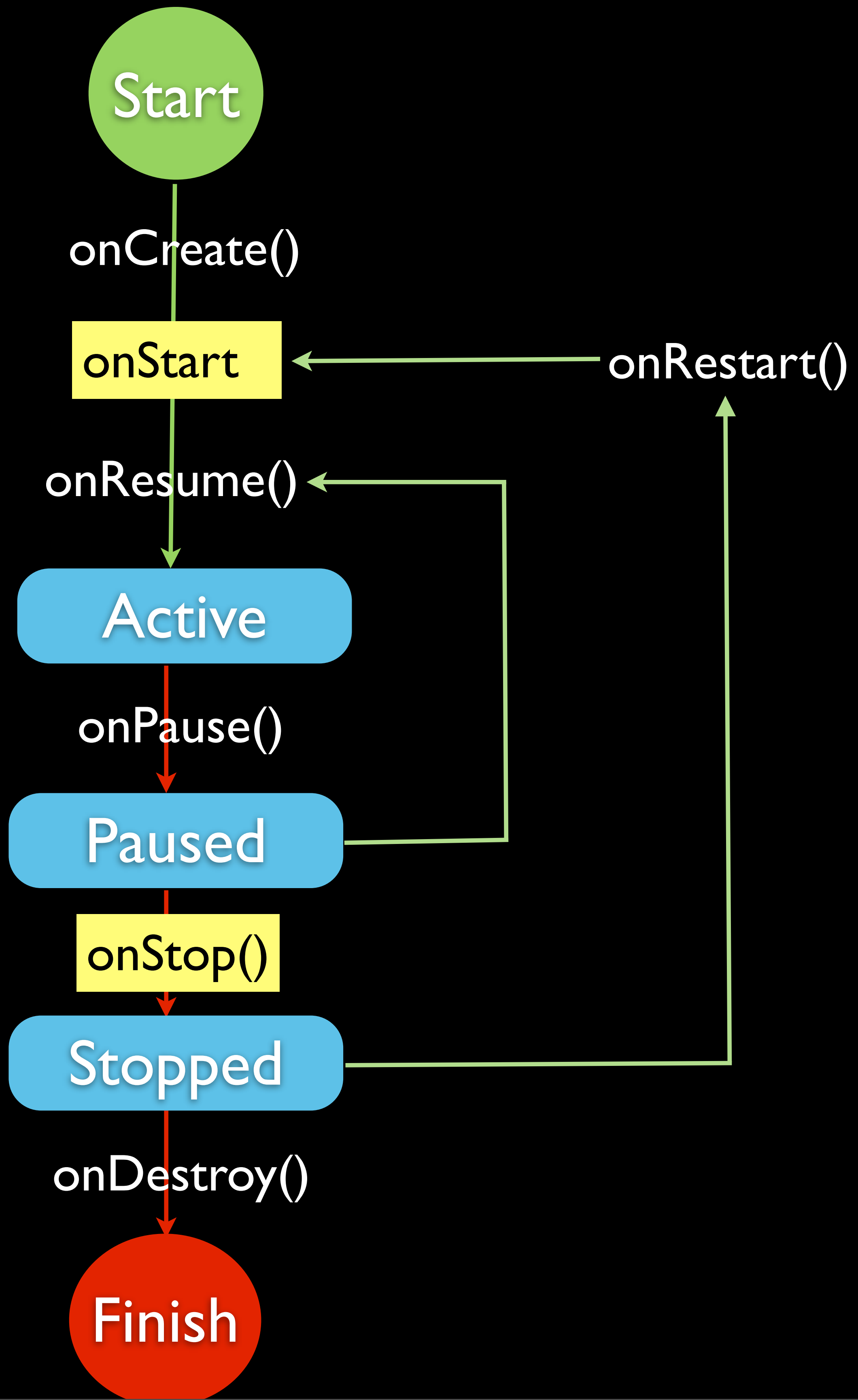
Lifecycle



Entire lifetime

- Long lived objects
- Create UI

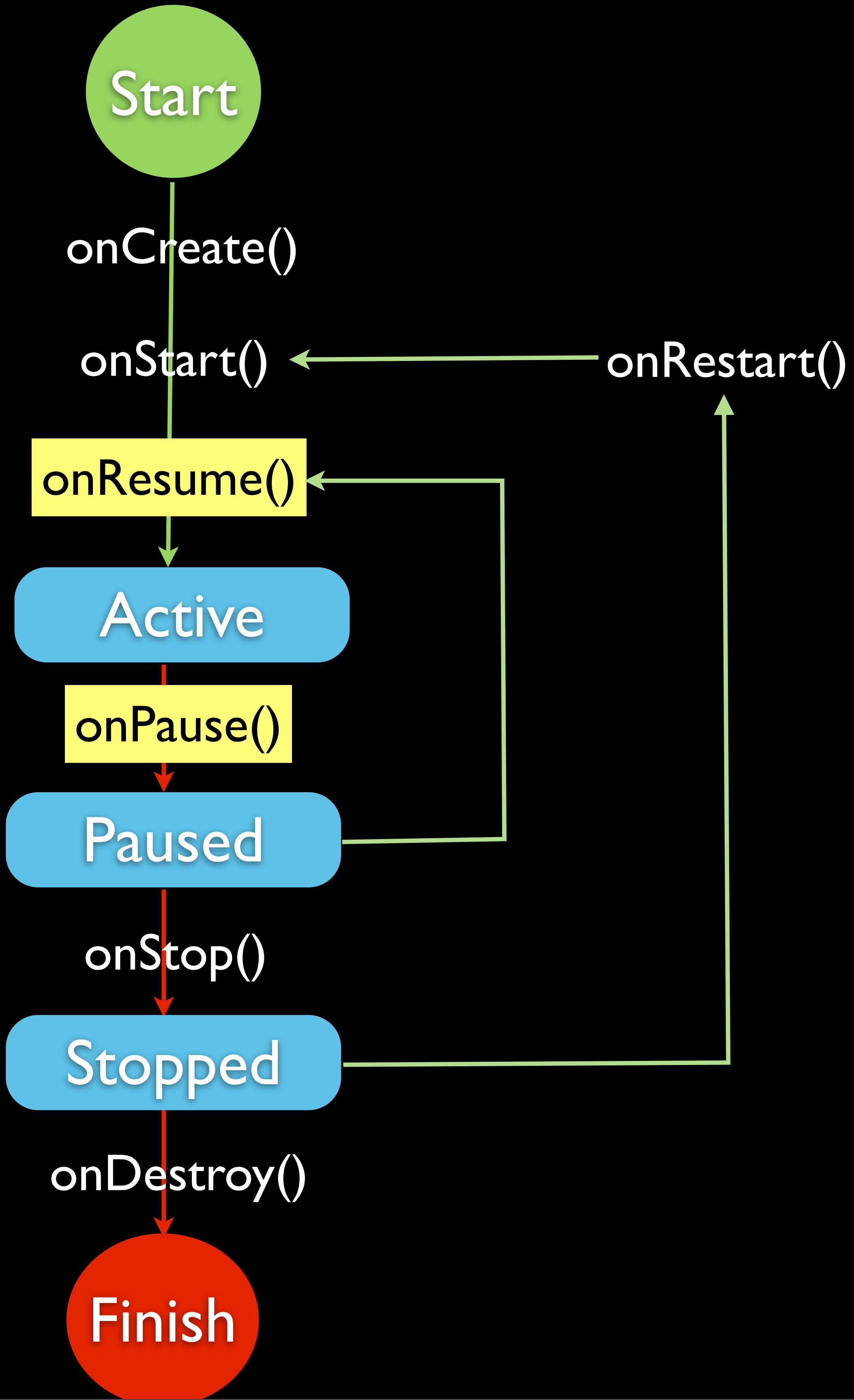
Lifecycle



Visible lifetime

- Visible but possibly not in foreground

Lifecycle

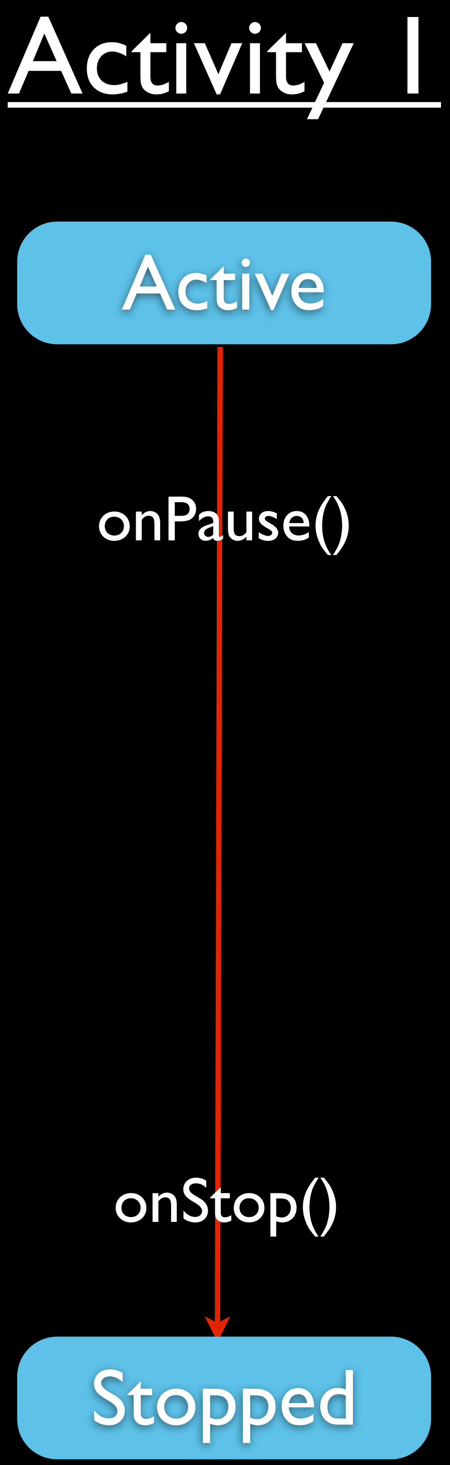


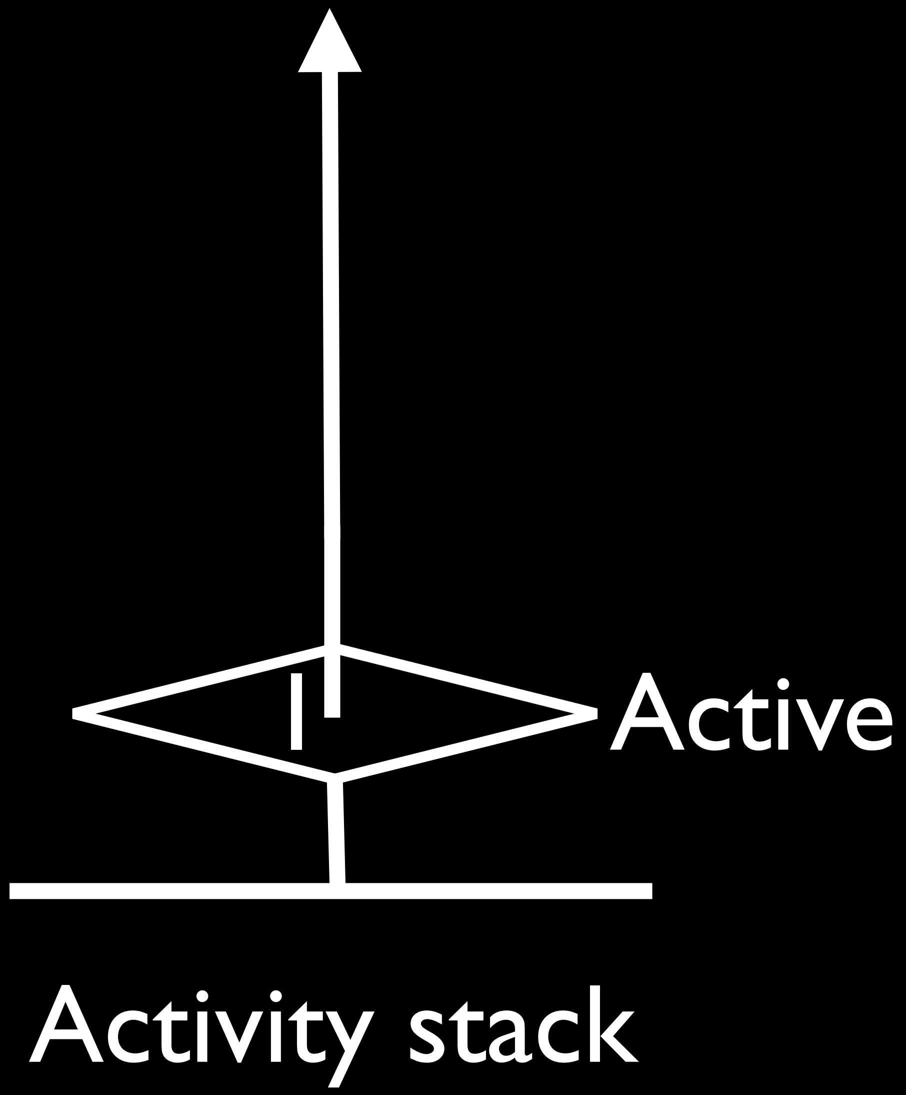
Foreground lifetime

- User interaction
- Frequent cycle
    - Example: Device screen sleep mode
    - Lightweight code

JAYWAY

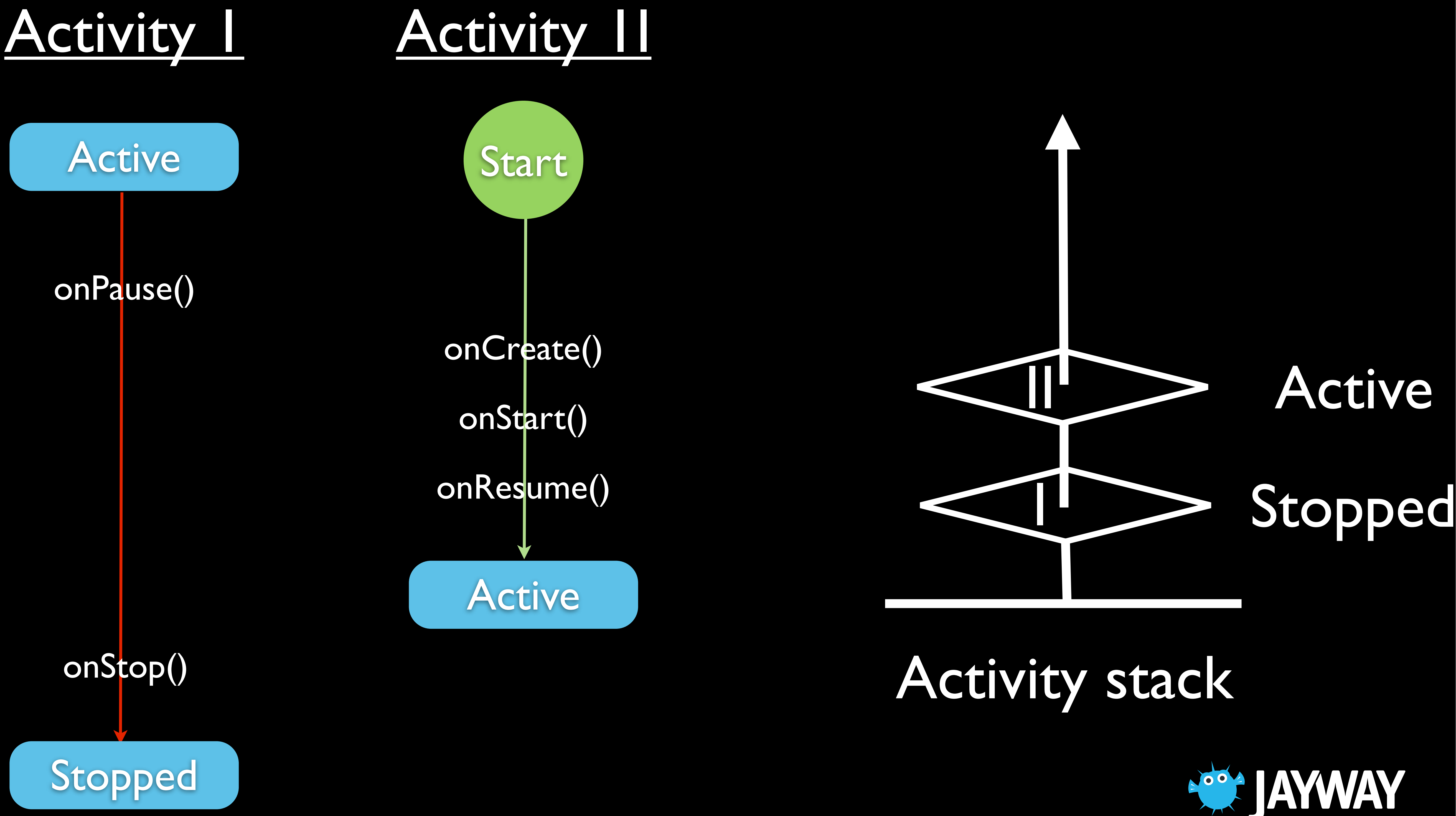
Activity transition



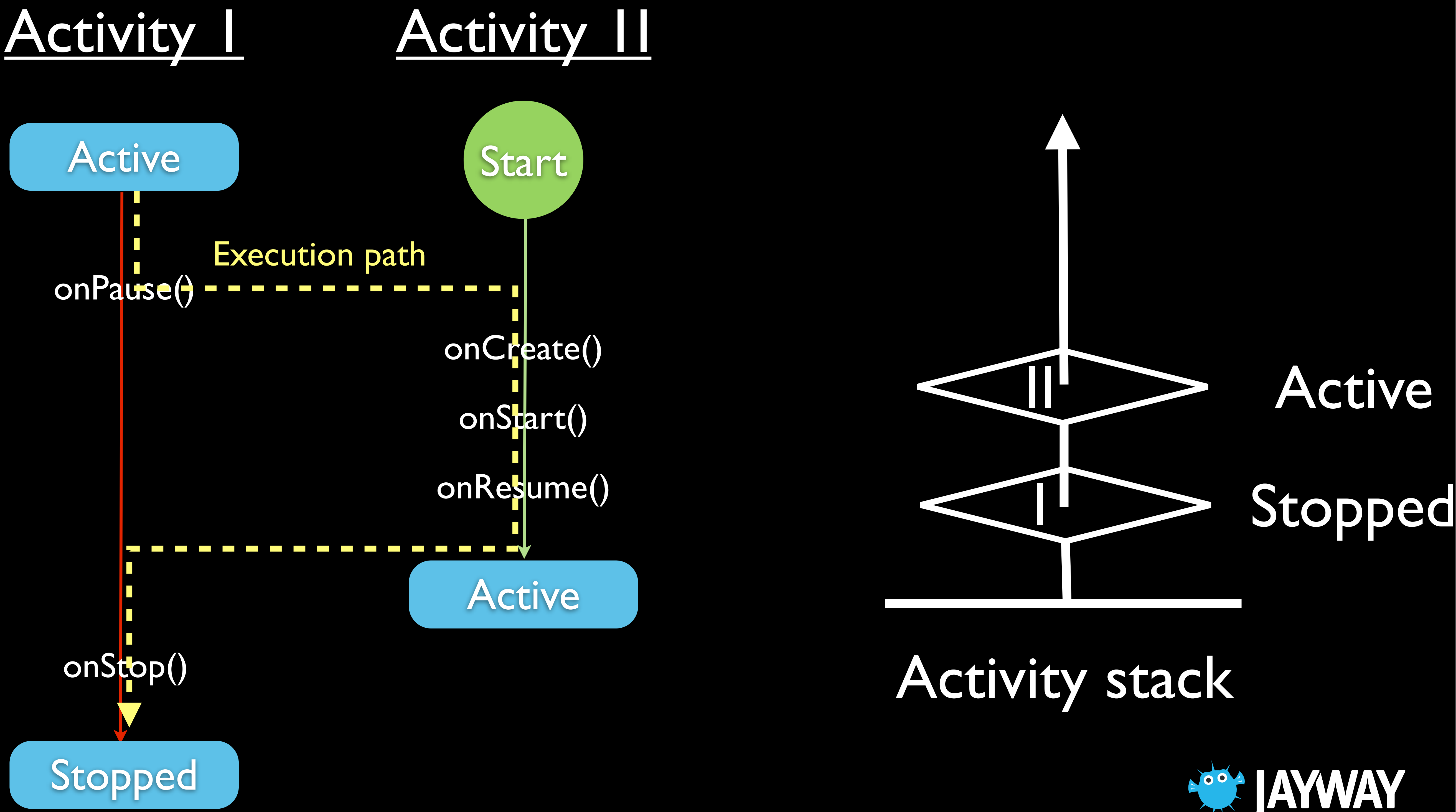


JAYWAY

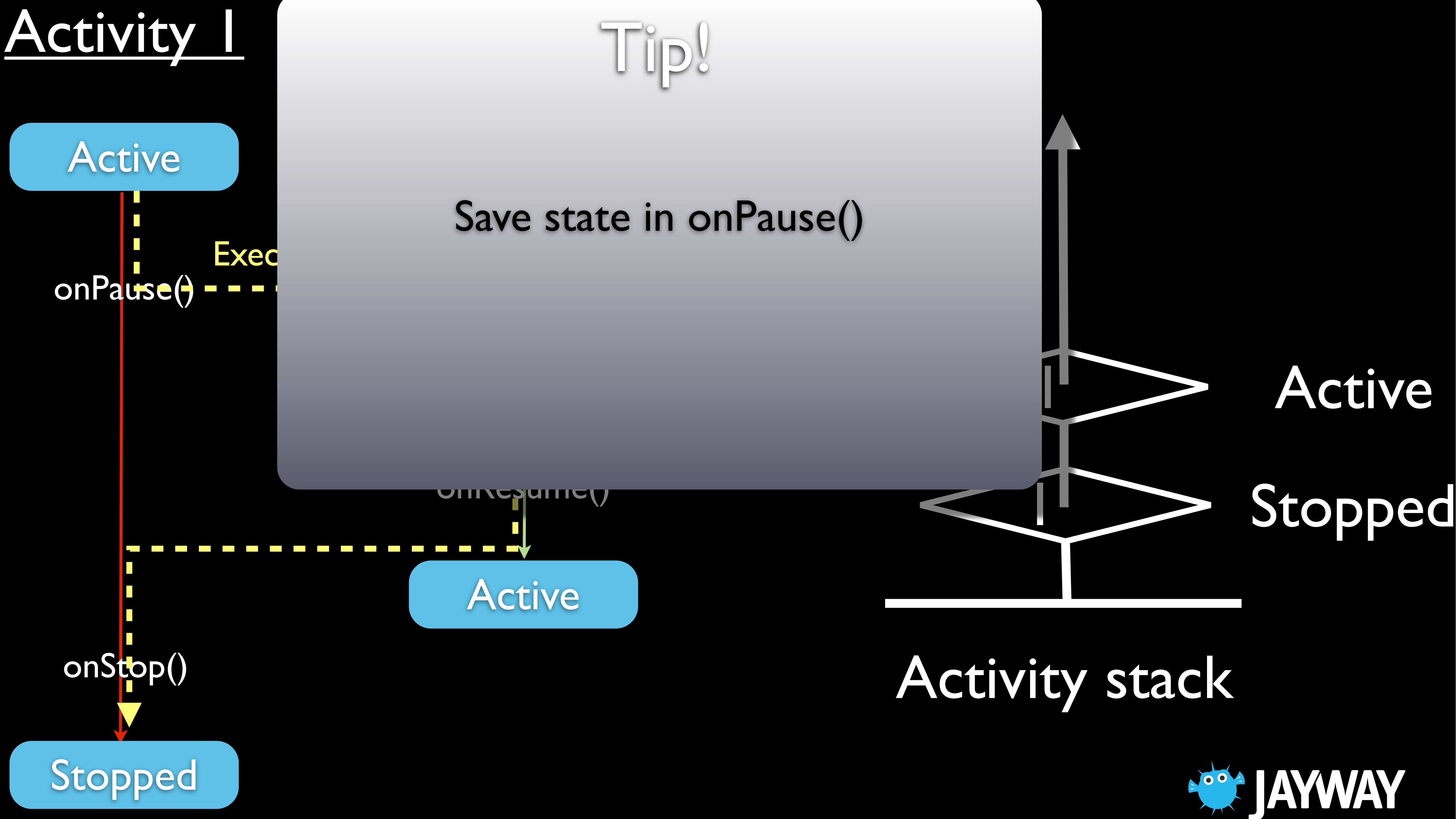
Activity transition



Activity transition



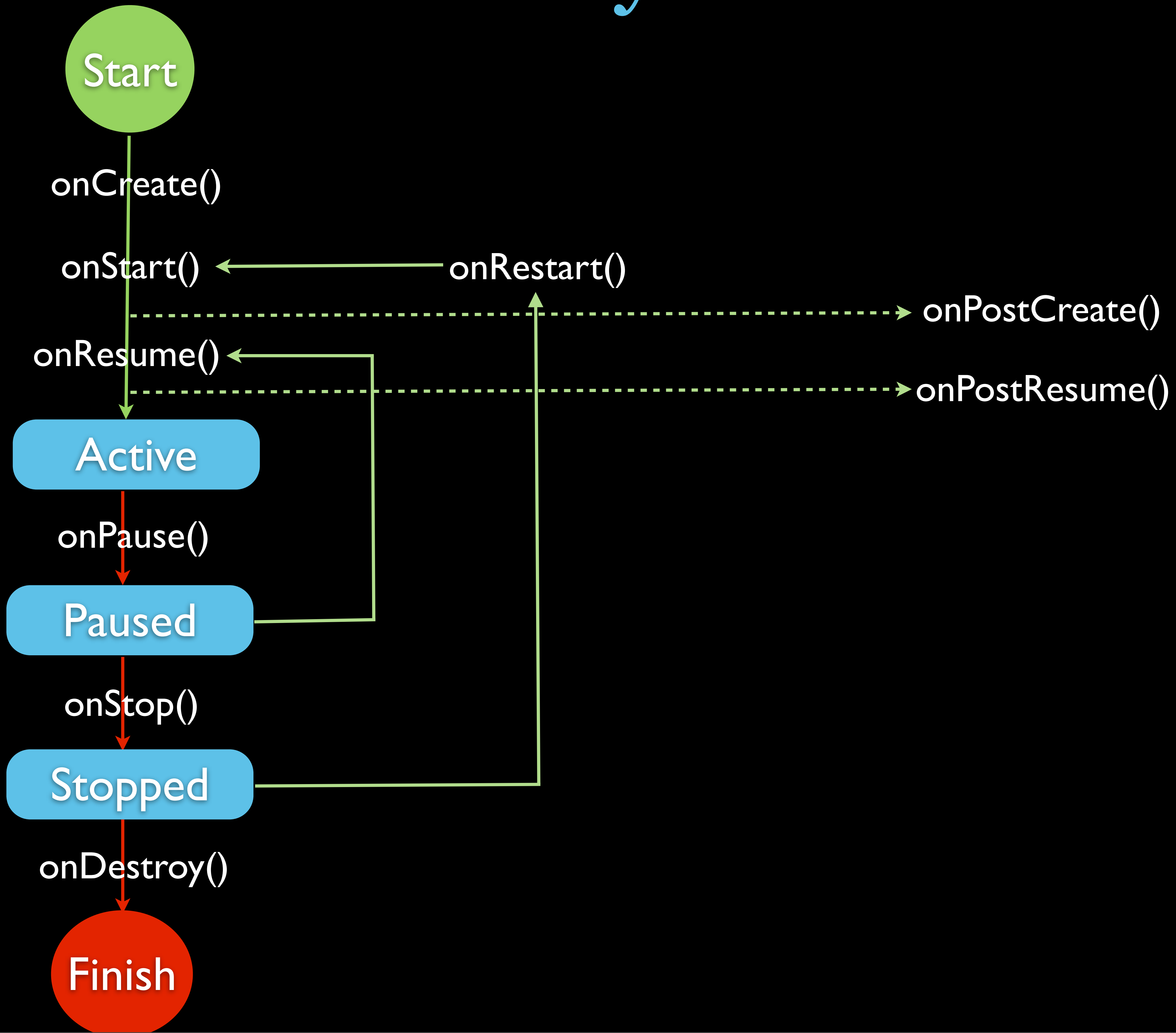
Activity transition



Tip!

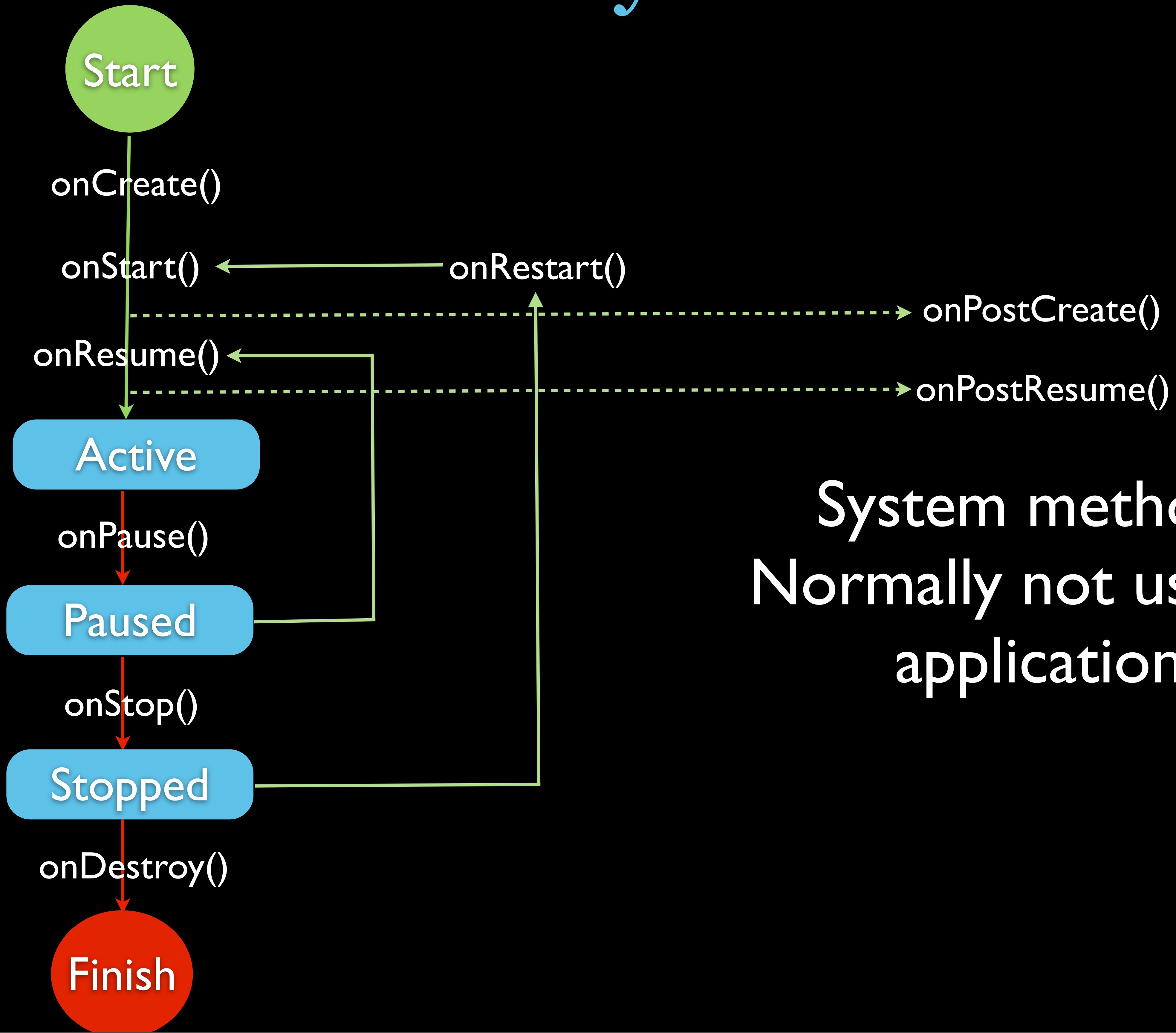
Save state in onPause()

Lifecycle Extras



JAYWAY

Lifecycle Extras

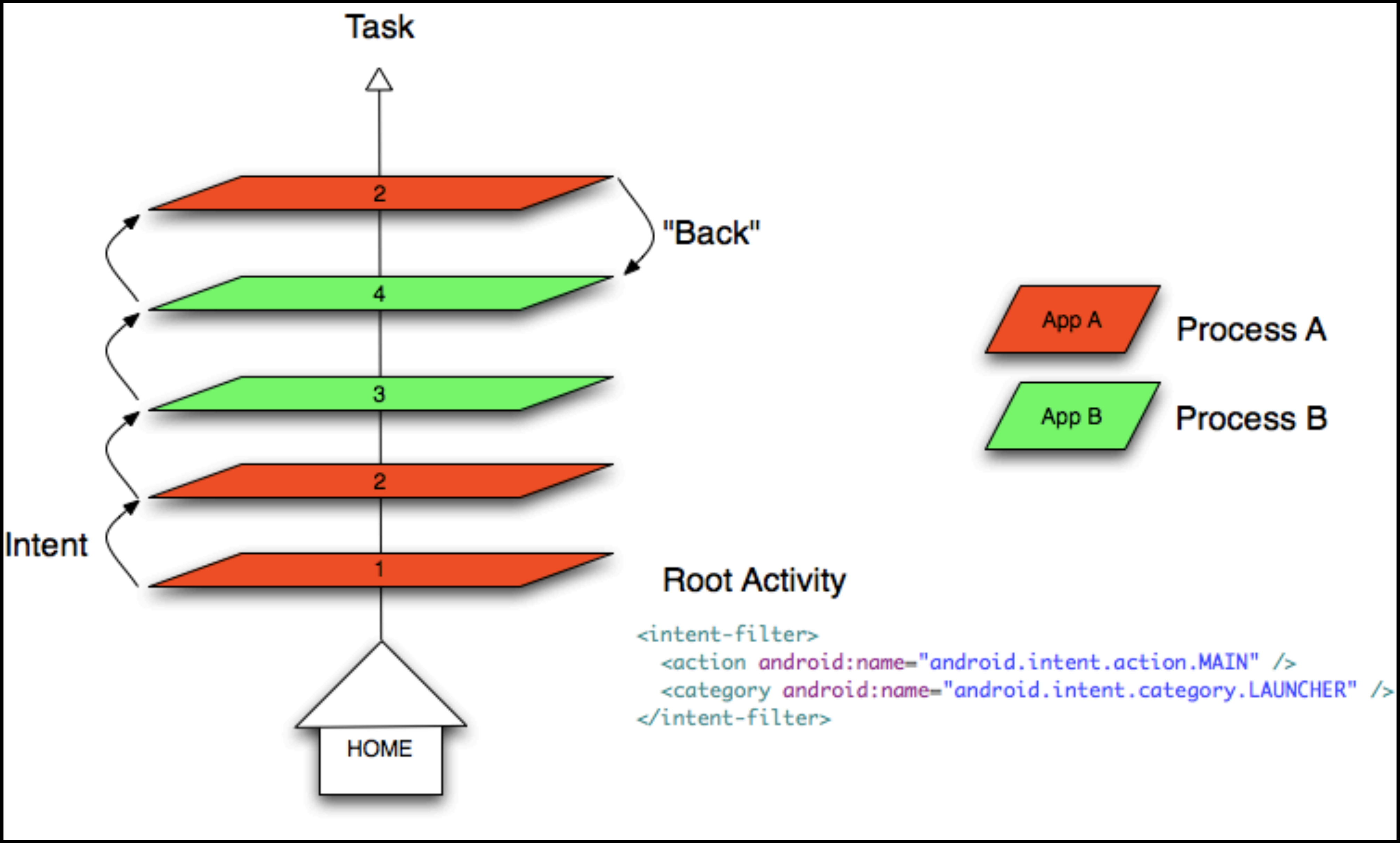


System methods. Normally not used by application.

Lifecycle changes

- The User navigates away from the Activity
  - To another Activity
    - onStop(), onDestroy() if finish() is called
  - BACK-button (Don’t override unless well motivated)
    - onDestroy()
  - HOME-button
    - onStop()
  - Other App in front, e.g. incoming call, onStop()
  - Screensaver
    - onPause()

Task


Task
2
"Back"
4
3
2
1
Intent
Root Activity
HOME
App A
Process A
App B
Process B

```
<intent-filter>
  <action android:name="android.intent.action.MAIN" />
  <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
```

System Cleanup

- System can shut down a process at any time due to low available resources
- The processes are ranked according to importance
    - Foreground process
    - Visible process
    - Service process
    - Background process
    - Empty process

Process Hierarchy

Foreground

- Activity in the foreground (onResume called)
- Service executing platform callback (onCreate, onStartCommand, onDestroy)
- BroadcastReceiver executing platform callback

Process Hierarchy

Foreground

- Activity in the foreground (onResume called)
- Service executing platform callback (onCreate, onStartCommand, onDestroy)
- BroadcastReceiver executing platform callback

Visible

- Visible Activity (onPause)
- Hosts Service bound to by visible Activity

JAYWAY

User Experience ► System Cleanup

Process Hierarchy

Foreground

- Activity in the foreground (onResume called)
- Service executing platform callback (onCreate, onStartCommand, onDestroy)
- BroadcastReceiver executing platform callback

Visible

- Visible Activity (onPause)
- Hosts Service bound to by visible Activity

Service

- Explicitly started Service (startService)

JAYWAY

User Experience ➤ System Cleanup

Process Hierarchy

Foreground

- Activity in the foreground (onResume called)
- Service executing platform callback (onCreate, onStartCommand, onDestroy)
- BroadcastReceiver executing platform callback

Visible

- Visible Activity (onPause)
- Hosts Service bound to by visible Activity

Service

- Explicitly started Service (startService)

Background

- Background Activity (onStop)

JAYWAY

User Experience → System Cleanup

Process Hierarchy

Foreground

- Activity in the foreground (onResume called)
- Service executing platform callback (onCreate, onStartCommand, onDestroy)
- BroadcastReceiver executing platform callback

Visible

- Visible Activity (onPause)
- Hosts Service bound to by visible Activity

Service

- Explicitly started Service (startService)

Background

- Background Activity (onStop)

Empty

- No active components

User Experience → System Cleanup

Process Hierarchy

Foreground
- Activity in the foreground (onResume called)
- Servi mmand, onDestroy)
- Broa

Visible
- Visibl
- Host

Service
- Expli

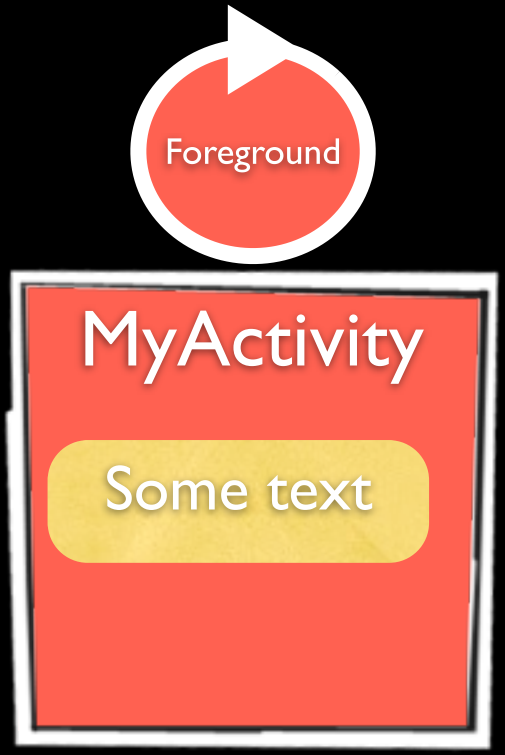
Background
- Background Activity (onStop)

Empty
- No active components

Tip!

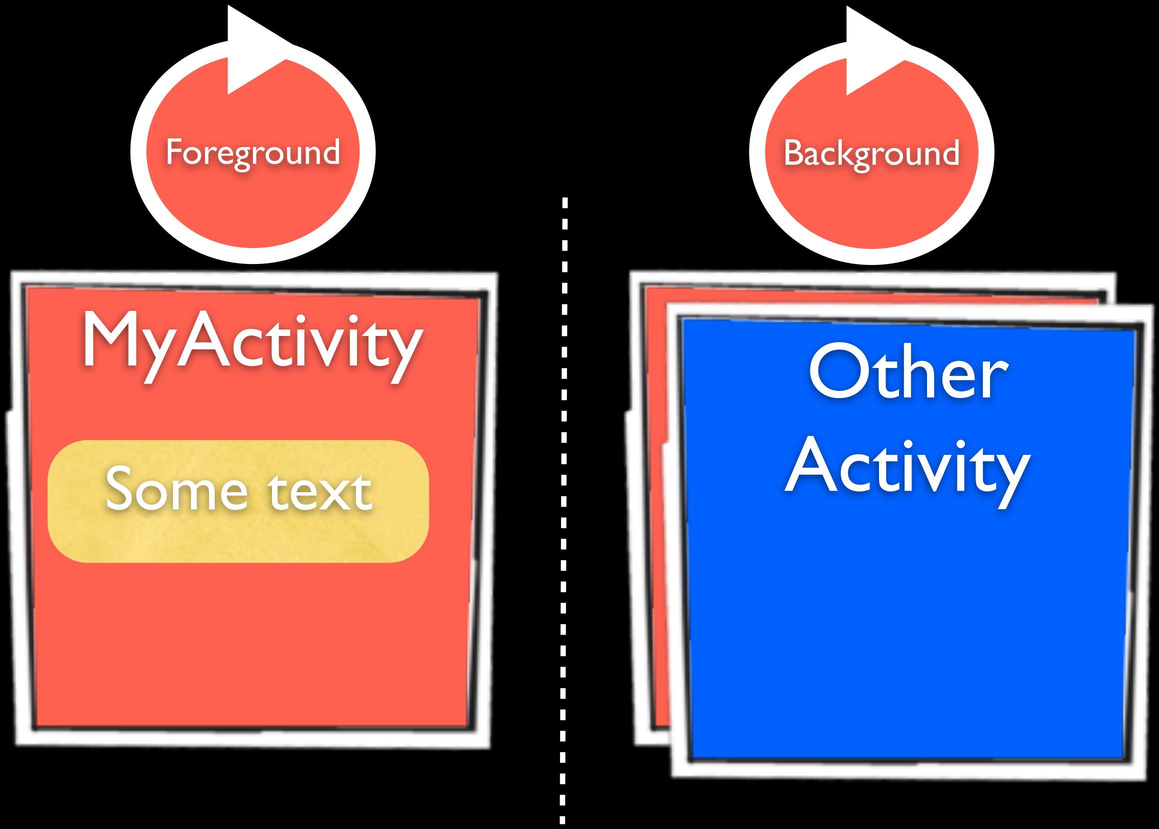
Long running tasks in Activity or BroadcastReceiver are best handled in a Service instead of local Thread. The process is then ranked higher and the risk of interruption is decreased.

Background Process Problem



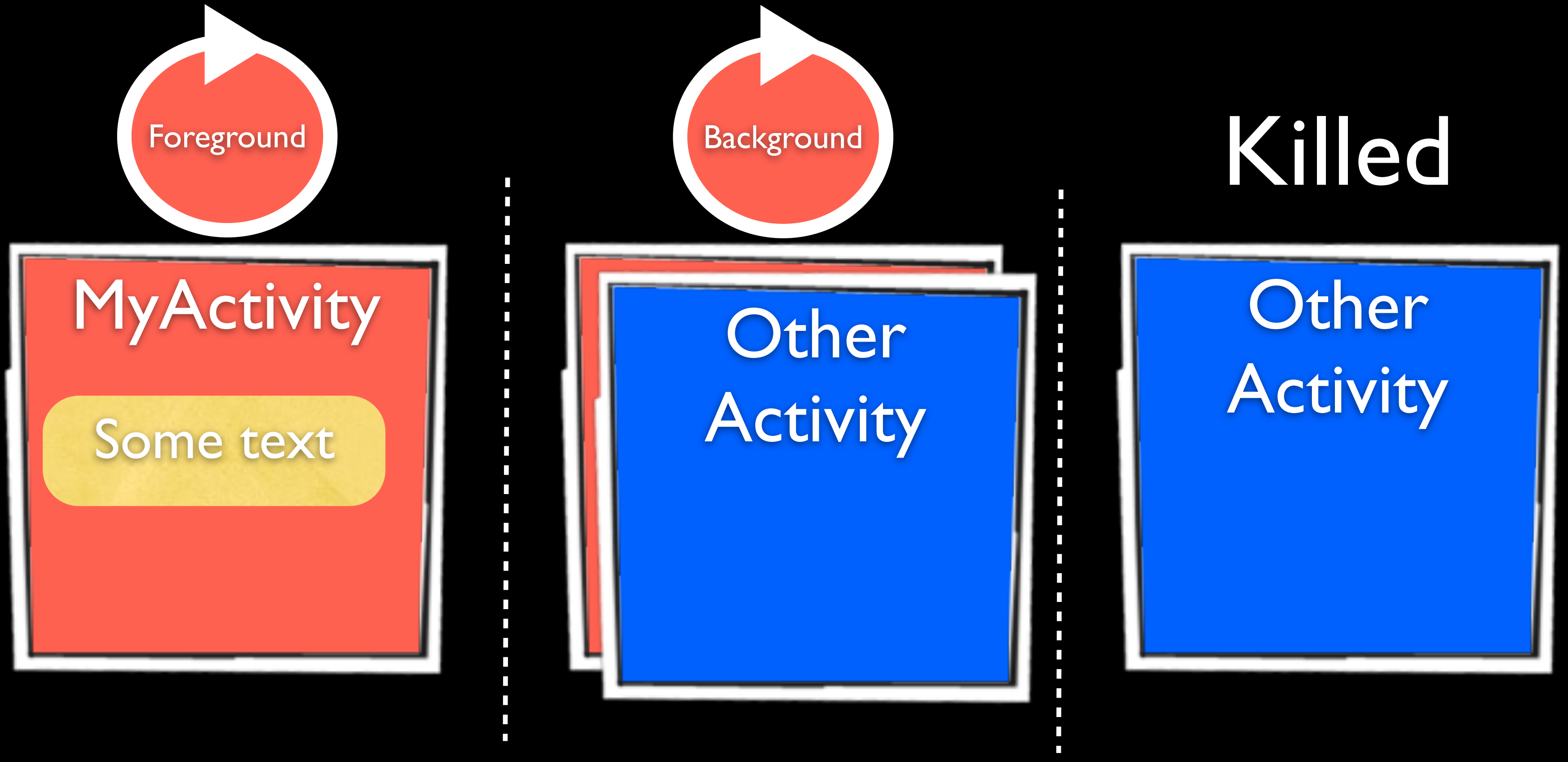
JAYWAY

Background Process Problem



OtherActivity starts

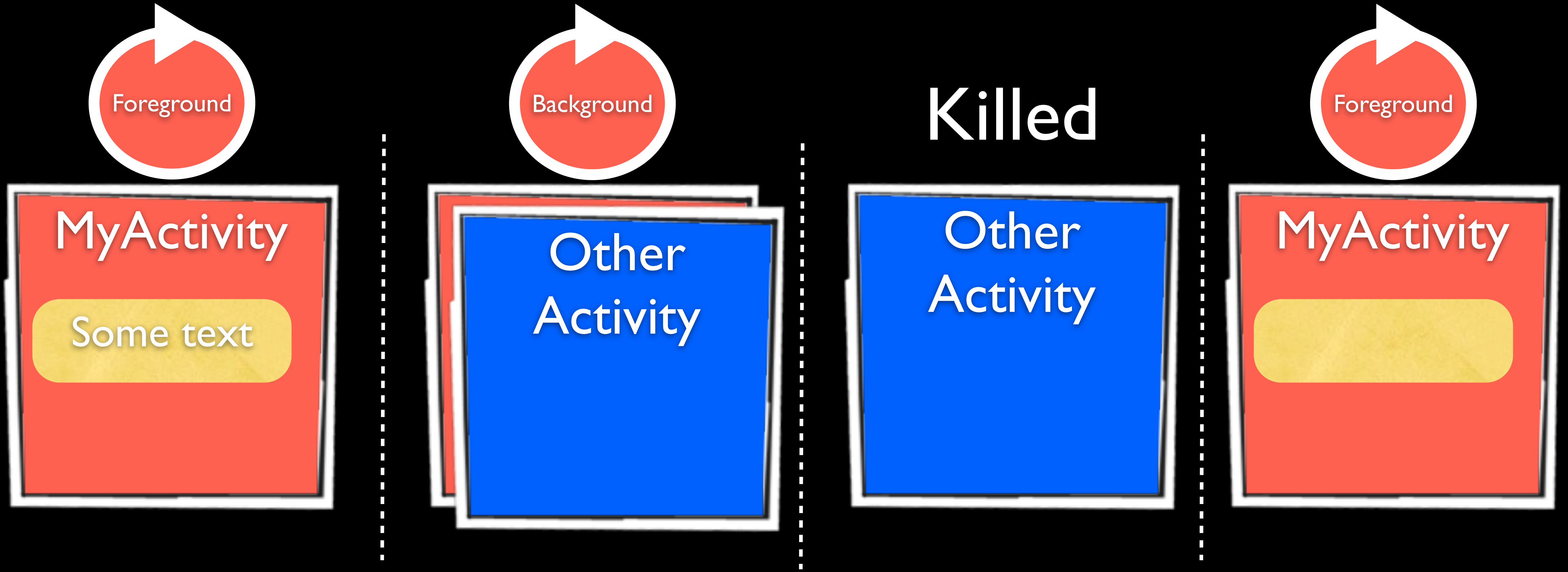
Background Process Problem



OtherActivity starts

System shuts down MyActivity

Background Process Problem



OtherActivity starts

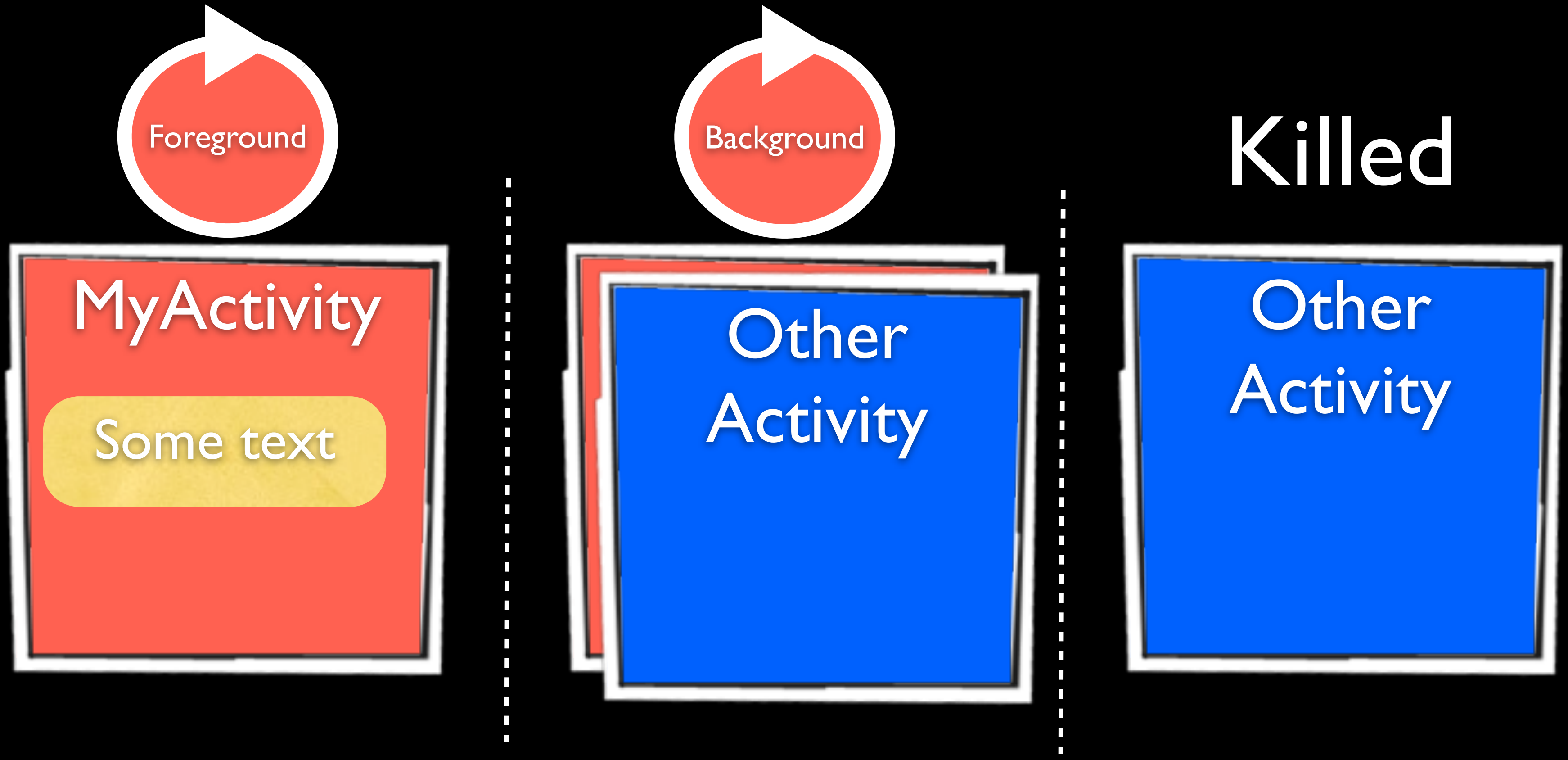
System shuts down MyActivity

User presses “Back”

Background Process Solution

- Activity.onSaveInstanceState(Bundle)
  - Called when Activity is destroyed by the system
  - Not a lifecycle method
  - Data that should be retrieved is set in a Bundle
    - put*-methods

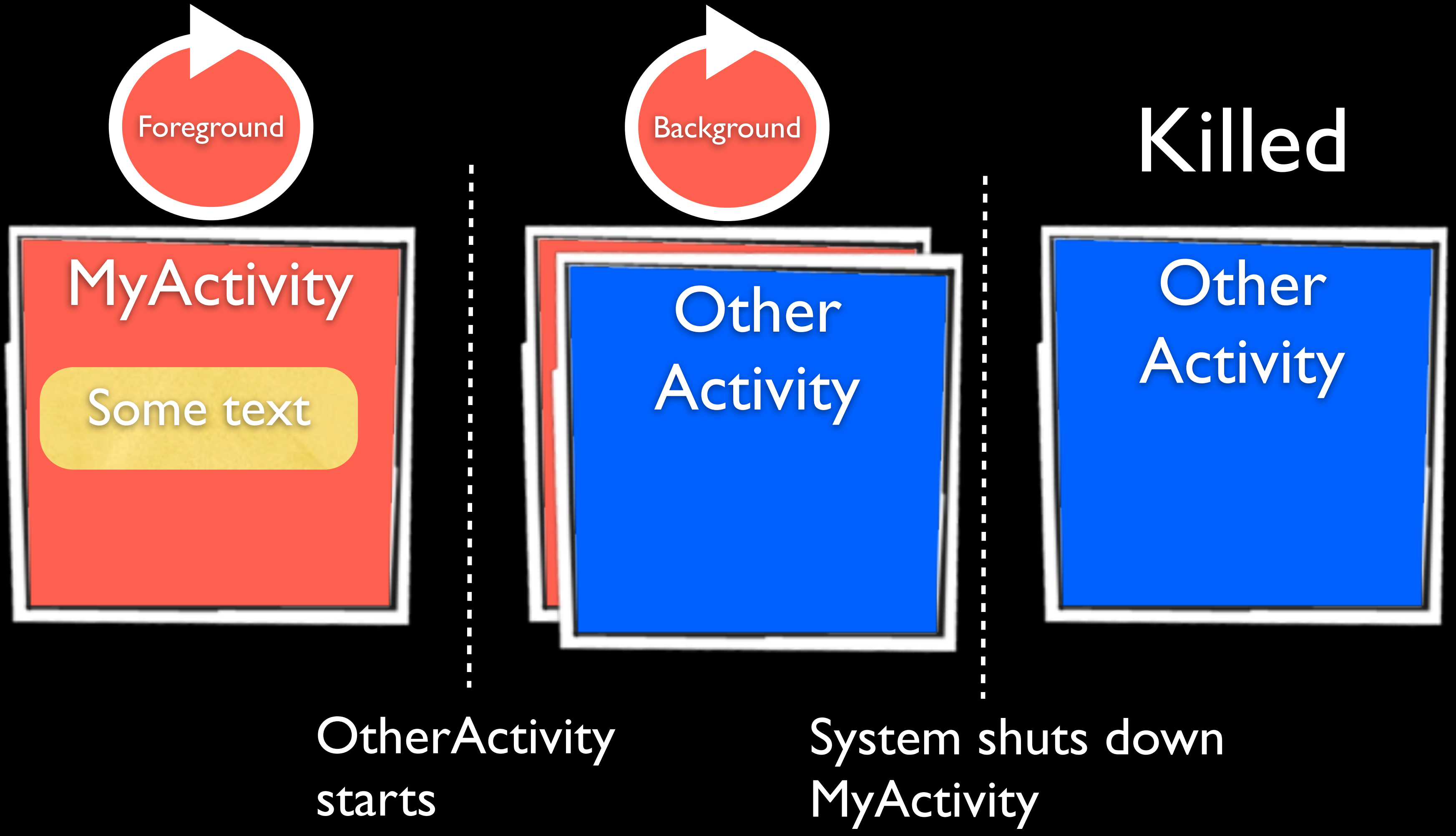
Background Process Solution



OtherActivity starts

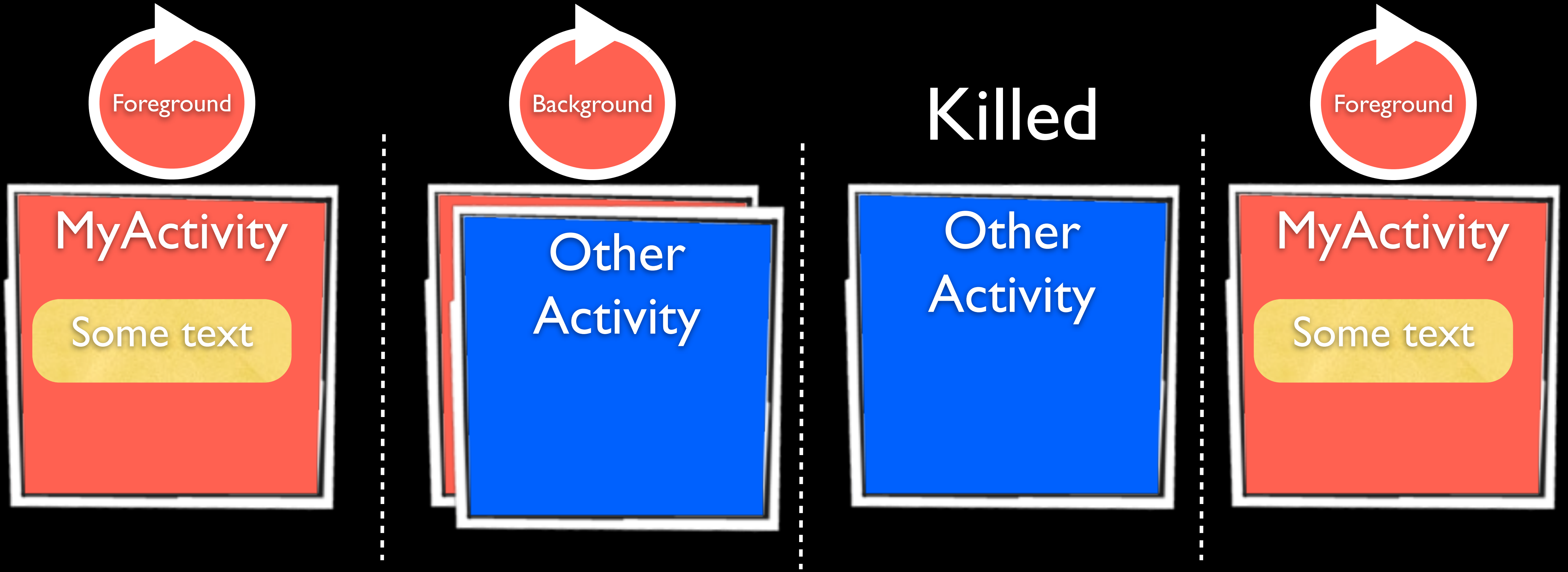
System shuts down MyActivity

Background Process Solution



```
@Override
protected void onSaveInstanceState(Bundle bundle) {
 super.onSaveInstanceState(outState);
 bundle.putString(key, “Some Text”);
}
```

Background Process Solution



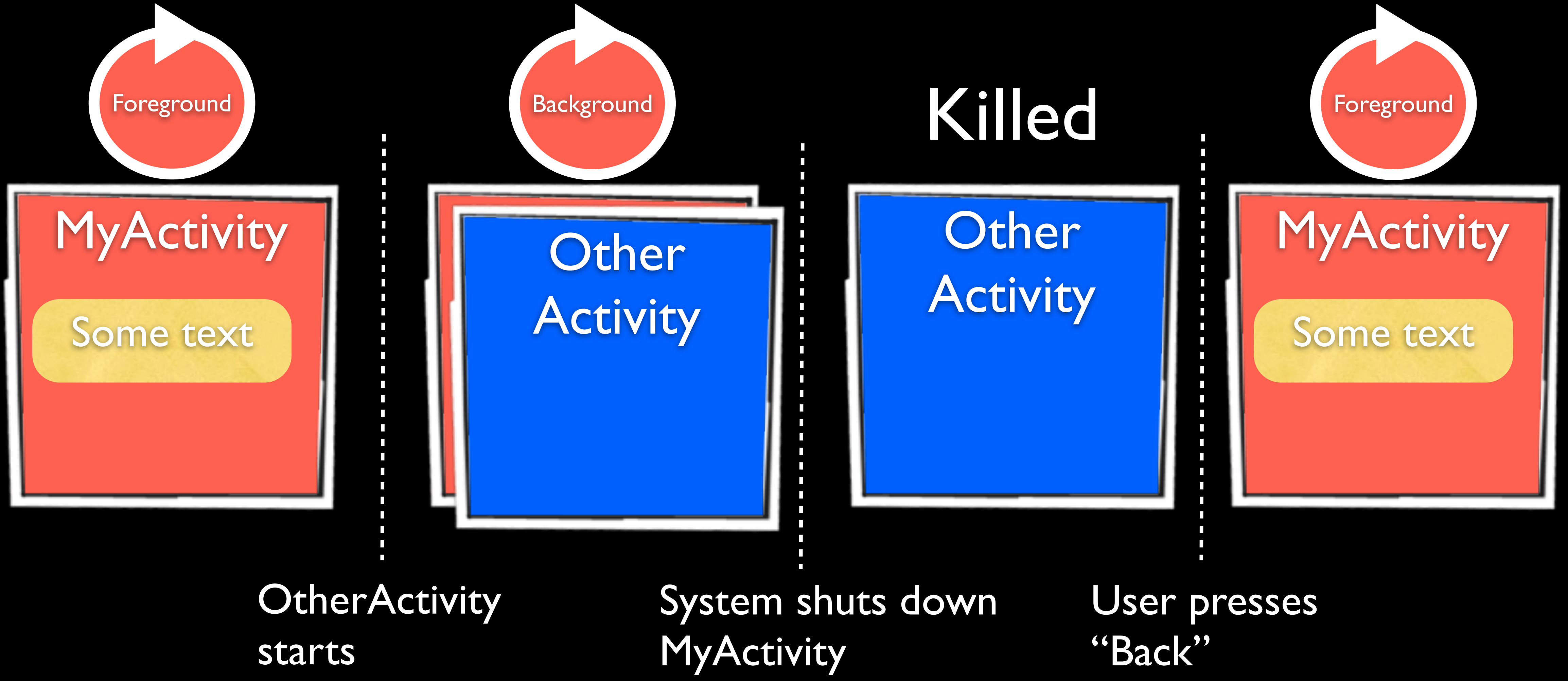
OtherActivity starts

System shuts down MyActivity

User presses "Back"

```
@Override
protected void onSaveInstanceState(Bundle bundle) {
 super.onSaveInstanceState(outState);
 bundle.putString(key, "Some Text");
}
```

Background Process Solution



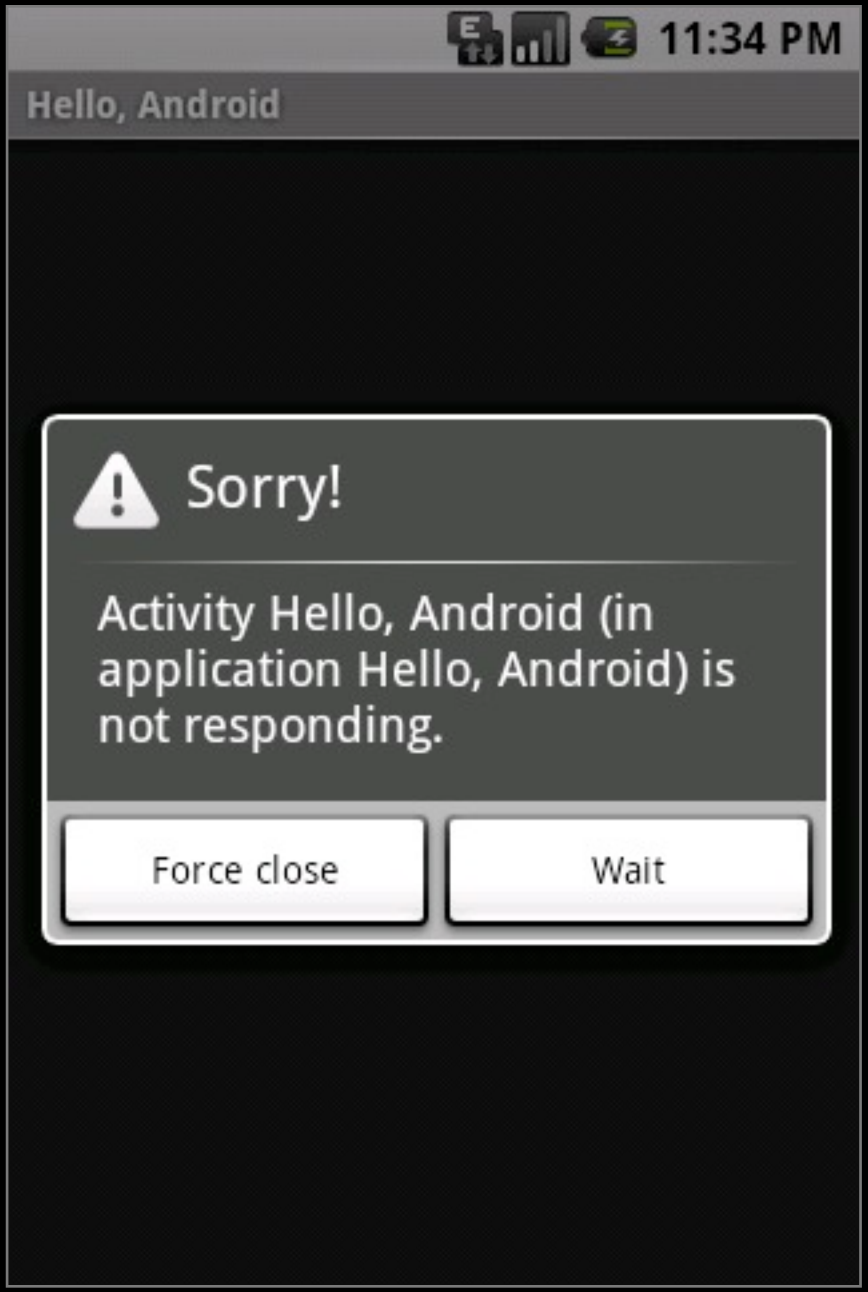
```
@Override
protected void onSaveInstanceState(Bundle bundle) {
 super.onSaveInstanceState(outState);
 bundle.putString(key, "Some Text");
}
```

```
@Override
protected void onCreate(Bundle bundle) {
 super.onCreate(bundle);
 String str = bundle.getString(key);
 ...
}
```

Master the thread model

- Each application runs in one Linux process by default
- Single-threaded model - Main / UI thread
    - All components and system calls run in the UI thread.
    - Blocking the UI thread blocks all components
- UI toolkit not thread-safe
    - All UI operations must be made on the UI thread

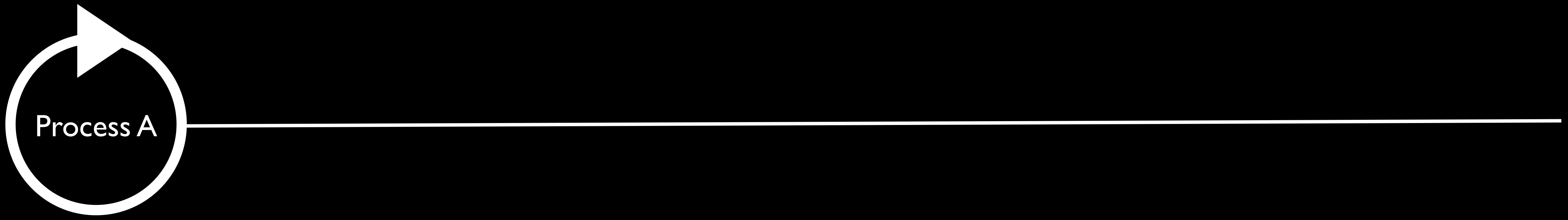
Application Not Responding



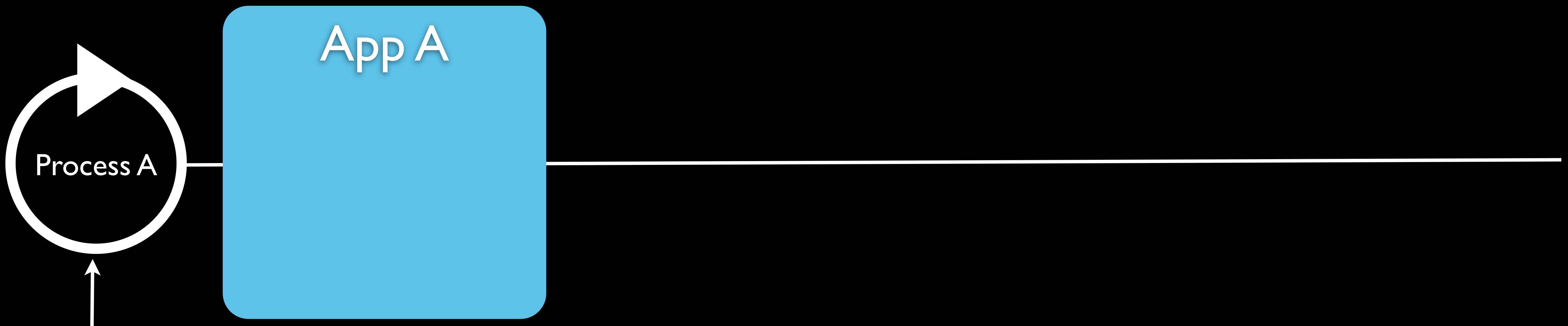
Activity Hello, Android (in application Hello, Android) is not responding.

JAYWAY

Thread Model

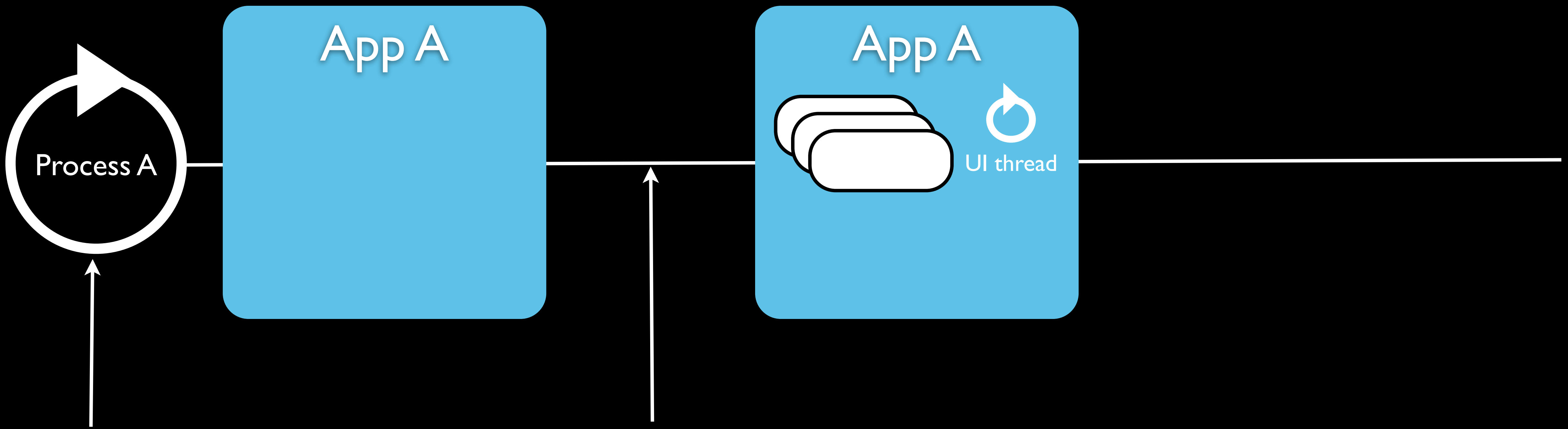


Thread Model



System starts
App A

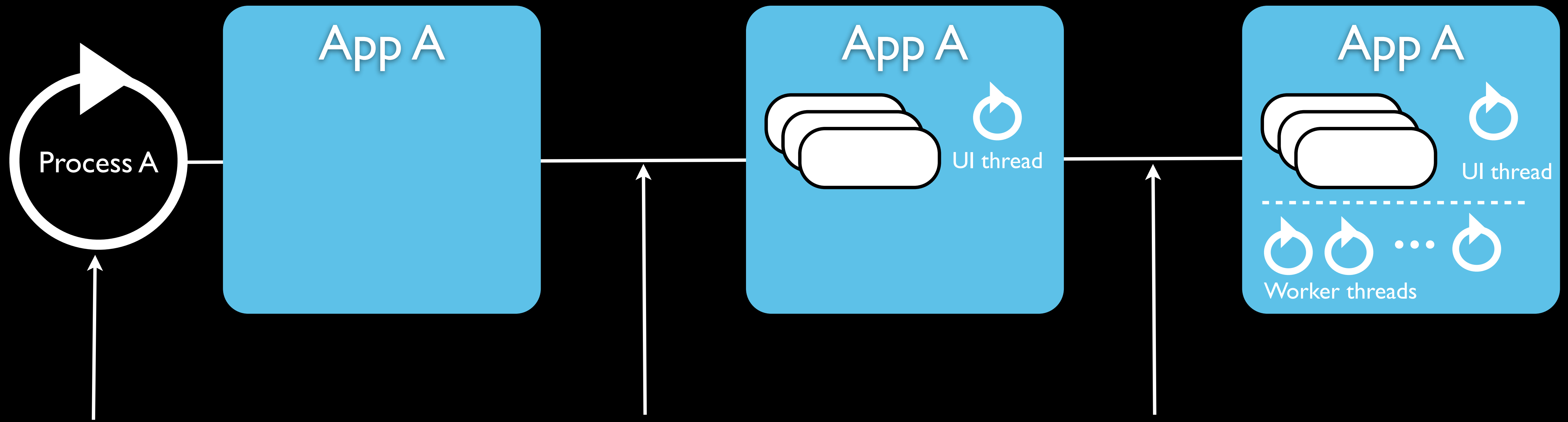
Thread Model



System starts App A

App starts components

Thread Model



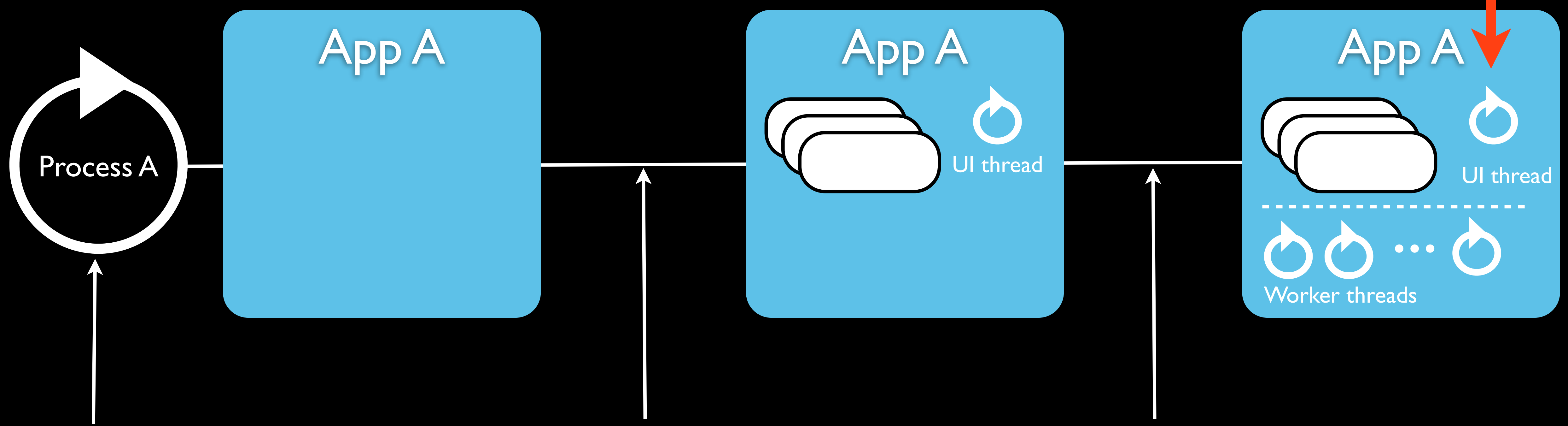
System starts App A

App starts components

App starts worker threads

Thread Model

- Blocks the UI thread
- Use only lightweight tasks
- All UI manipulations

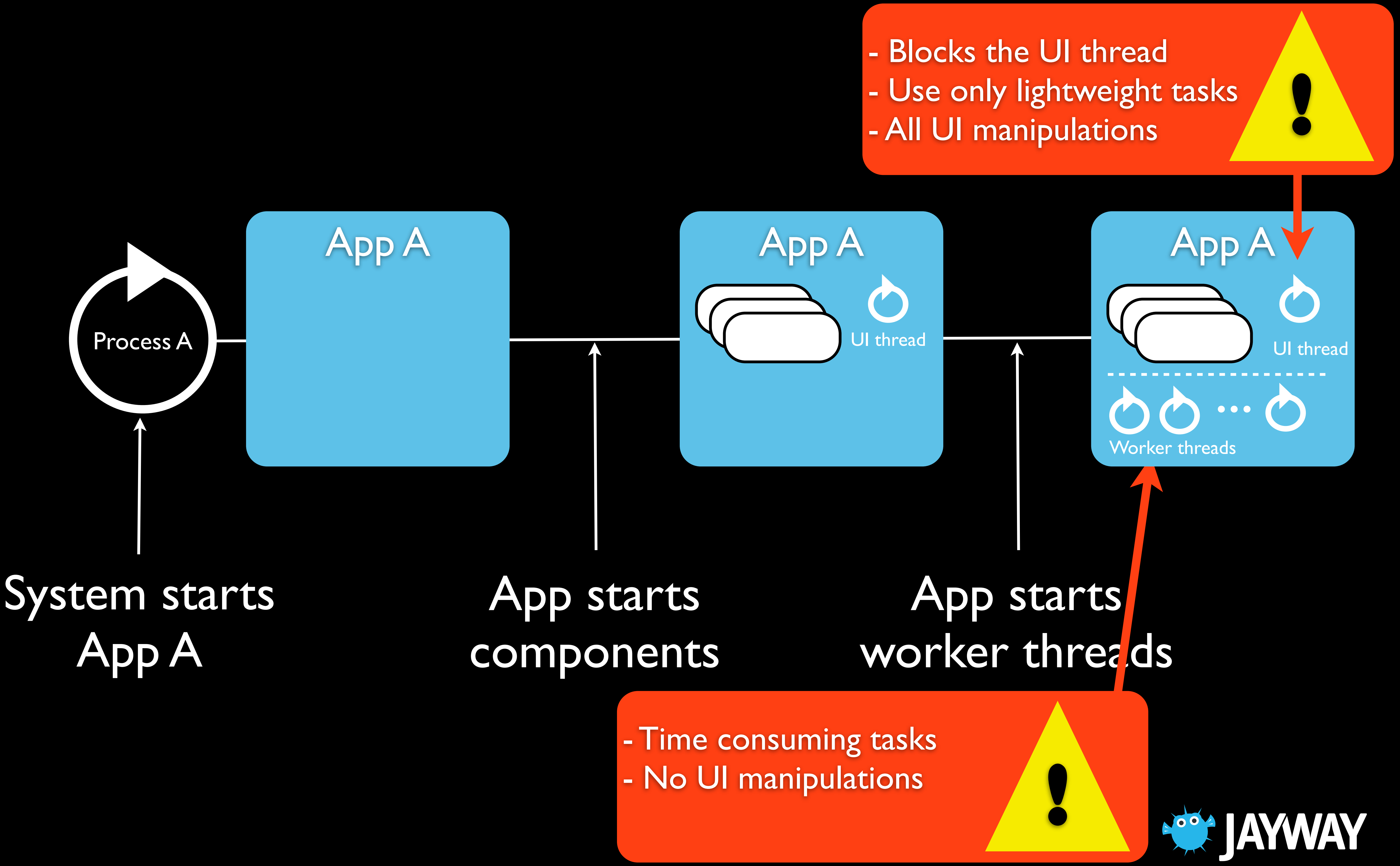


System starts App A

App starts components

App starts worker threads

Thread Model



JAYWAY

Thread Handler API

1. Activity.runOnUiThread(Runnable)
2. View.post(Runnable)
3. Handler

```
public class MyActivity extends Activity {

    private TextView textView;

    private void startLongRunningOperation() {

       new Thread(new Runnable() {
          public void run() {
            results = doSomethingExpensive();
            runOnUiThread(new Runnable() {
               public void run() {
                  textView.setText(results);
               }
            });
          }
       }).start();
    }
```

Thread Handler API

1. Activity.runOnUiThread(Runnable)
2. View.post(Runnable)
3. Handler

```
private TextView textView;

private void startLongRunningOperation() {

   new Thread(new Runnable() {
      public void run() {
        results = doSomethingExpensive();
        textView.post(new Runnable() {
           public void run() {
              textView.setText(results);
           }
        });
      }
   }).start();
}
```

Thread Handler API

1. Activity.runOnUiThread(Runnable)
2. View.post(Runnable)
3. Handler

```
public class MyActivity extends Activity {

    private TextView textView;

    // Need handler for callbacks to the UI thread
    final Handler handler = new Handler();

    // Create runnable for posting
    final Runnable updateResults = new Runnable() {
        public void run() {
            // UPDATE UI WITH “results”
            textView.setText(results);
        }
    };

    private void startLongRunningOperation() {

        //Fire off a thread to do some work that we shouldn't do
        //directly in the UI thread
        Thread t = new Thread() {
            public void run() {
                results = doSomethingExpensive();
                handler.post(updateResults);
            }
        };
        t.start();
    }
```

Thread Handler API

1. Activity.runOnUiThread(Runnable)
   - Possible to update many UI objects
2. View.post(Runnable)
   - Updates one UI object
3. Handler
   - Generic mechanism not only UI manipulation

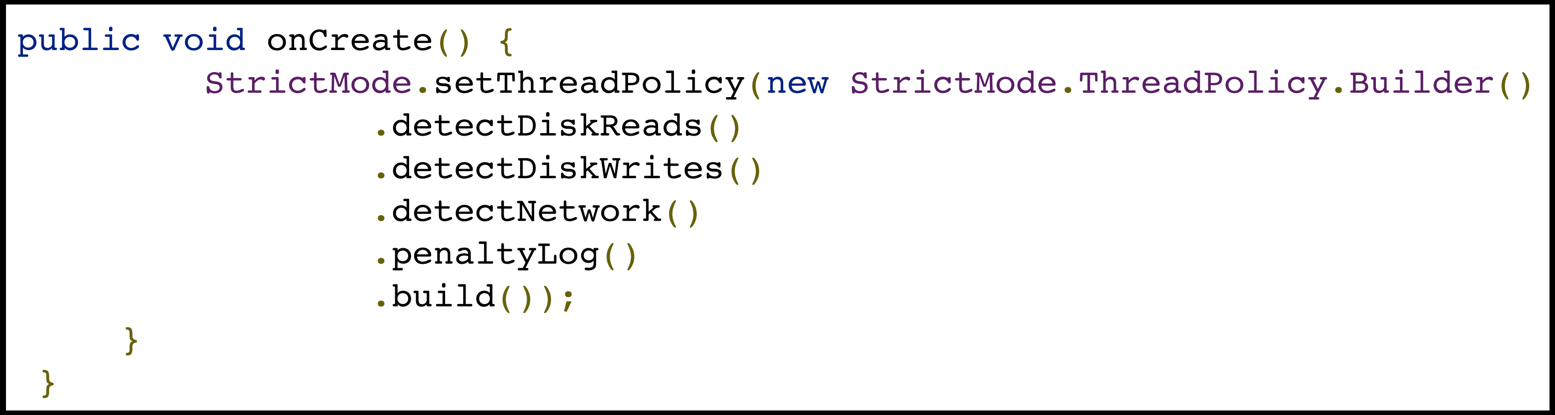
JAYWAY

AsyncTask

- Handles thread management
- Solves common use case
- Code separation
- Subclass
- Create on the UI thread

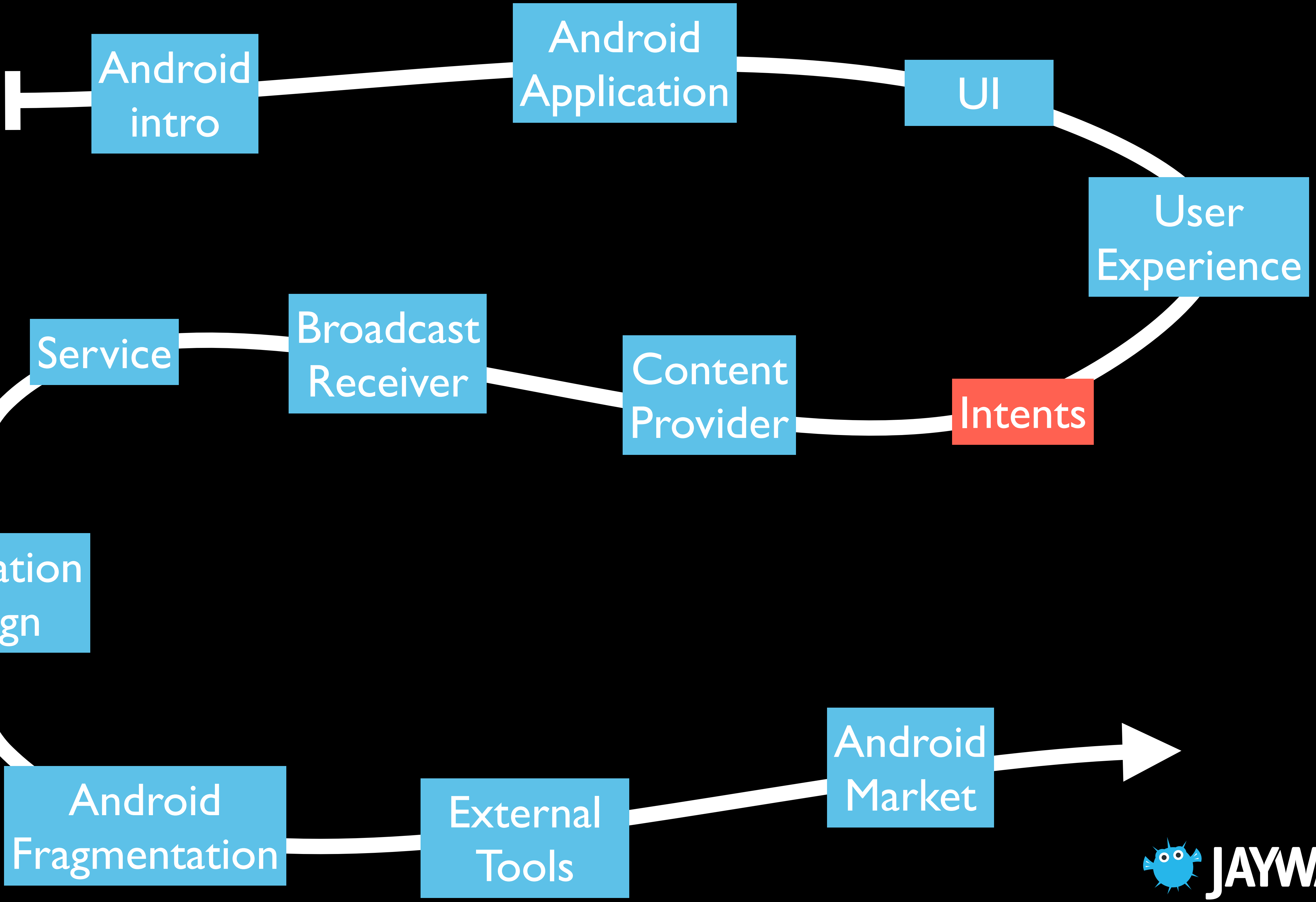
StrictMode

- Development API defined added in 2.3/Gingerbread
- Define thread policy for what’s allowed
  - Detect: Network access, disk reads/writes
  - Act: Log, show dialog, crash, dropbox



```
public void onCreate() {
        StrictMode.setThreadPolicy(new StrictMode.ThreadPolicy.Builder()
                .detectDiskReads()
                .detectDiskWrites()
                .detectNetwork()
                .penaltyLog()
                .build());
     }
 }
```

Agenda



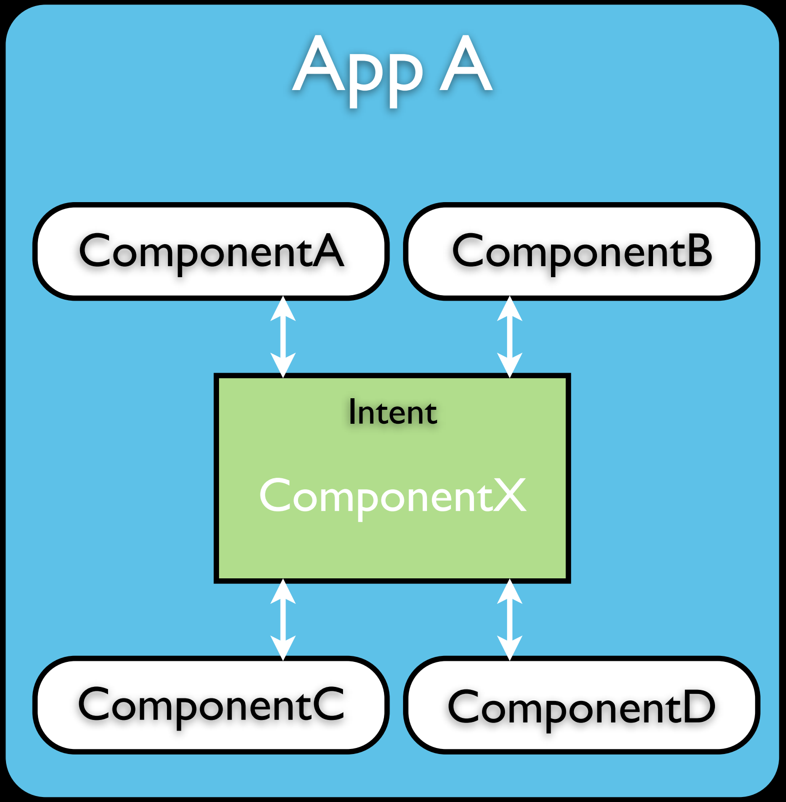
JAYWAY

Intent

- Android message system
- Intent is a declare of need
- “What you want to do?”
    - “I want to lookup a contact record”
    - “I want to launch this web site”
    - “Show the order confirmation screen”
- Activate components
    - Activity, Service, BroadcastReceiver

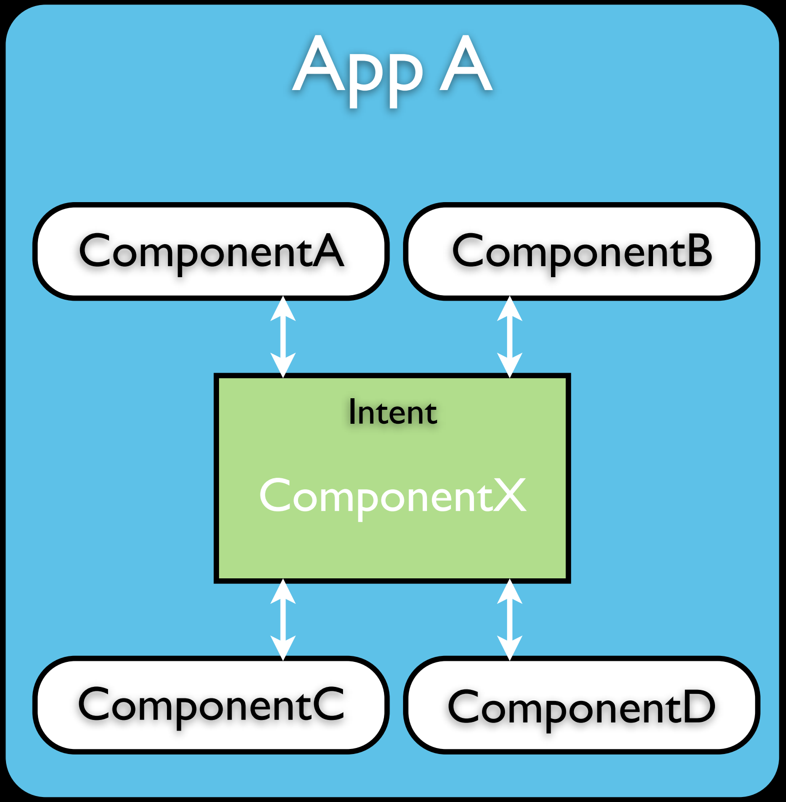
Types

Explicit Intent

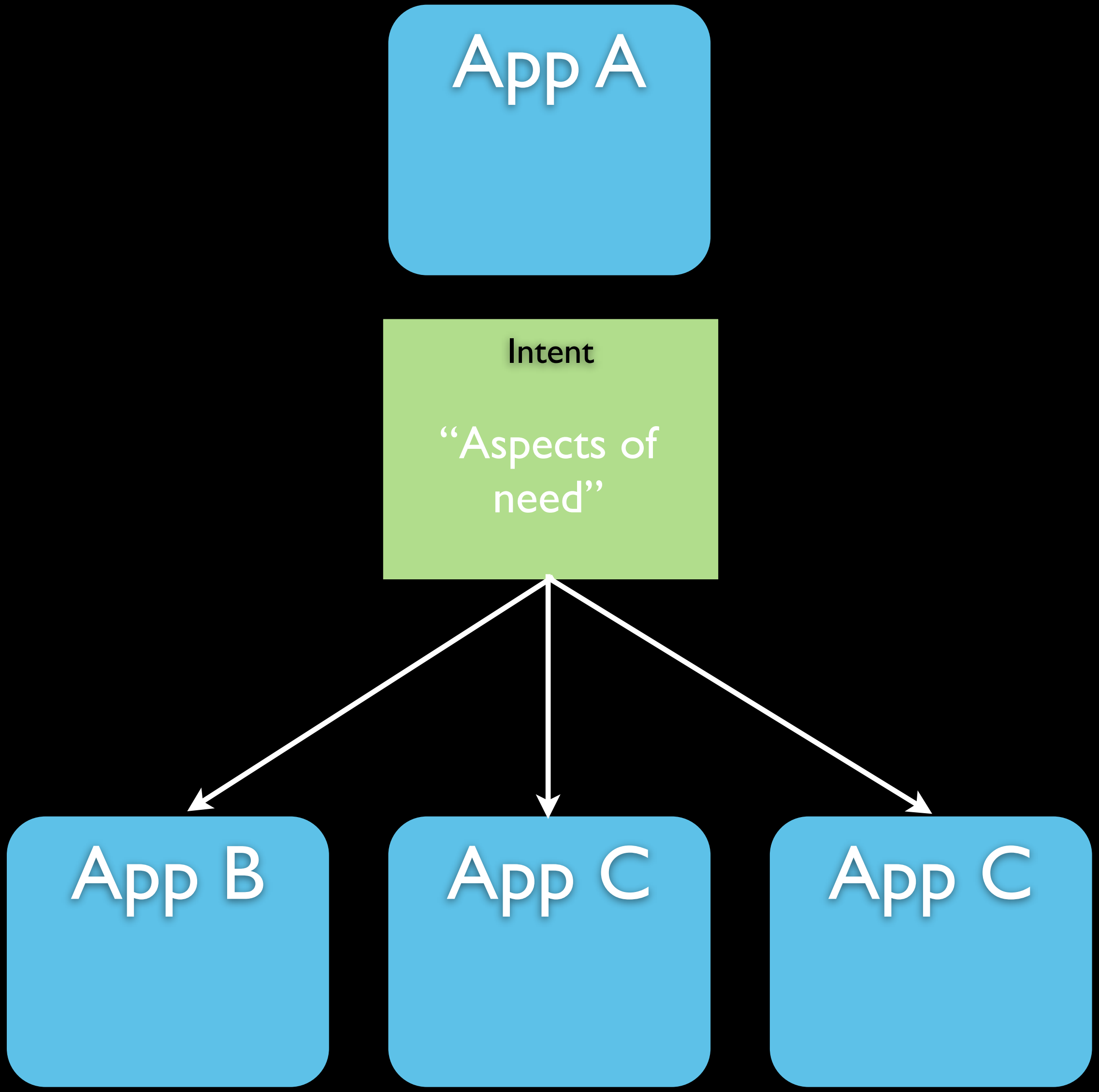


Types

Explicit Intent

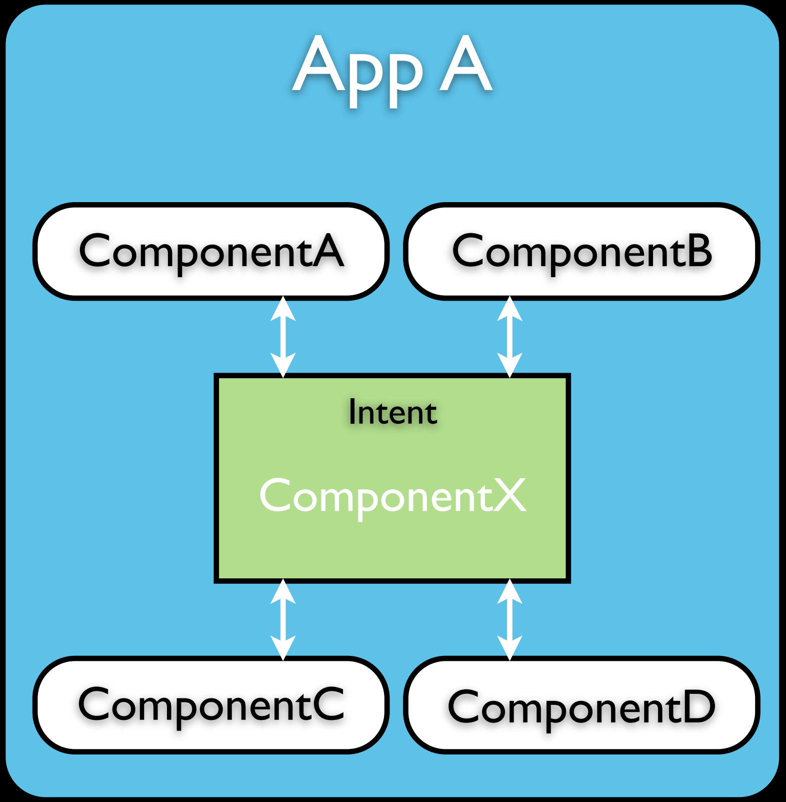


Implicit Intent

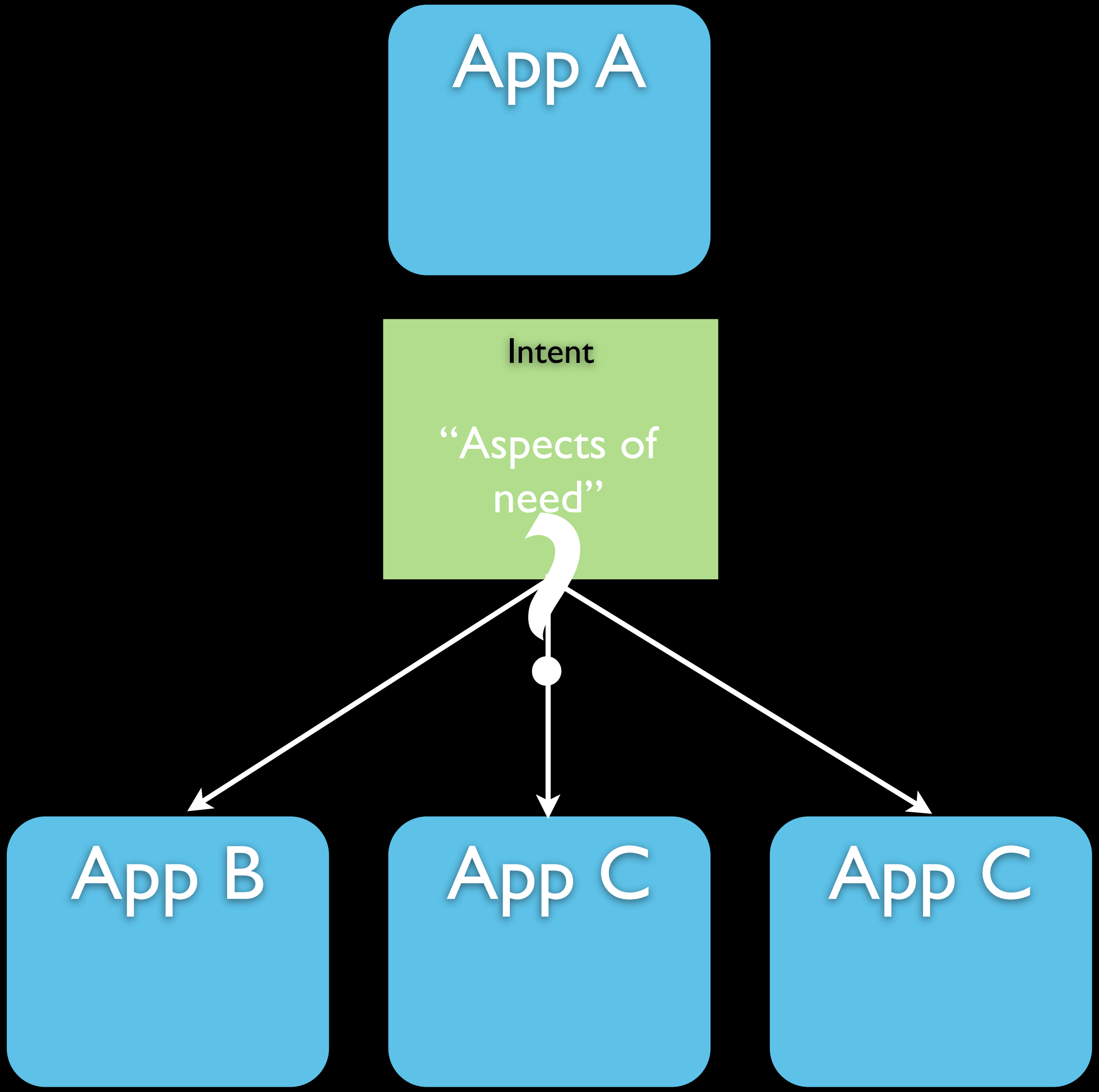


Types

Explicit Intent



Implicit Intent



JAYWAY

Types

- Explicit intent
  - Directed to a specific component in the application
  - Compile-time binding

```
Intent intent = new Intent(this, MyOtherActivity.class);
startActivity(intent);
```

- Implicit intent
  - Directed to any component that can handle the intent
  - Run-time binding

```
Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse(uriString));
startActivity(intent);
```

Definition

- Component name
- Action
- Data
- Category
- Extras
- Flags

Definition

- Component name
- Action
- Data
- Category
- Extras
- Flags

- Package name + Class name
- Explicit intent
- Application internal

Definition

- Component name
- Action
- Data
- Category
- Extras
- Flags

- String: Names the action to perform
  - ACTION_EDIT, ACTION_VIEW, etc.
  - System defined
    - Intent class
    - Application
  - User defined
    - Package name + Action name

JAYWAY

Definition

- Component name
- Action
- Data
- Category
- Extras
- Flags

- URI and/or MIME_TYPE
- Specifies the content to act upon

Definition

- Component name
- Action
- Data
- Category
- Extras
- Flags

- Kind of component that can handle the intent
- CATEGORY_BROWSABLE, etc.

Definition

- Component name
- Action
- Data
- Category
- Extras
- Flags

- Key-Value pairs with extra information

Definition

- Component name
- Action
- Data
- Category
- Extras
- Flags

- Typically: Instruct how an Activity should be launched

IntentFilter

- Implicit intents
  - Define intents that can be received
- IntentFilter defines
  - Action, Data, Category
- AndroidManifest.xml

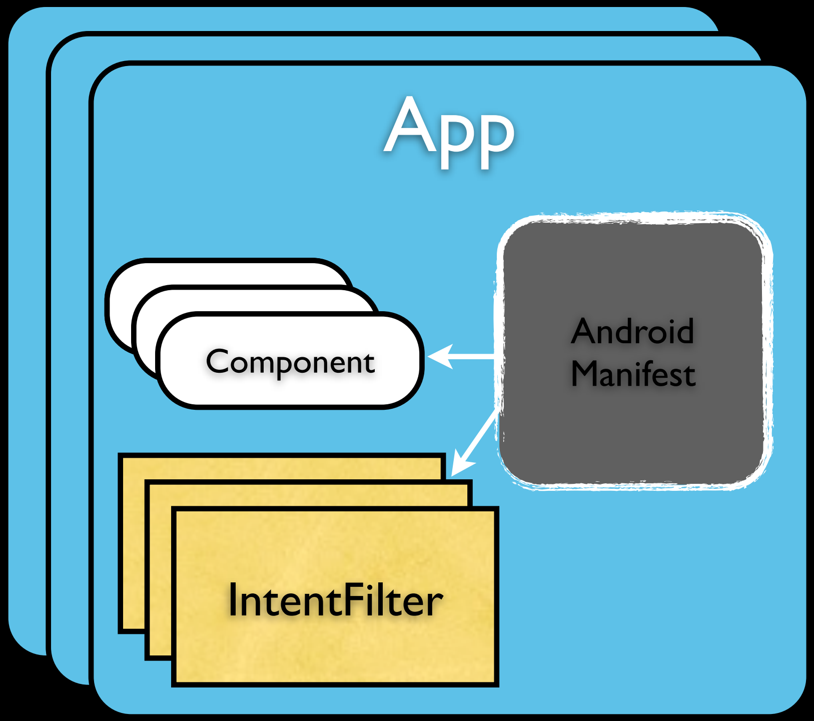
```
<manifest>
  <application>
    <activity>
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>
</manifest>
```

IntentFilter

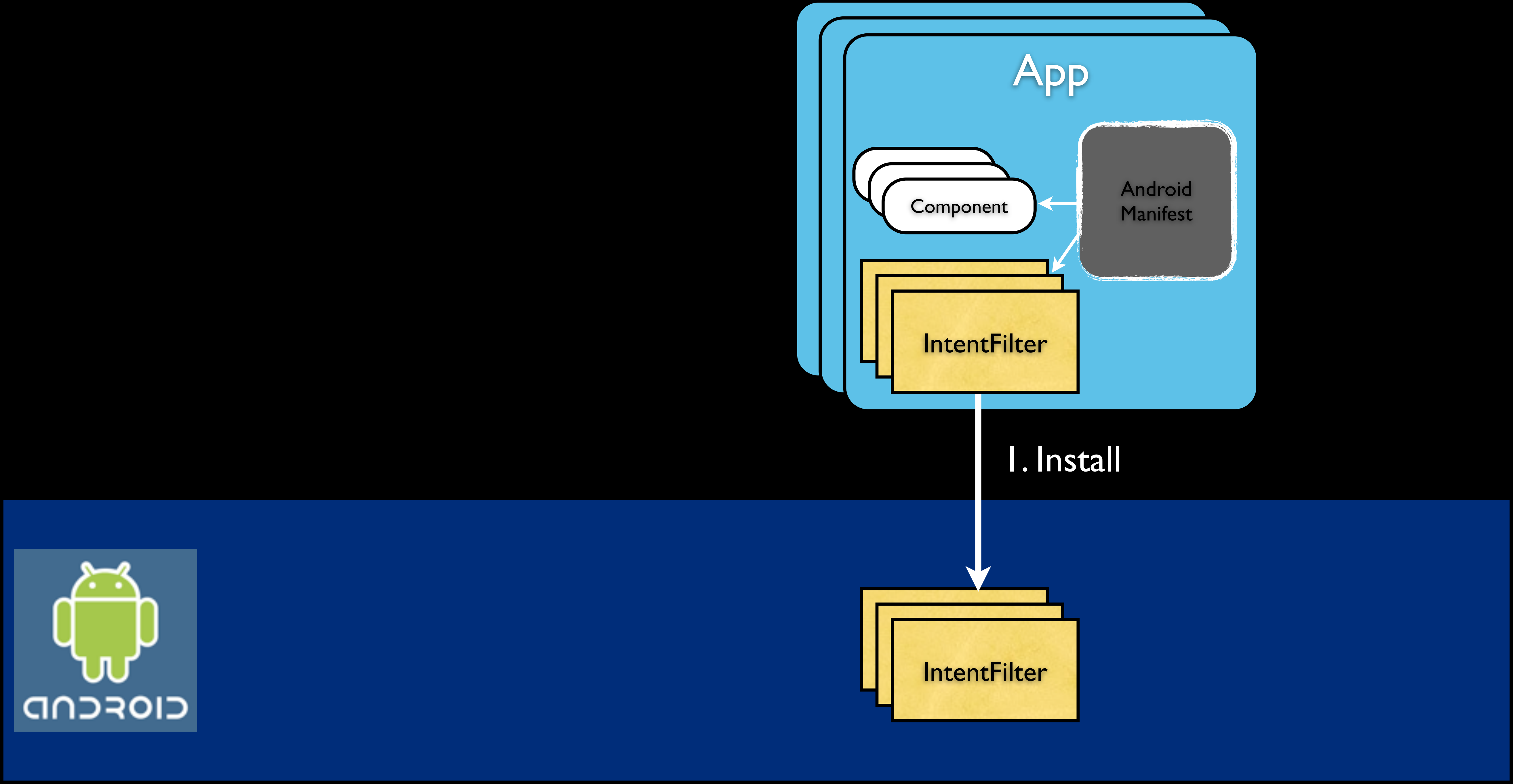
```
 <activity android:name="NotesList" android:label="@string/title_notes_list">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <action android:name="android.intent.action.EDIT" />
        <action android:name="android.intent.action.PICK" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="vnd.android.cursor.dir/vnd.google.note" />
    </intent-filter>
    <intent-filter>
        <action android:name="android.intent.action.GET_CONTENT" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="vnd.android.cursor.item/vnd.google.note" />
    </intent-filter>
</activity>

<activity android:name="NoteEditor"
          android:theme="@android:style/Theme.Light"
          android:label="@string/title_note" >
    <intent-filter android:label="@string/resolve_edit">
        <action android:name="android.intent.action.VIEW" />
        <action android:name="android.intent.action.EDIT" />
        <action android:name="com.android.notepad.action.EDIT_NOTE" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="vnd.android.cursor.item/vnd.google.note" />
    </intent-filter>
    <intent-filter>
        <action android:name="android.intent.action.INSERT" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="vnd.android.cursor.dir/vnd.google.note" />
    </intent-filter>
</activity>
```

Intent Overview

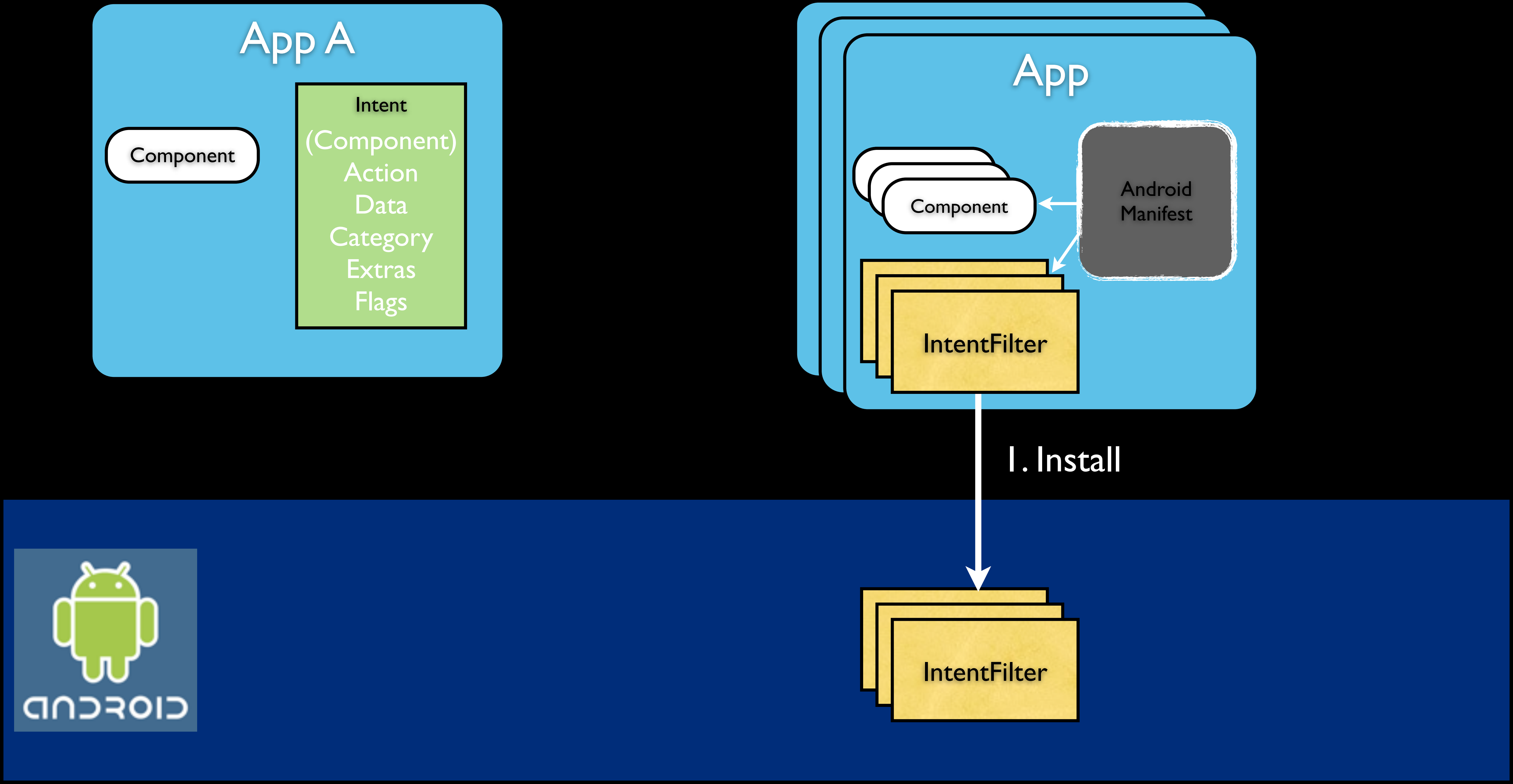
Intent Overview



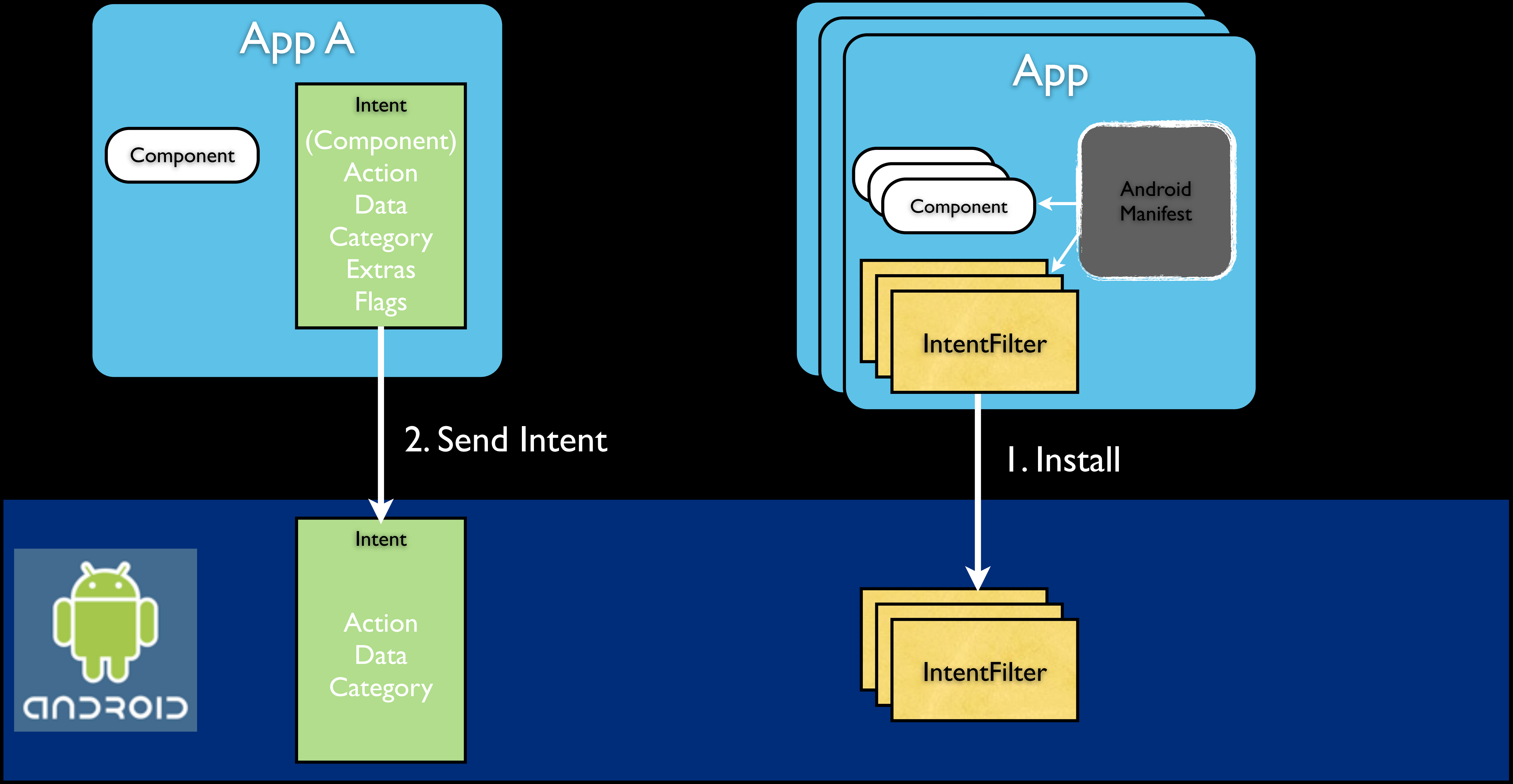
Intent Overview



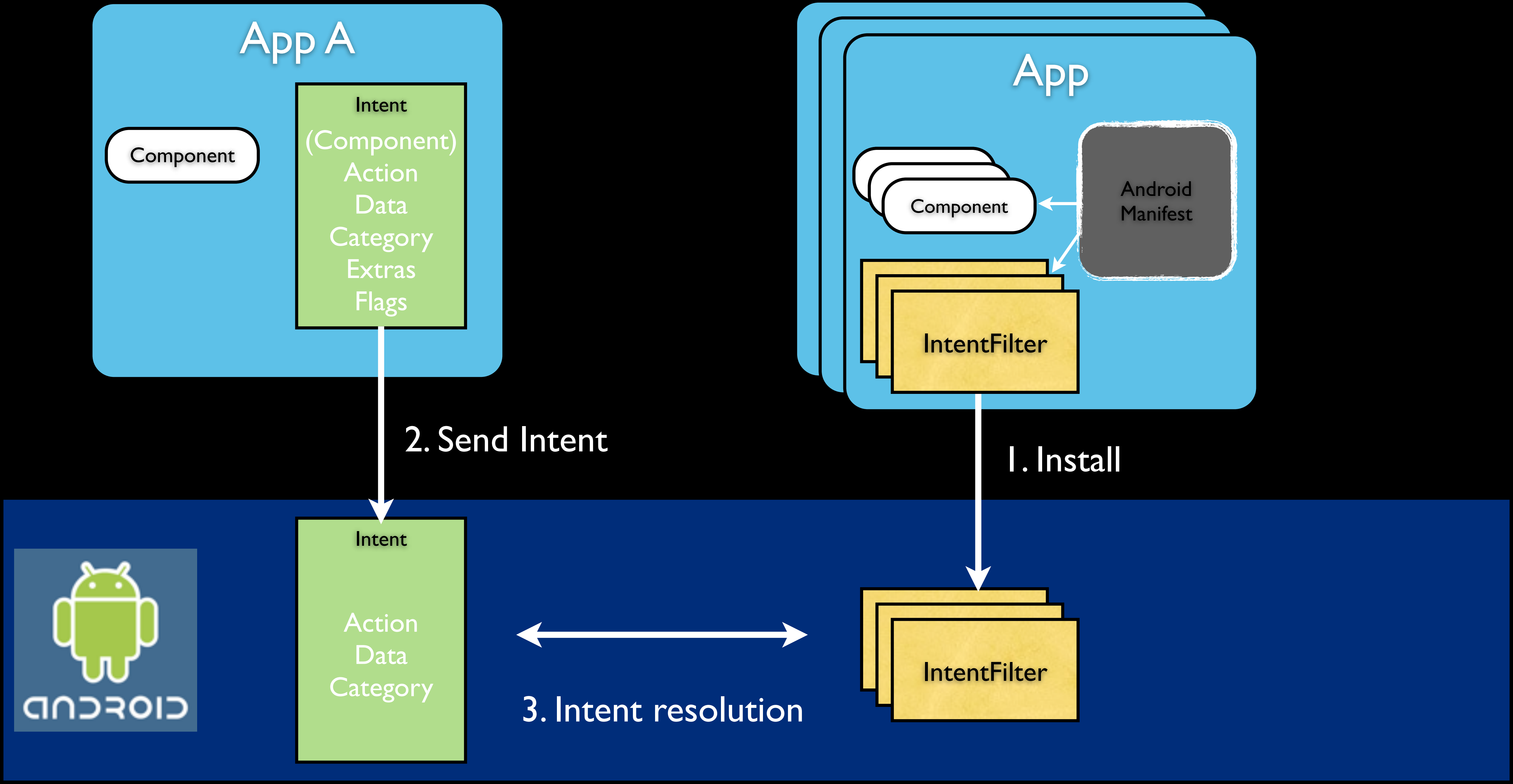
Intent Overview



Intent Overview

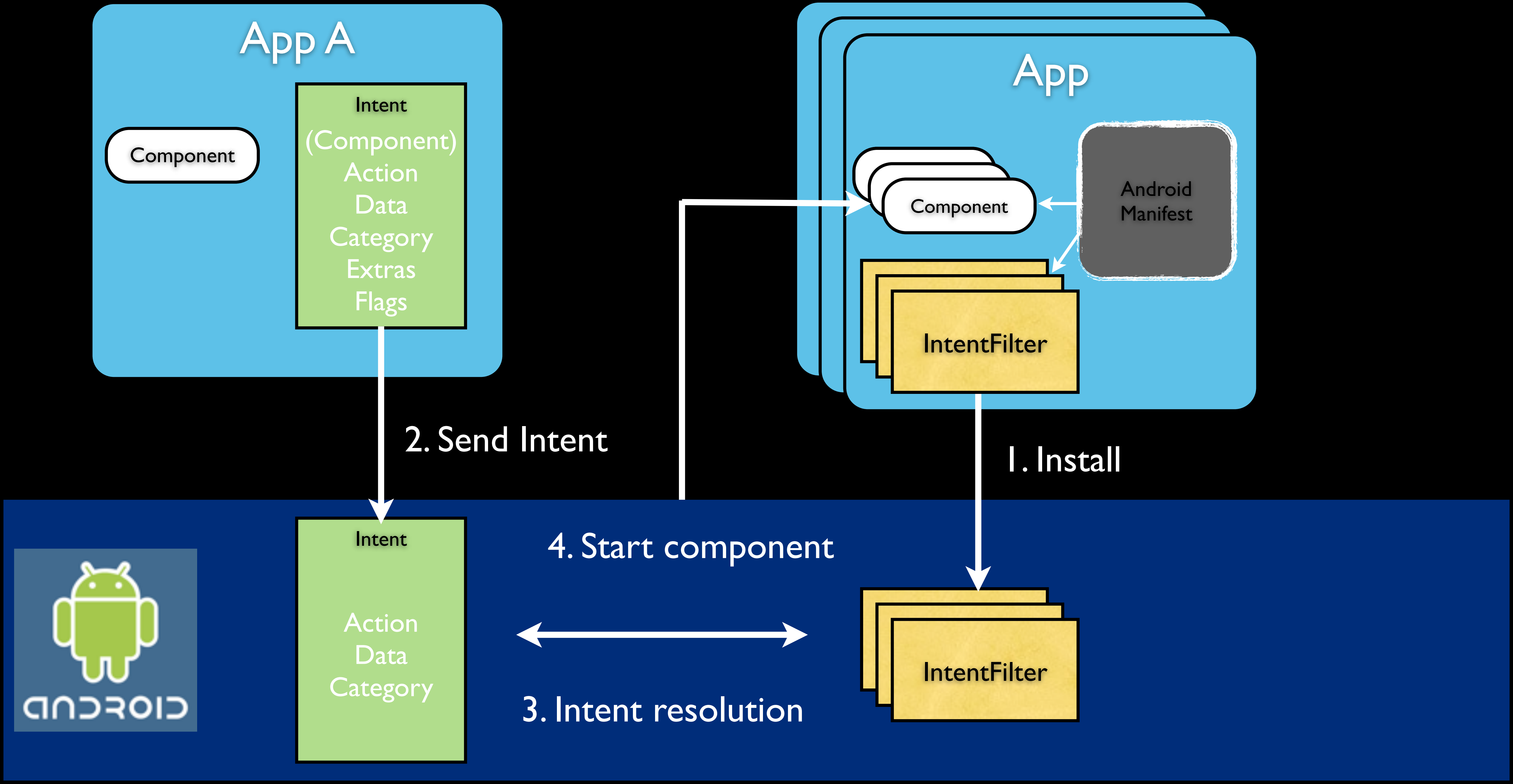


Intent Overview

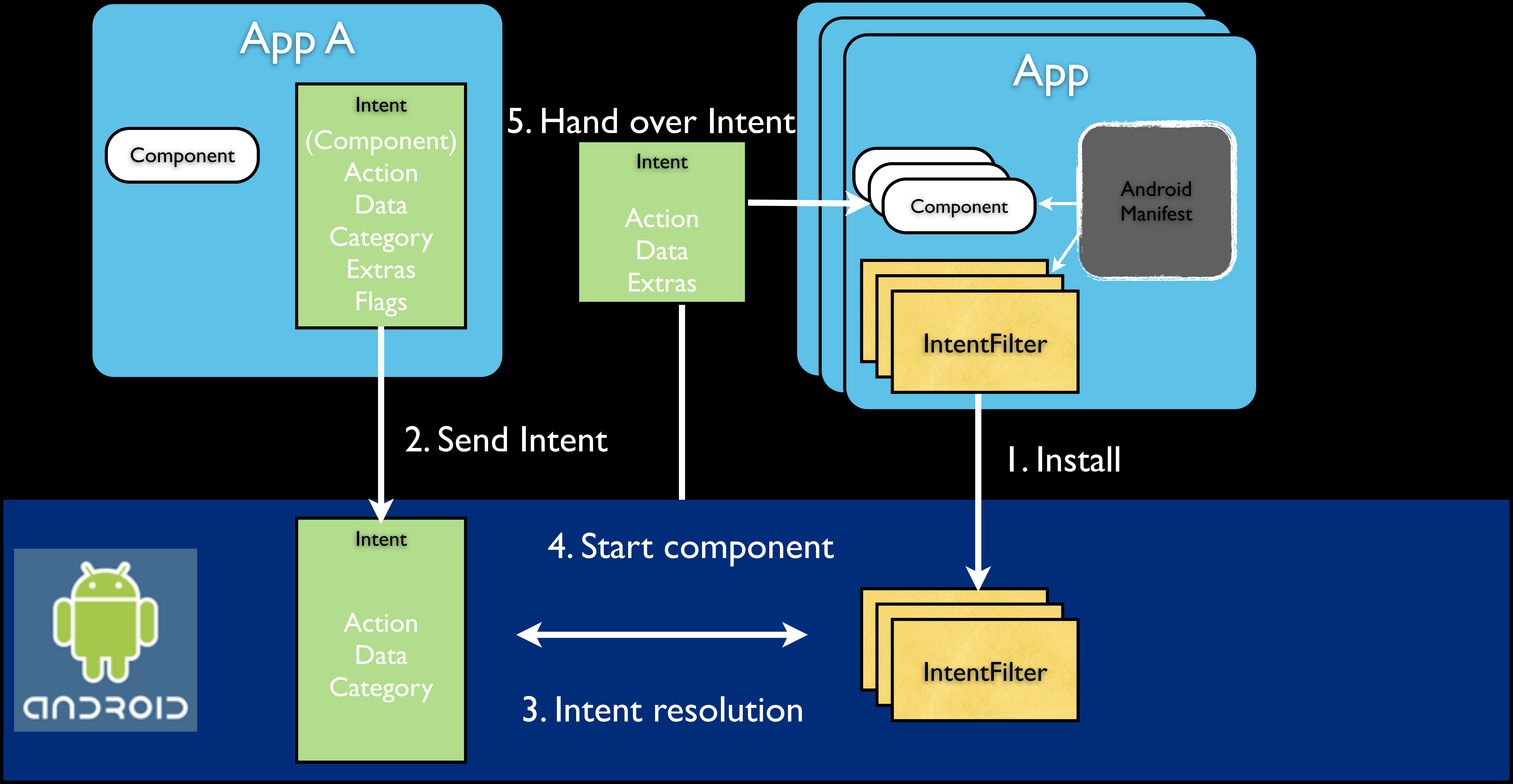


JAYWAY

Intent Overview



Intent Overview



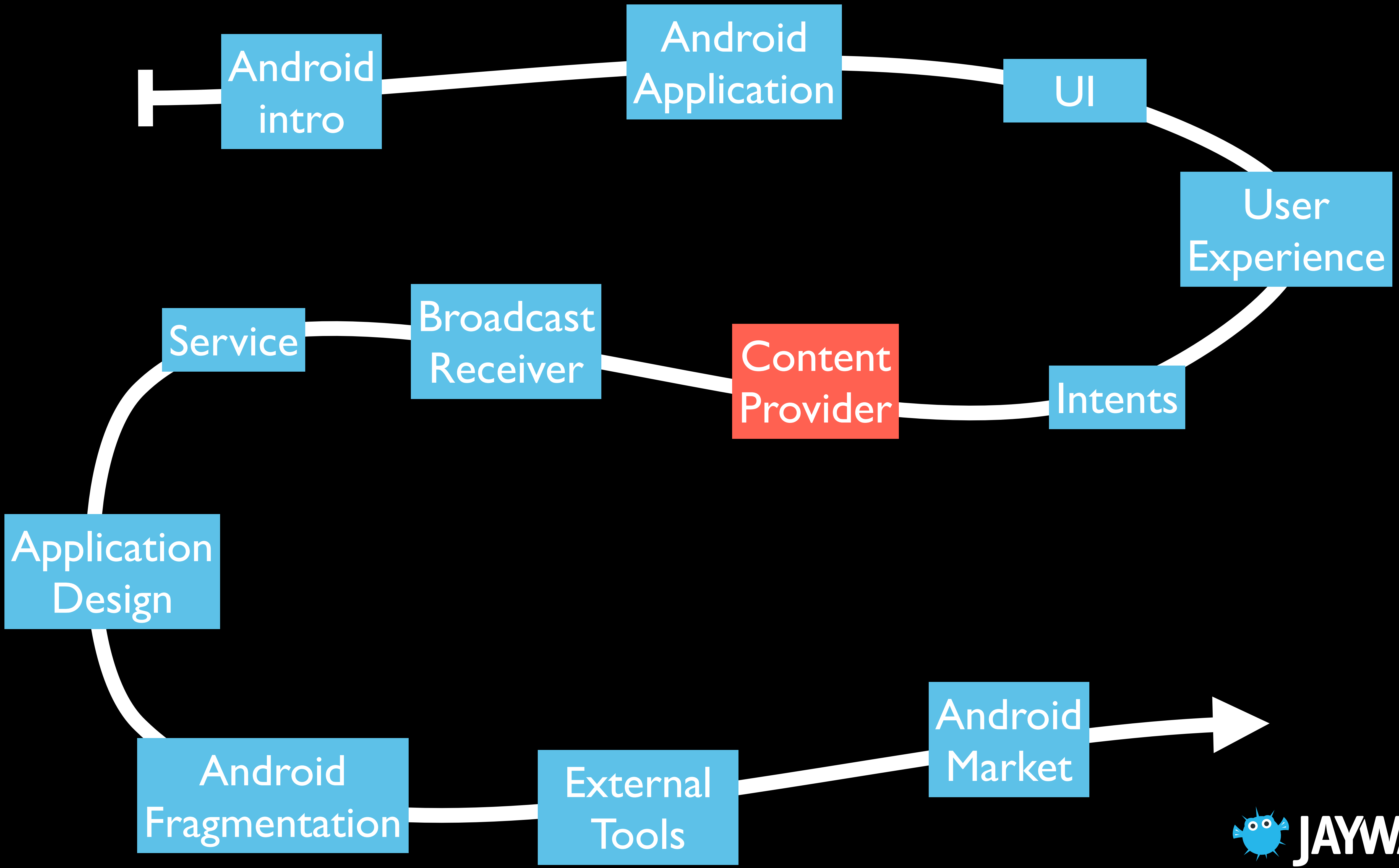
Intent Resolution

- ACTION test (Filter: 1..*)
    - Pass: 1 IntentFilter-action matches Intent-action
- CATEGORY test (Filter: 1..*)
    - Pass: All Intent-categories matches IntentFilter-categories
- DATA test (Filter: 0..*)
    - Uri + MIME-type
    - Pass: Intent-data is subset of IntentFilter-data

Best Practices

- Only reusable Activities shall specify IntentFilter
- Check if intent can be handled
    - Pre-check
        - PackageManager.queryIntentActivities()
    - Post-check
        - Catch exception from startActivity()

Agenda



JAYWAY

Content Provider - Data Storage

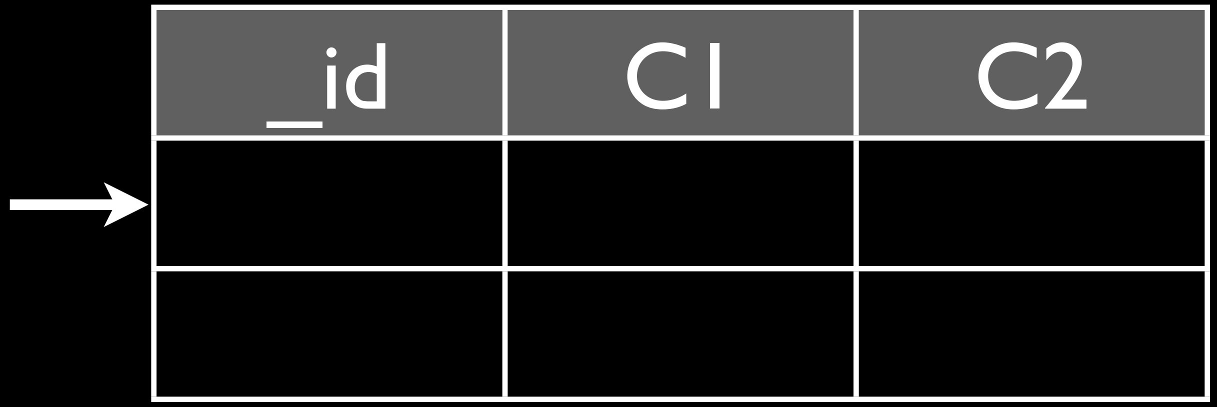
- File system
  - Internal storage & SD Card
- SharedPreferences
  - Key/Value pairs
- SQLite DB

SQLite

- Relational database
- Self-contained
- Transactional
- No need for separate server process
- Stored as a file
  - `/data/data/com.yourdomain.yourAppPackage/databases`

Android SQLite API

- SQLiteDataBase
- SQLiteOpenHelper
- ContentValues
- Cursor
  - Container for a query result

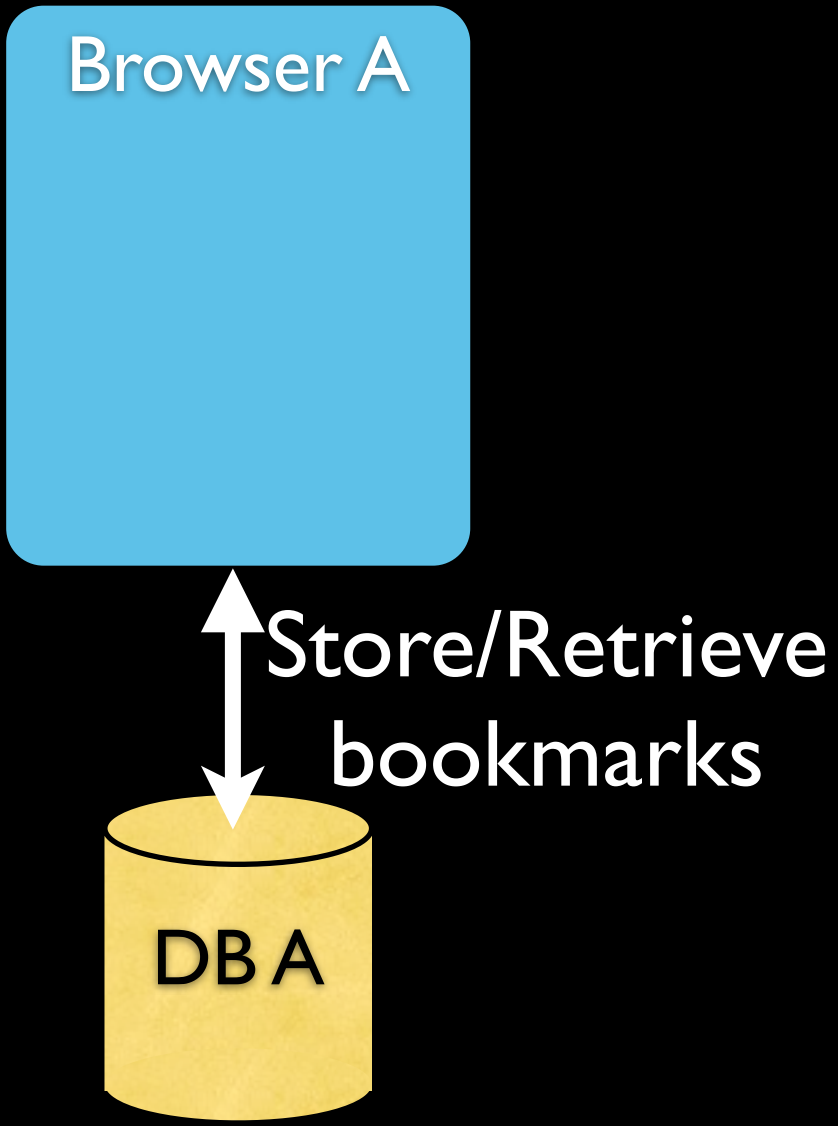


```
Cursor cur = db.query("tbl_countries",
     null, null, null, null, null, null);
cur.moveToFirst();
while (cur.isAfterLast() == false) {
     //Do something
     cur.moveToNext();
}
cur.close();
```

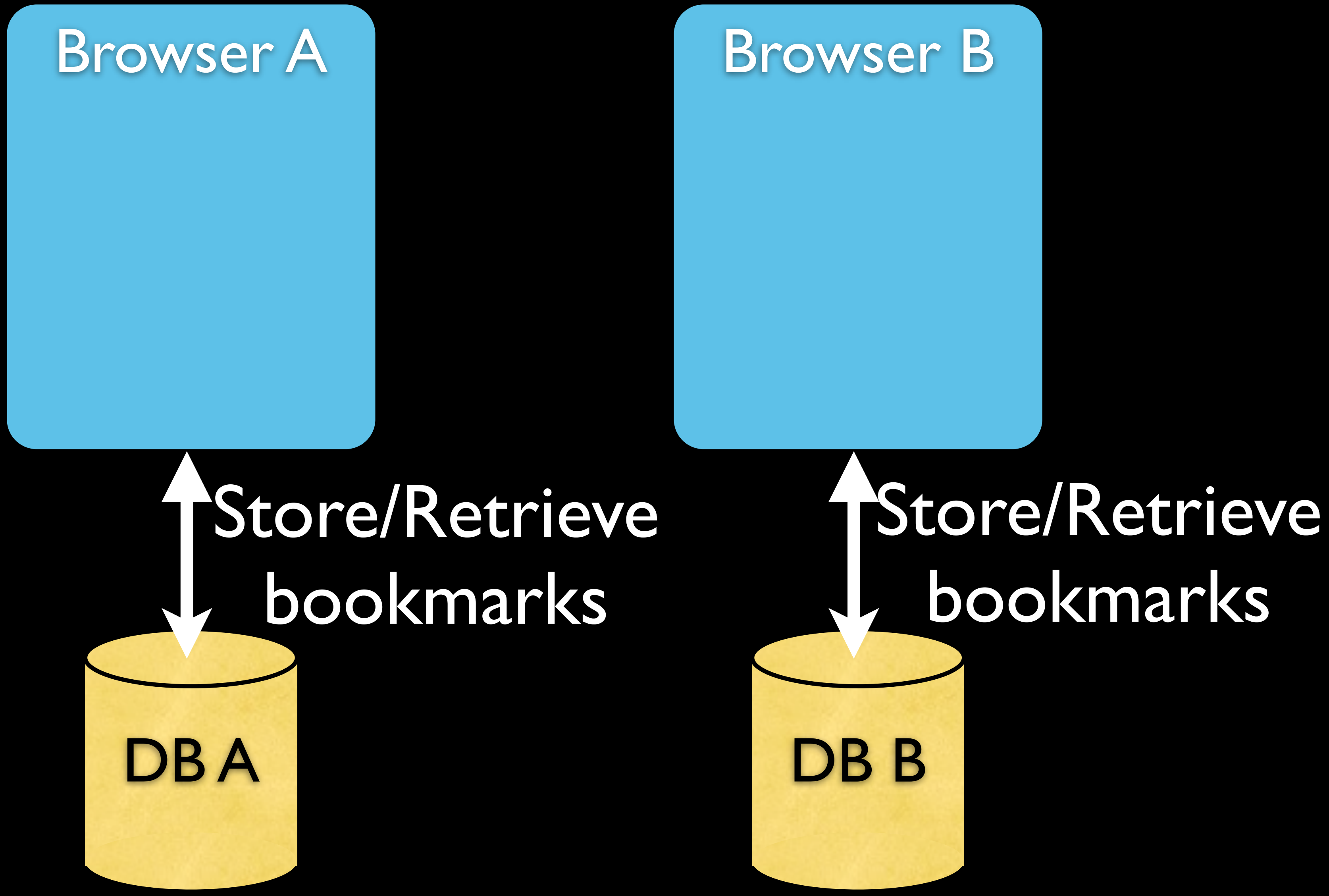
ContentProvider

- Generic data handling mechanism for any content type
  - Store application data
  - Retrieve application data
  - Share data between applications

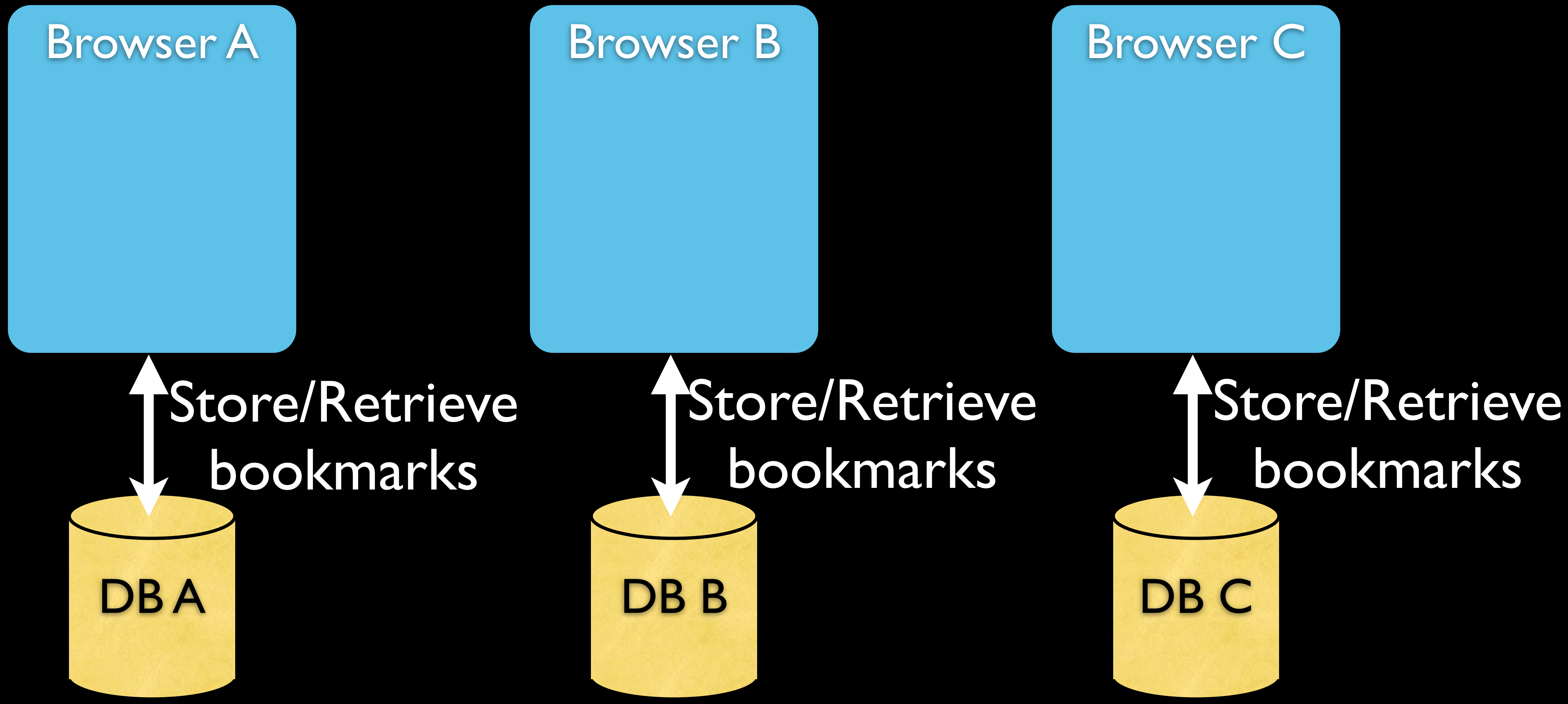
Motivation - Bookmark example



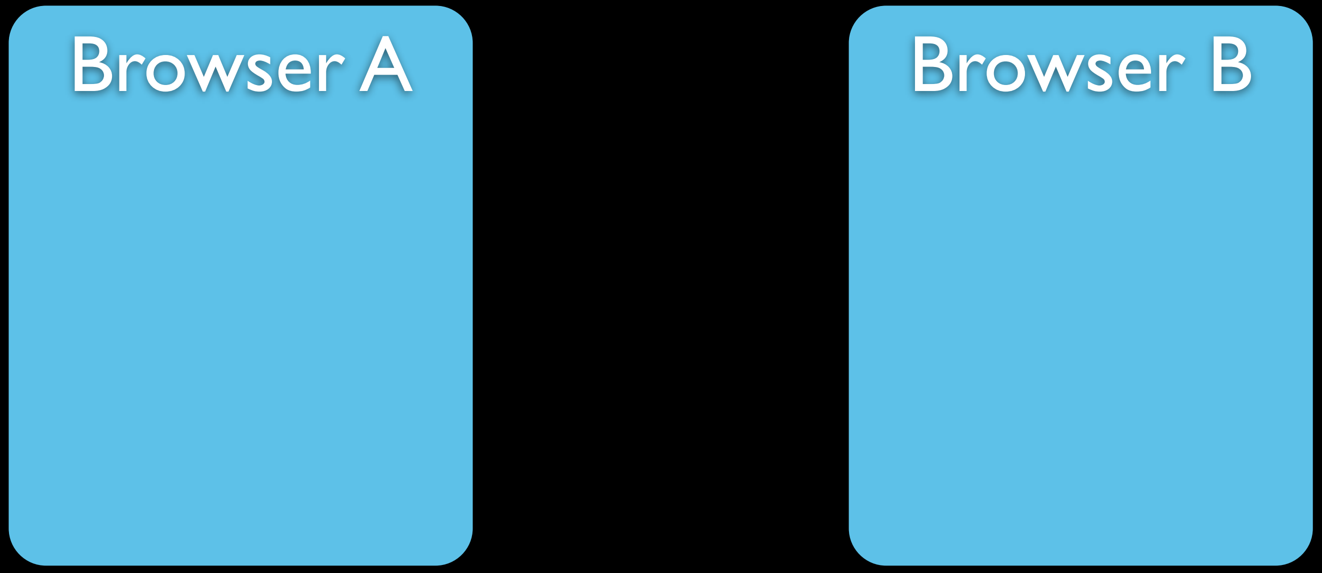
Motivation - Bookmark example



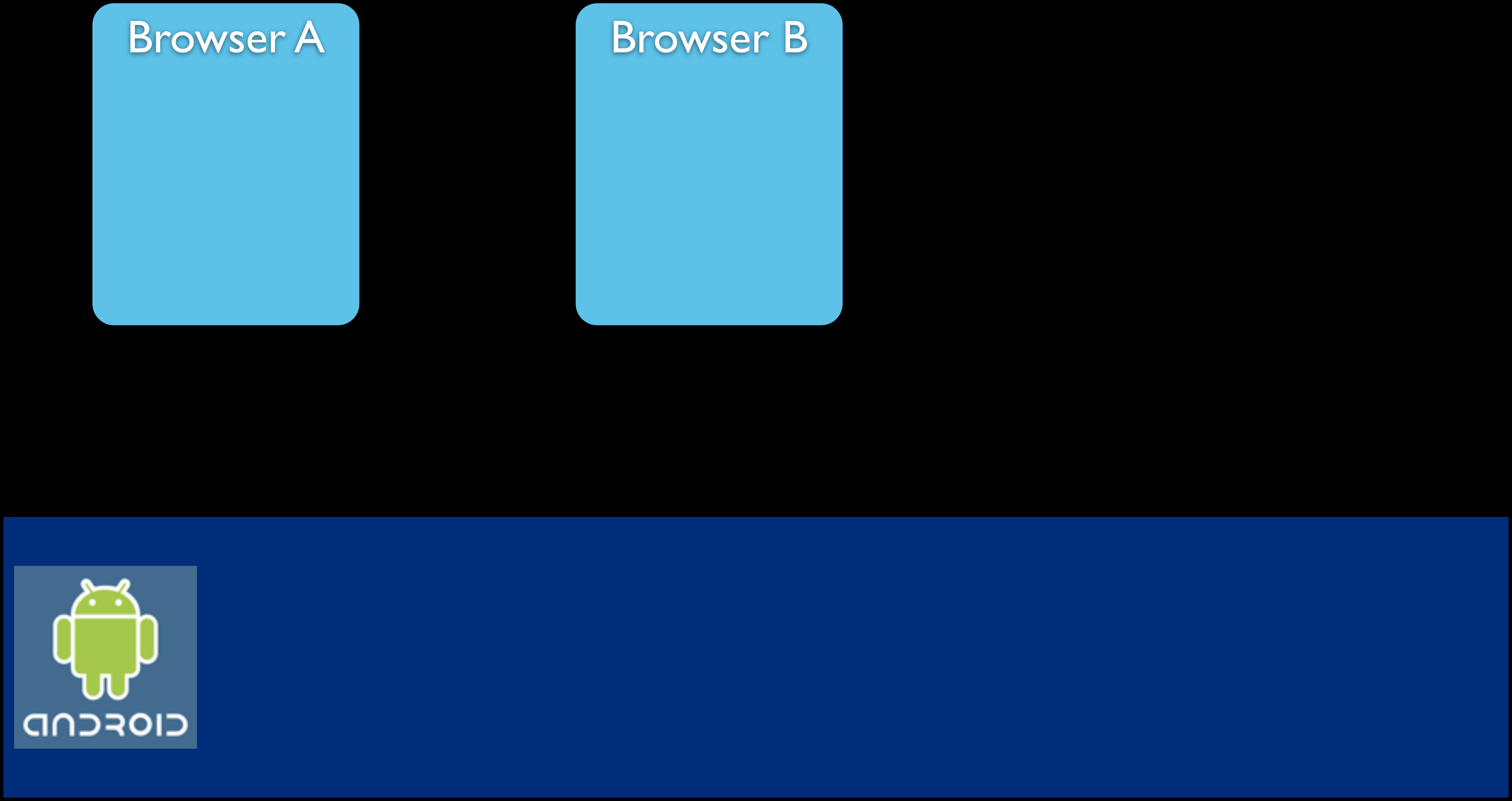
Motivation - Bookmark example



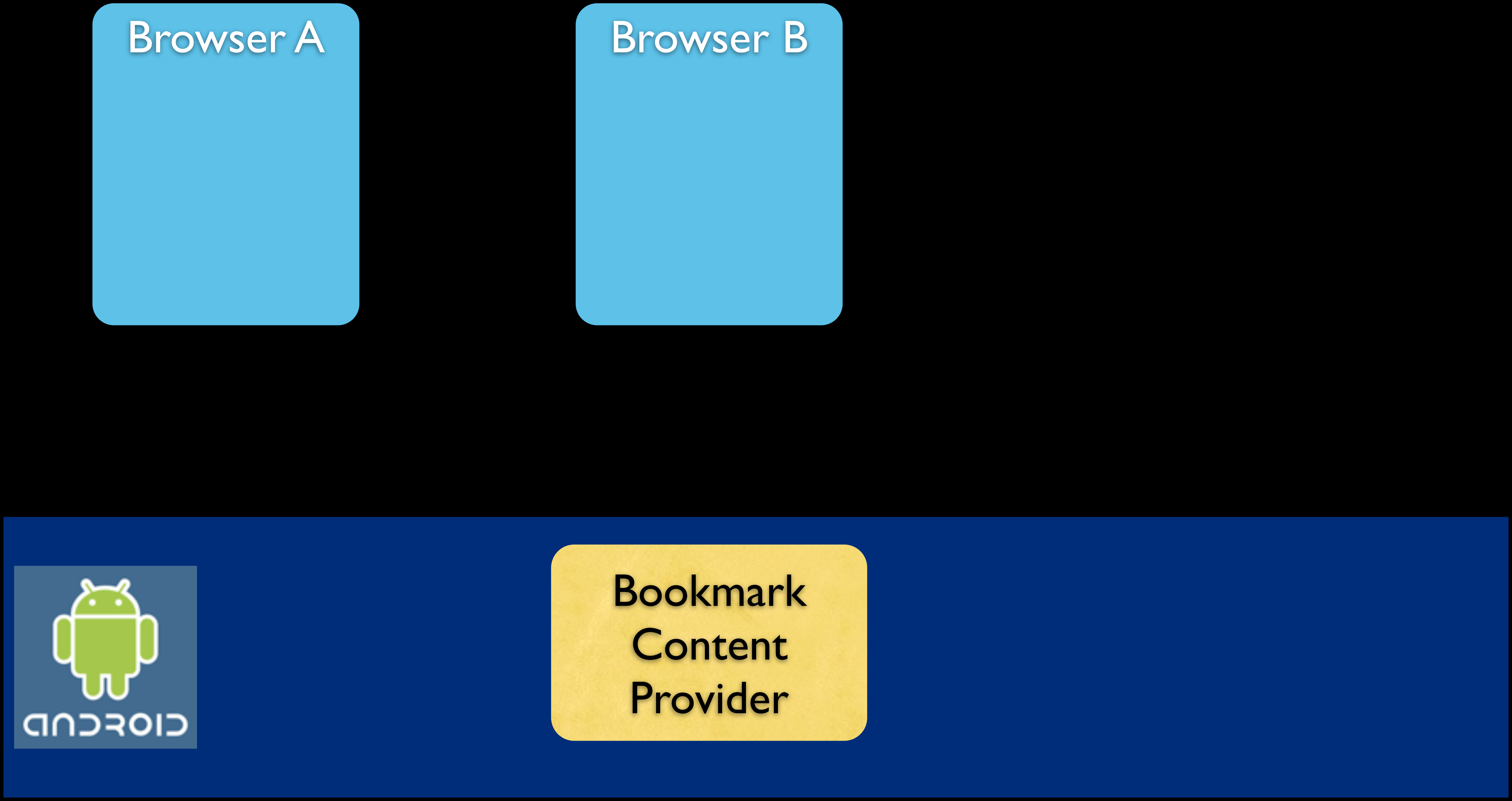
Motivation - Bookmark example



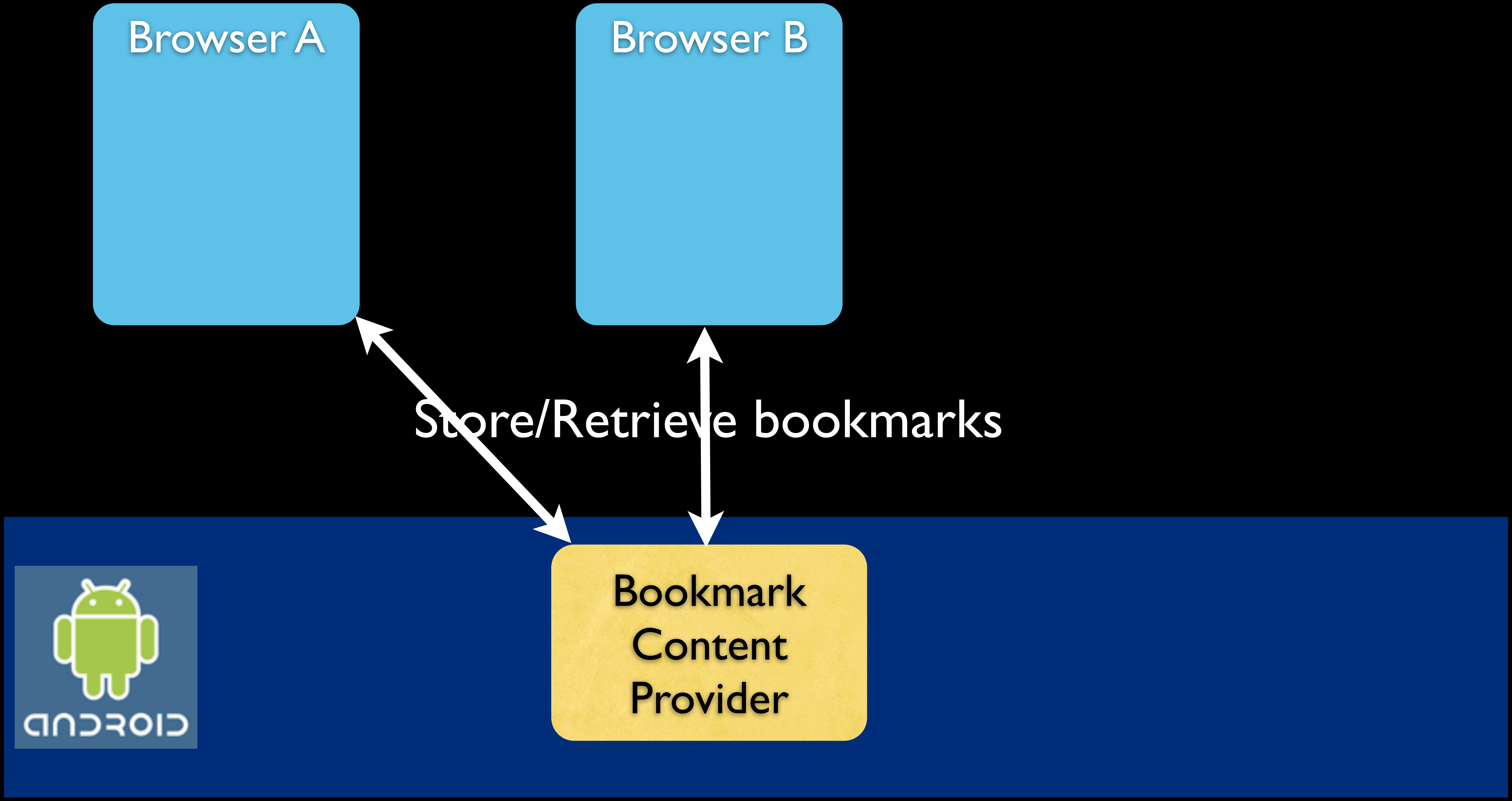
Motivation - Bookmark example



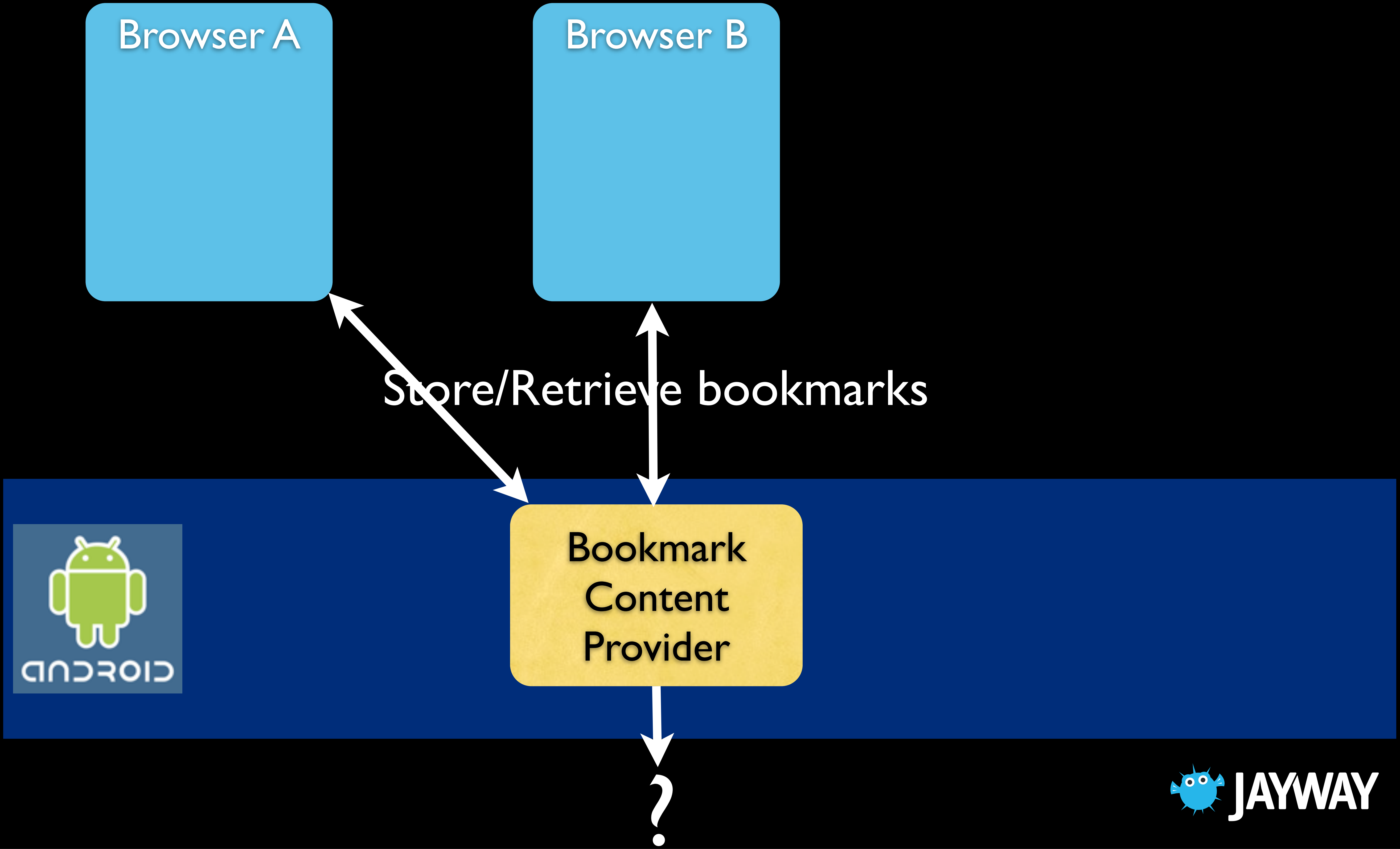
Motivation - Bookmark example



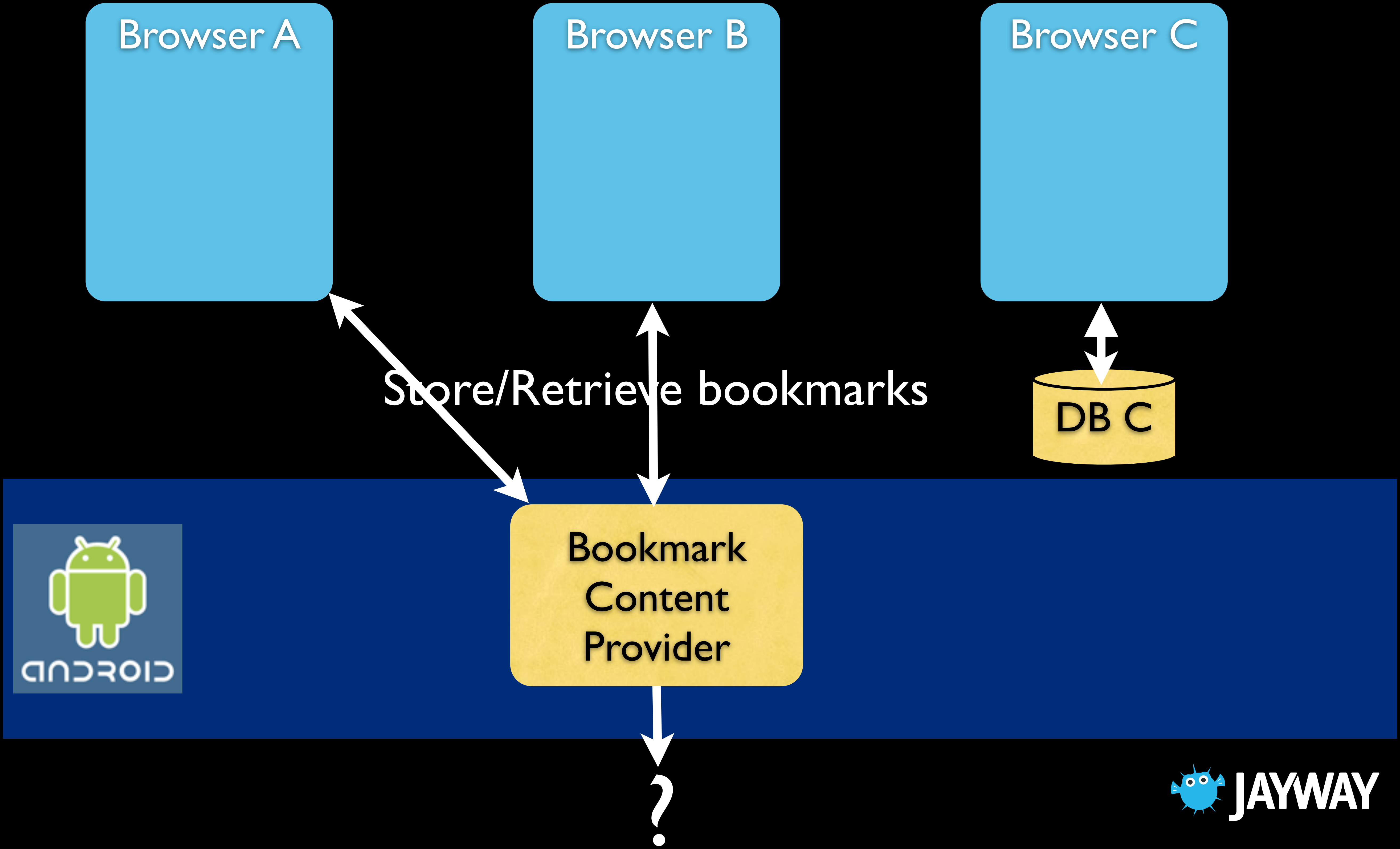
Motivation - Bookmark example



Motivation - Bookmark example

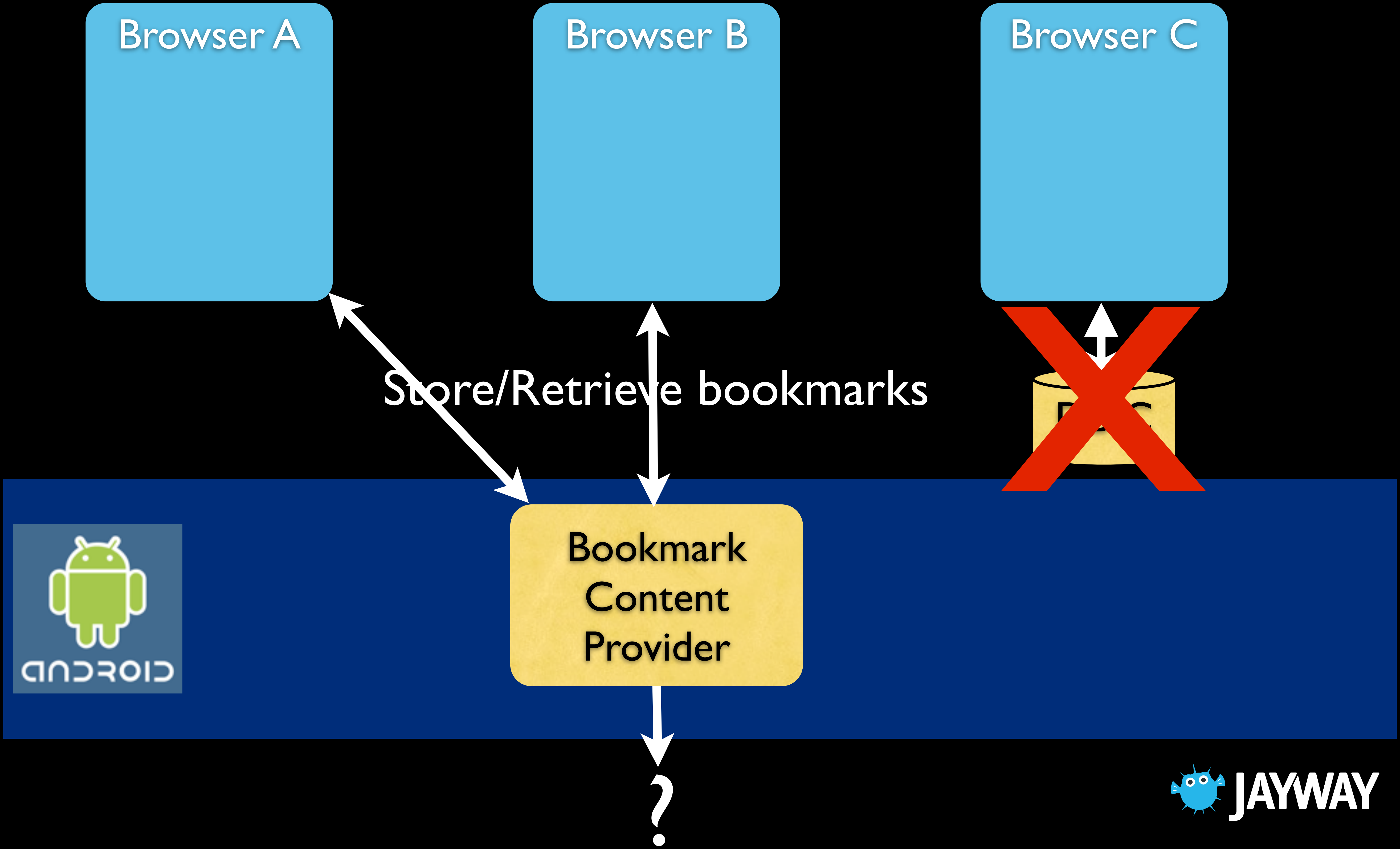


Motivation - Bookmark example

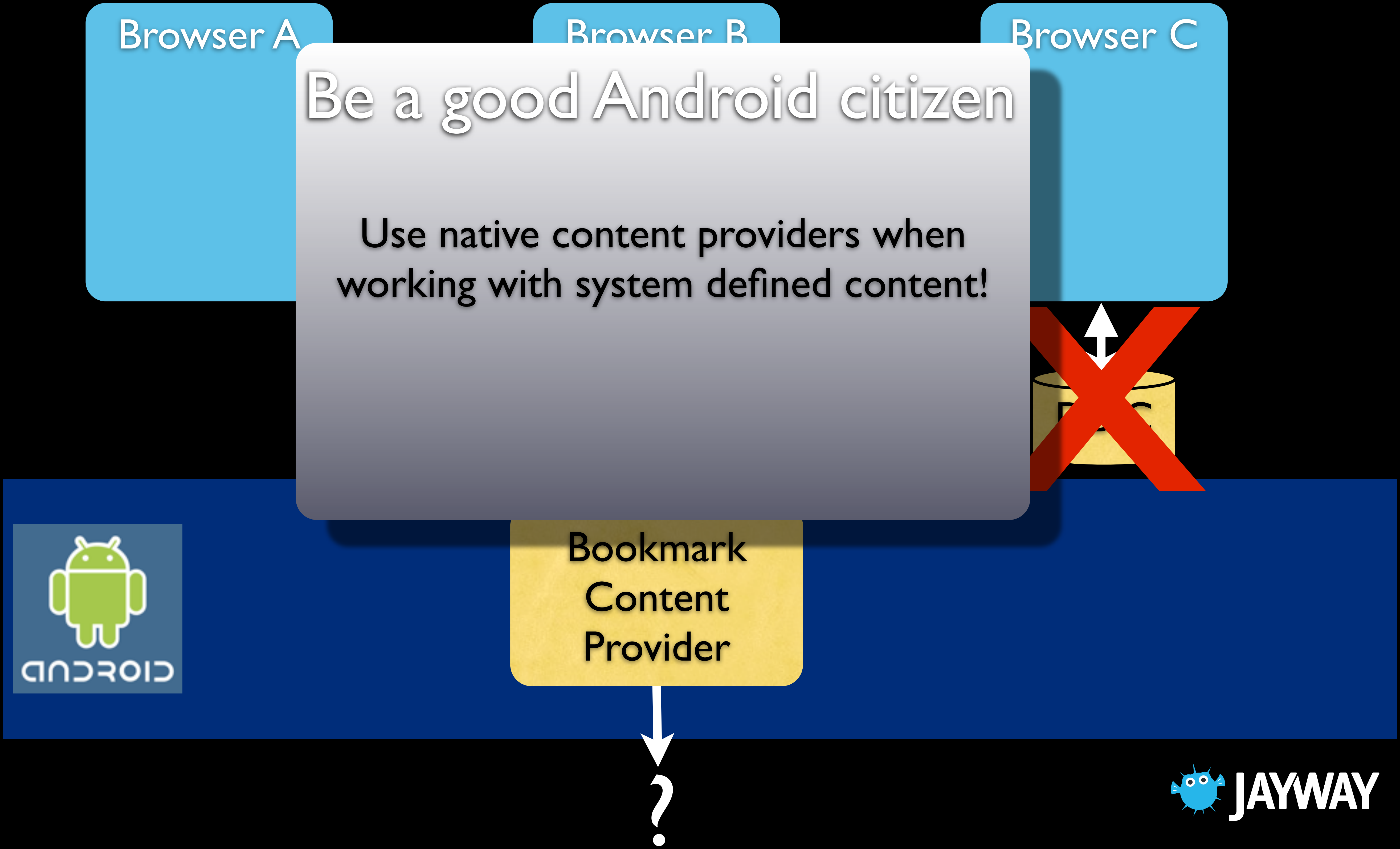


JAYWAY

Motivation - Bookmark example



Motivation - Bookmark example



Be a good Android citizen

Use native content providers when working with system defined content!

Native Content

- android.provider.*
  - Bookmarks
  - CallLog
  - Contacts
  - Dictionary
  - Media
    - Audio, Images, Video
  - Settings

- Device domain model

JAYWAY

Fundamentals

- Content Provider is based on DB concept, but abstracts storage mechanism
- Database storage the normal use case
    - ContentProviders exposes data outside the application
    - ContentProviders expose a full CRUD interface
        - Create, Retrieve, Update, Delete

Fundamentals

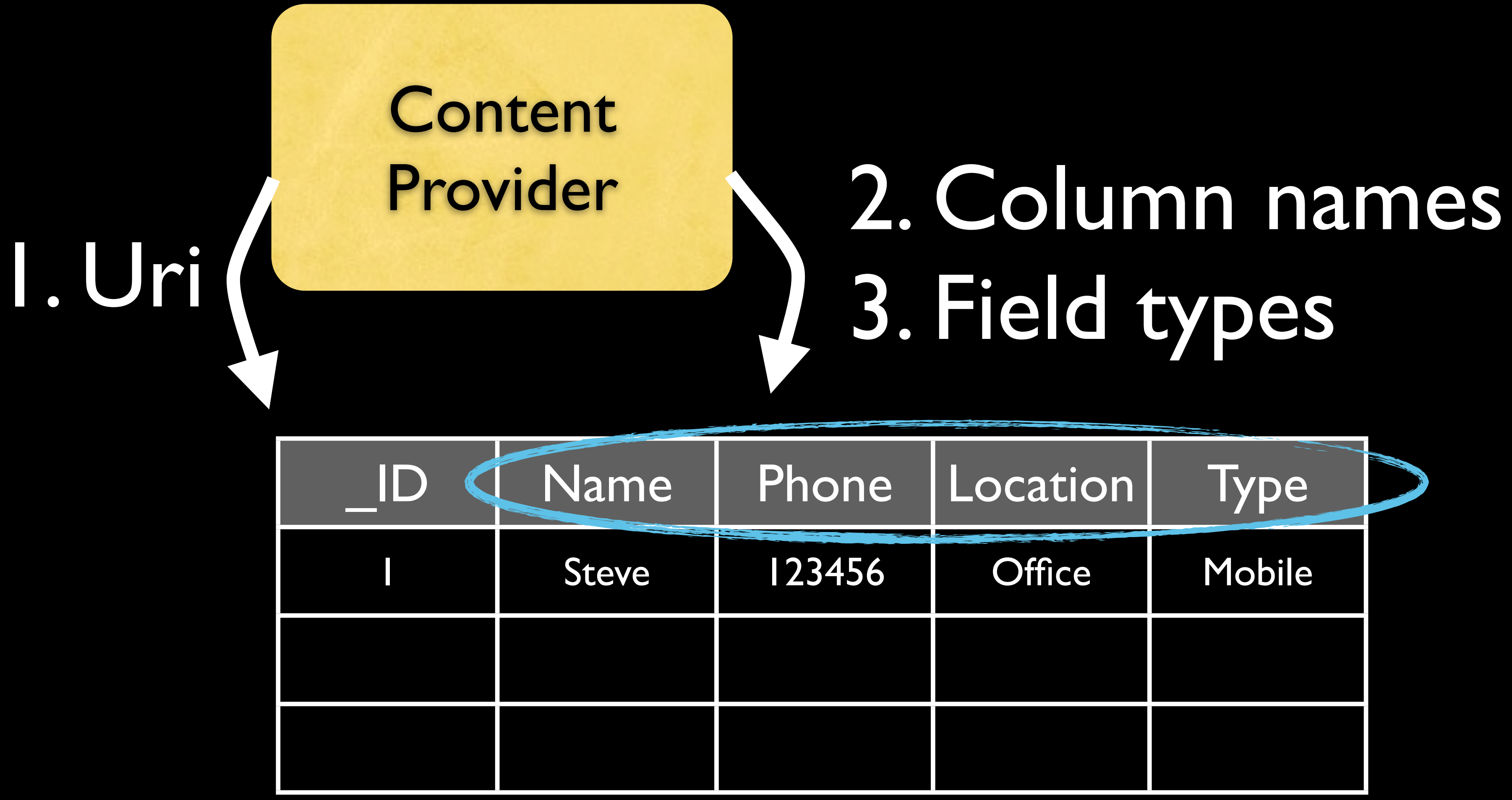
- Data model = Database table

All ContentProviders,
Unique,
(cmp Primary Key)

_ID				

Fundamentals

- External interface
  - Uri, Column name, Field type, Mime type



_ID	Name	Phone	Location	Type
1	Steve	123456	Office	Mobile

Fundamentals

- Native ContentProviders expose URI and Column names as constants, e.g.:
  - android.provider.Settings.System.CONTENT_URI
  - android.provider.ContactsContract.Contacts.CONTENT_URI
- Native ContentProviders may require permissions, e.g.:
  - android.permission.READ_CONTACTS
  - android.permission.WRITE_SETTINGS

Querying for Content

- Clients interact with a ContentProvider through a ContentResolver

```
ContentResolver cr = getContentResolver();
```

- Queries are similar to database queries

```
Cursor cursor = cr.query(Uri uri, String[] columns, String where, String
[] selectionArgs, String sortOrder)
```

Manipulating Content

- ContentValues used for insert and update
  - key/value pairs
- Insert

```
Uri uri = insert(Uri uri, ContentValues values)
```

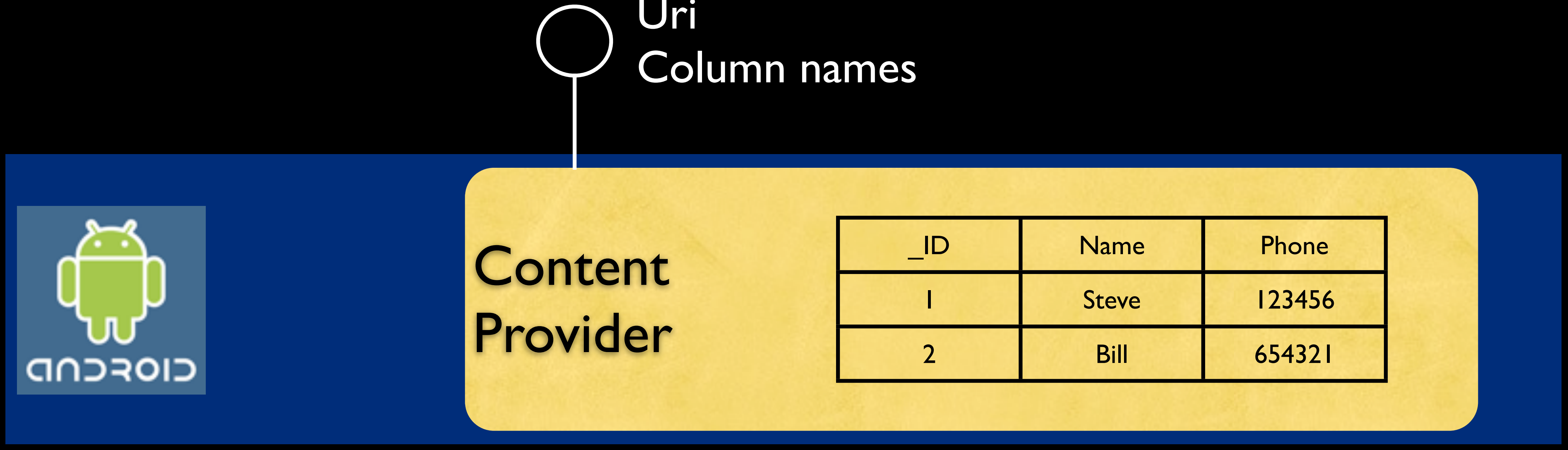
- Update

```
int rows = update(Uri uri, ContentValues values, String where, String[]
selectionArgs)
```

- Delete

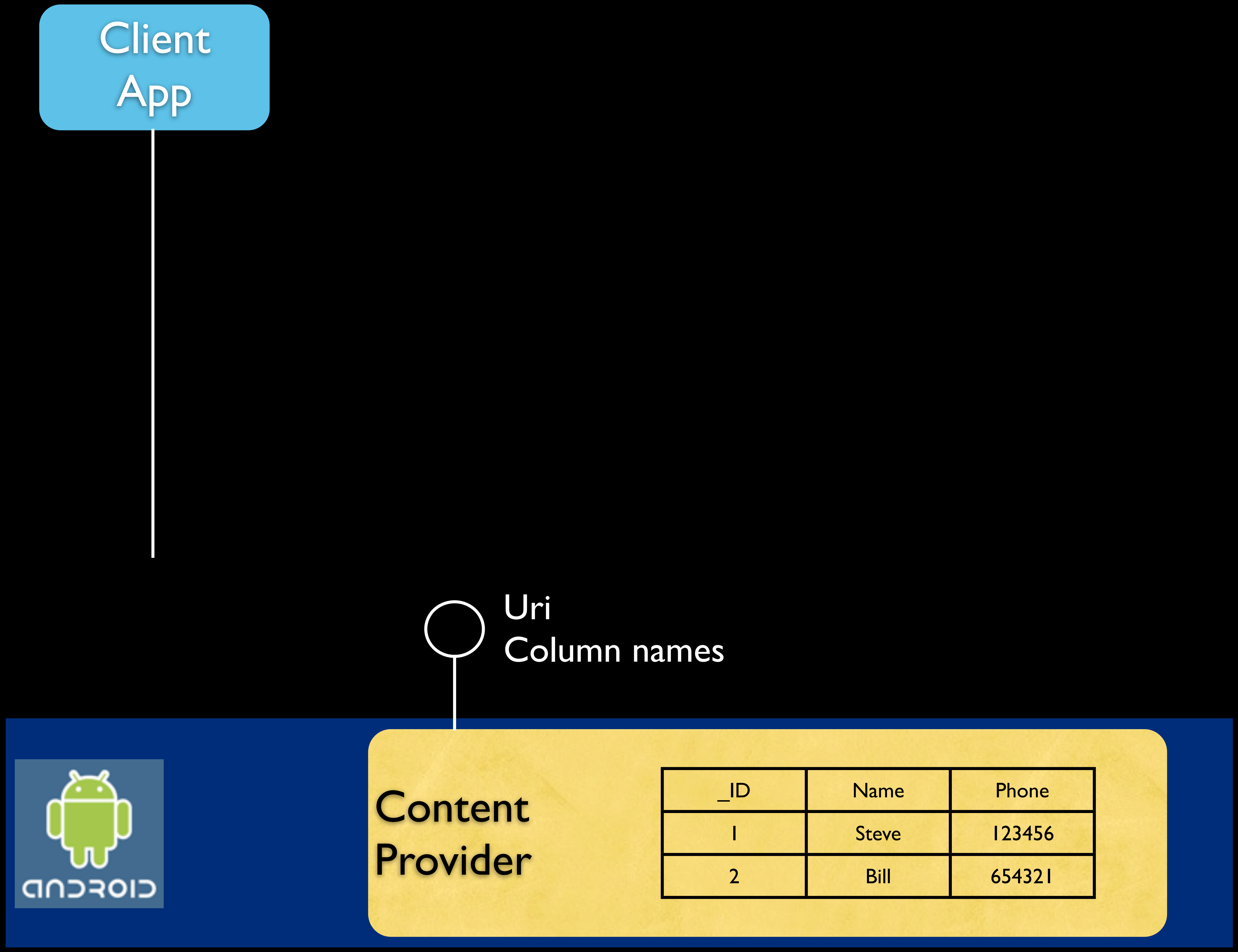
```
int rows = delete(Uri url, String where, String[] selectionArgs)
```

Cursor



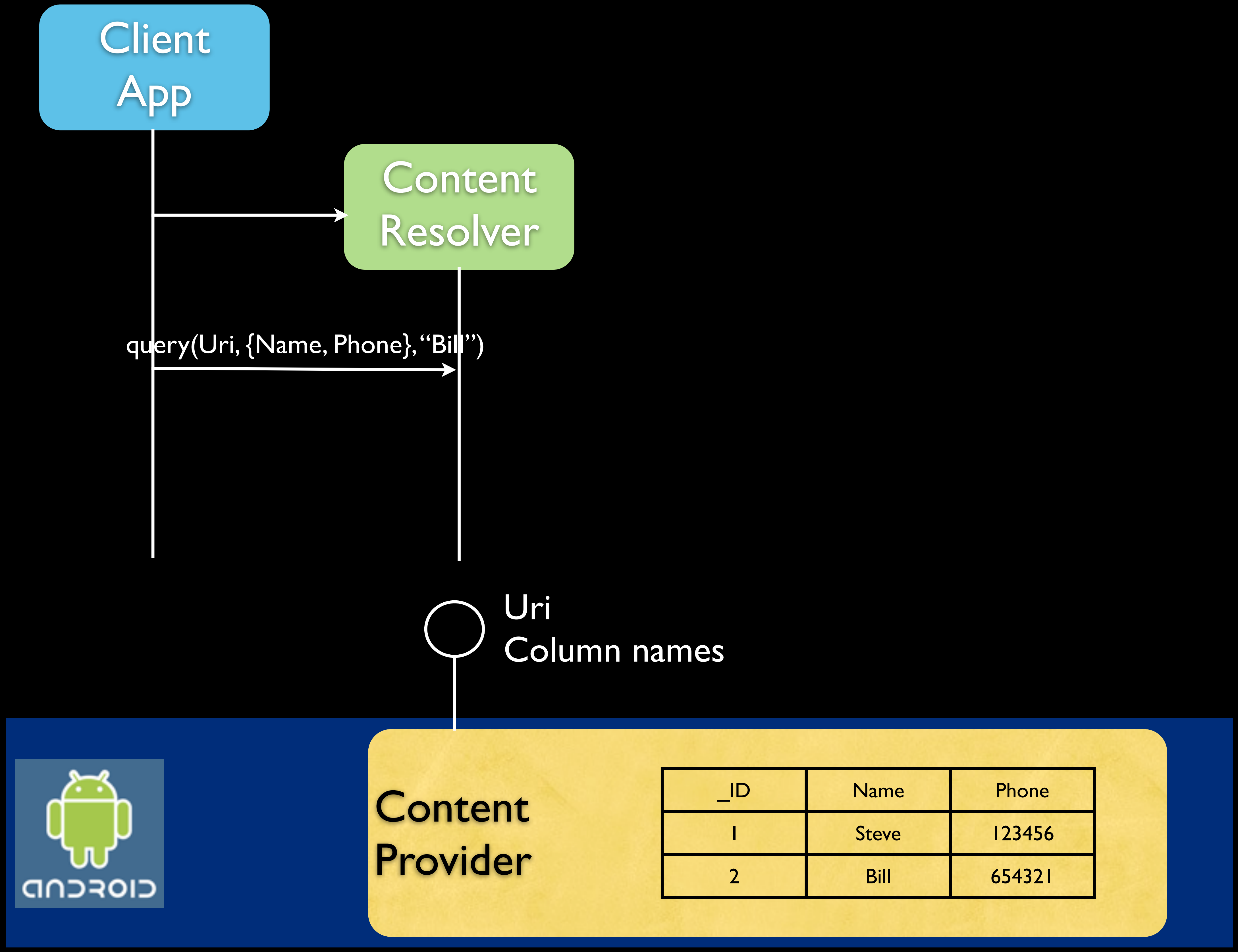
_ID	Name	Phone
1	Steve	123456
2	Bill	654321

Cursor



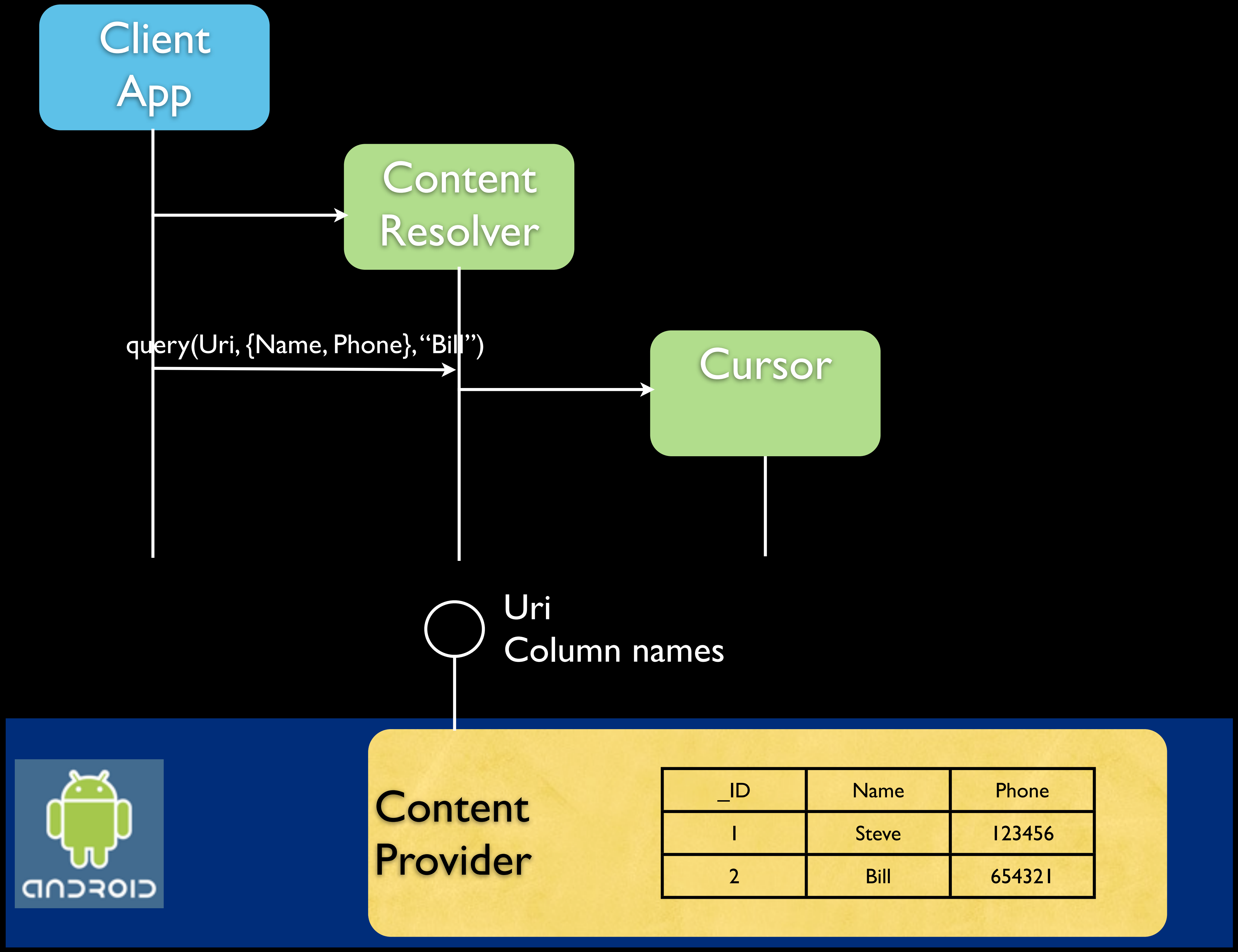
_ID	Name	Phone
1	Steve	123456
2	Bill	654321

Cursor



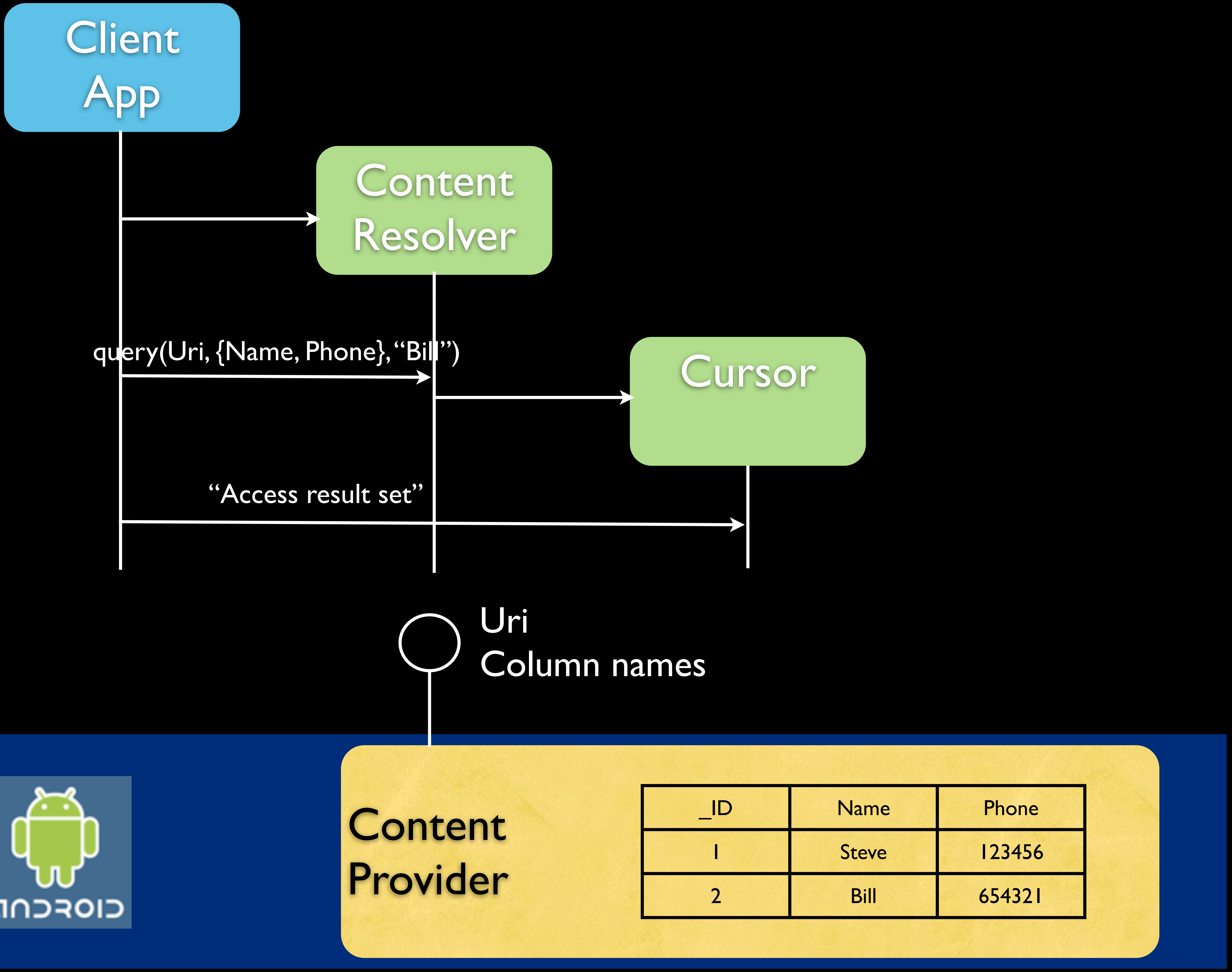
_ID	Name	Phone
1	Steve	123456
2	Bill	654321

Cursor



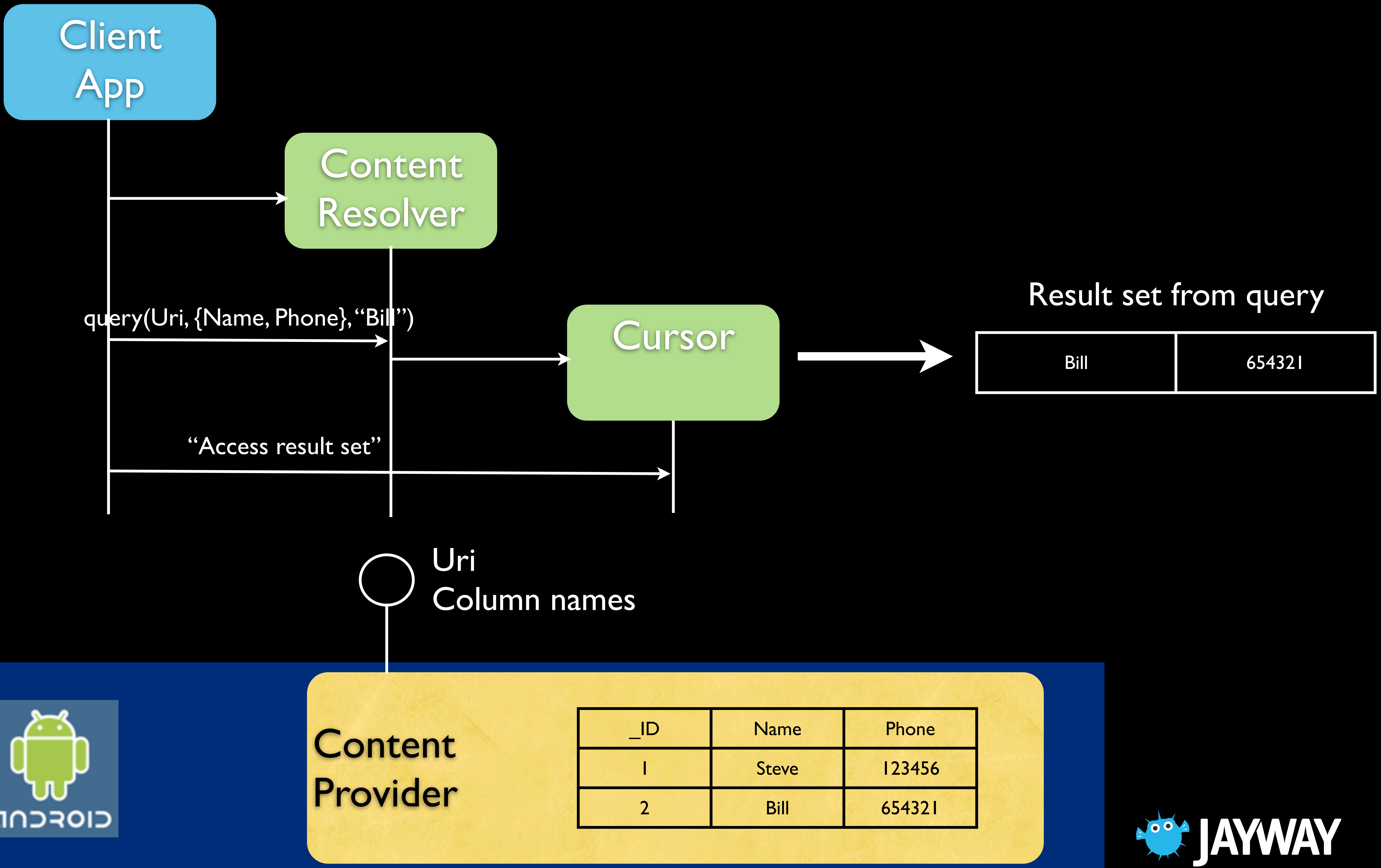
_ID	Name	Phone
1	Steve	123456
2	Bill	654321

Cursor



_ID	Name	Phone
1	Steve	123456
2	Bill	654321

Cursor



Bill	654321

_ID	Name	Phone
1	Steve	123456
2	Bill	654321

Cursor

- Access result set from query
- Connection to the result set that shall be closed
- Cursor management can be handled by the platform

```
Cursor cursor = managedQuery(Uri uri, String[] projection, String
selection, String[] selectionArgs, String sortOrder)
```

- Handles Cursor lifecycle
  - Unloads at onPause()
  - Requery at restart

JAYWAY

Cursor

- Acces
- Conne ed
- Curso latform

```
Cursor c n, String
selectio
```

- Har
  - Unloads at onPause()
  - Requery at restart

Utilize the platform

If a Cursor is accessed from an Activity always let the platform manage the Cursor for you.

JAYWAY

Expose Data - Step 1

- Decide on storage mechanism
  - File system
  - Network
  - Database

Expose Data - Step 2

- Extend the ContentProvider class

`onCreate()` - Initialize the provider on application start

`query(Uri, String[], String, String[], String)` - Return Cursor-object

`insert(Uri, ContentValues)` - Insert data

`update(Uri, ContentValues, String, String[])` - Update data

`delete(Uri, String, String[])` - Delete data

`getType(Uri)` - Return the data MIME type

Expose Data - Step 2

- Extend the ContentProvider class

`onCreate()` - Initialize the provider on application start ← UI Thread

---

`query(Uri, String[], String, String[], String)` - Return Cursor-object

`insert(Uri, ContentValues)` - Insert data

`update(Uri, ContentValues, String, String[])` - Update data

`delete(Uri, String, String[])` - Delete data

`getType(Uri)` - Return the data MIME type

JAYWAY

Expose Data - Step 2

- Extend the ContentProvider class

`onCreate()` - Initialize the provider on application start ← UI Thread

---

`query(Uri, String[], String, String[], String)` - Return Cursor-object

`insert(Uri, ContentValues)` - Insert data ← Thread safe

`update(Uri, ContentValues, String, String[])` - Update data

`delete(Uri, String, String[])` - Delete data

`getType(Uri)` - Return the data MIME type

Expose Data - Step 3

- Declare the ContentProvider

```
<manifest>

    <application>

        <provider android:name="com.jayway.provider.MyProvider"
          android:authorities="com.jayway.provider.myprovider"
          ...
        </provider>

    </application>

</manifest>
```

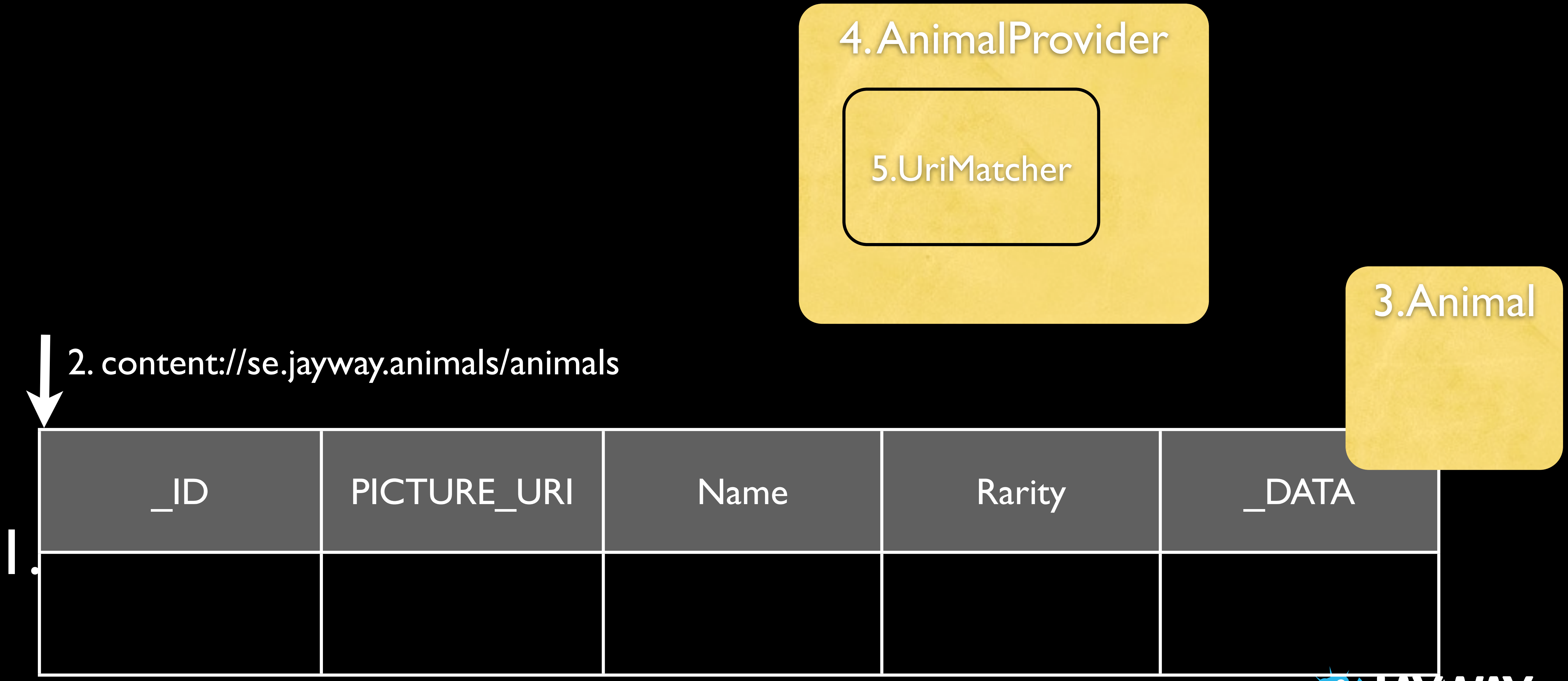
JAYWAY

Expose Data - Step 4

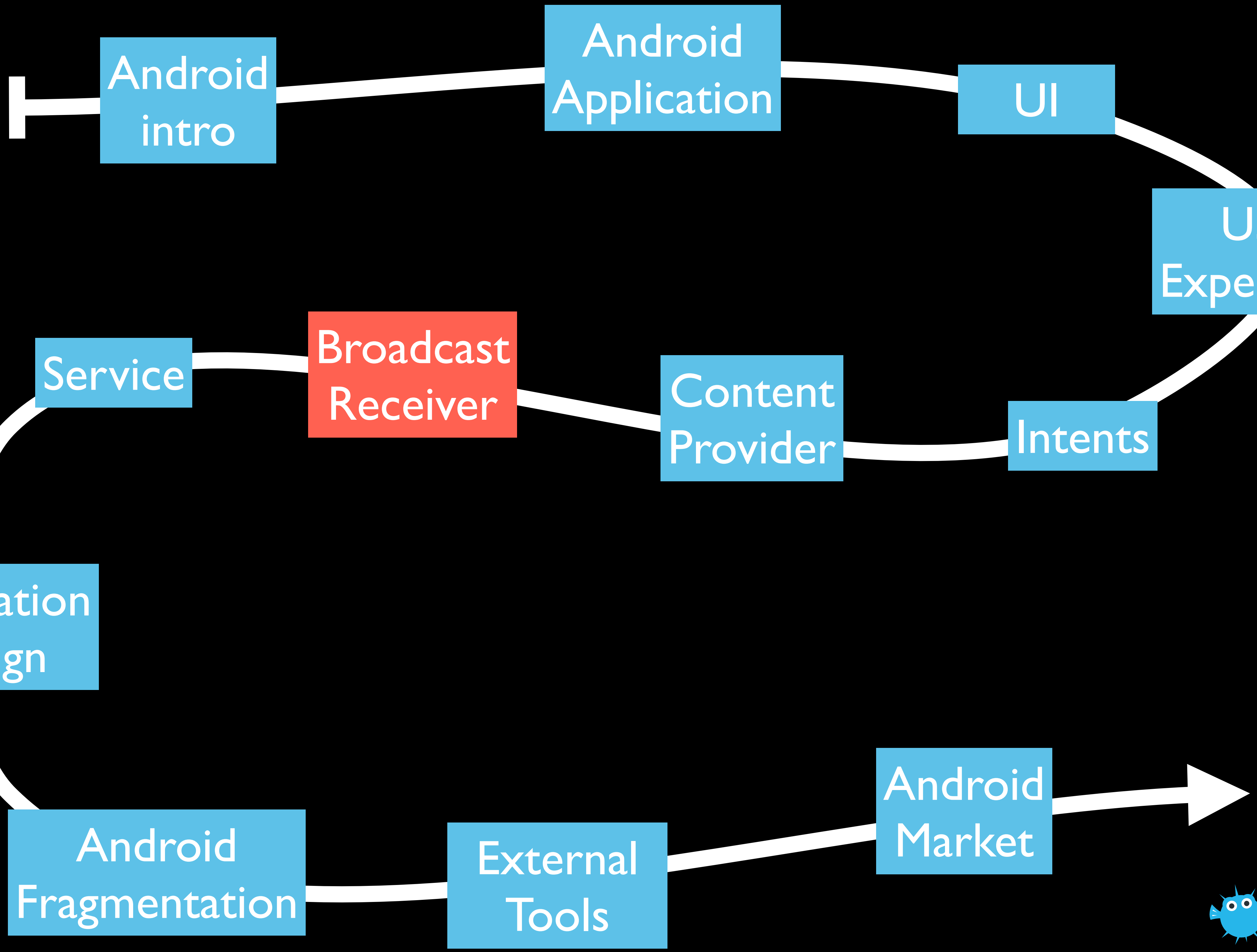
- Create ContentProvider definition according to Android standard
  - CONTENT_URI
  - Column names
  - Mime types
    - `vnd.android.cursor.item/vnd.`*`yourcompanyname.contenttype`*
    - `vnd.android.cursor.dir/vnd.`*`yourcompanyname.contenttype`*

Expose Data - Example

1. Storage: DB
2. CONTENT_URI
3. ContentProvider Definition
4. ContentProvider
5. UriMatcher



Agenda



JAYWAY

BroadcastReceiver

- Receive broadcasted events
    - System and Apps
- Subclass
    - onReceive()
- Registration
    - Static
    - Dynamic
        - Context.registerReceiver()

BroadcastReceiver

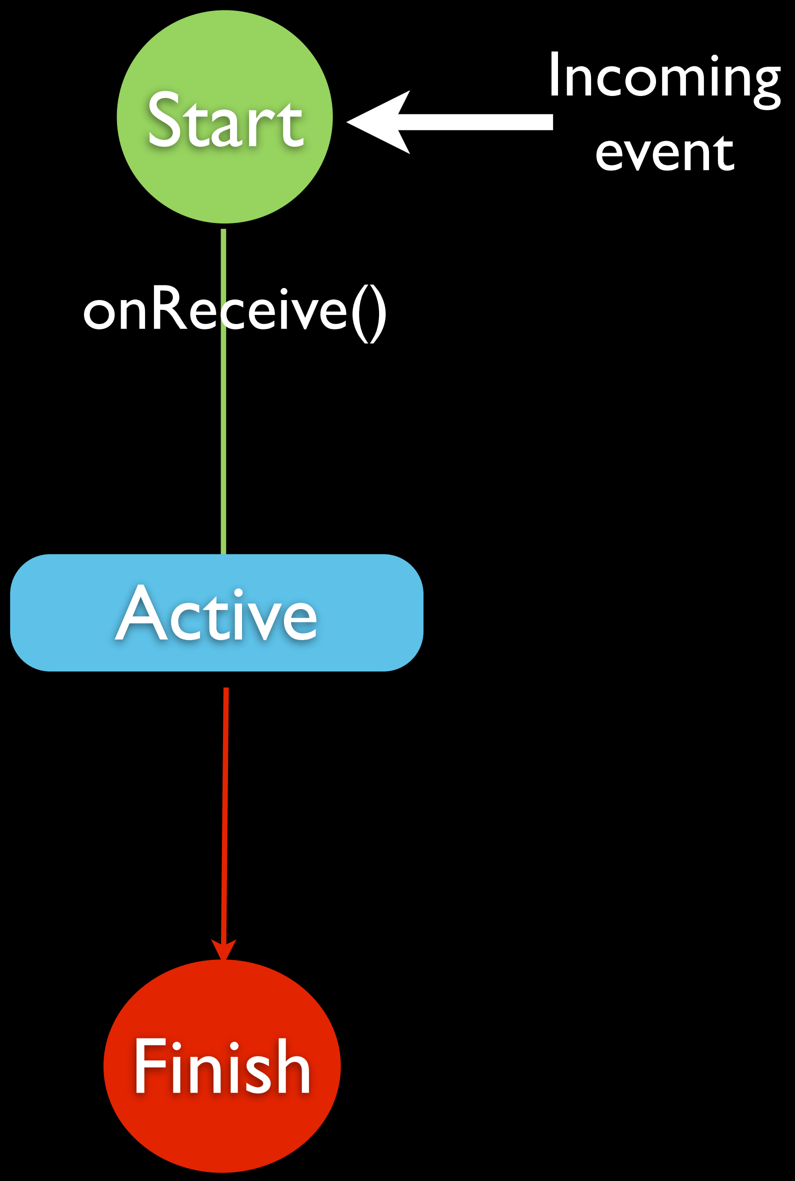
- Receive broadcasted events
  - System and Apps
- Subclass
  - onReceive() $\longleftarrow$ UI Thread
- Registration
  - Static
  - Dynamic
    - Context.registerReceiver()

BroadcastReceiver

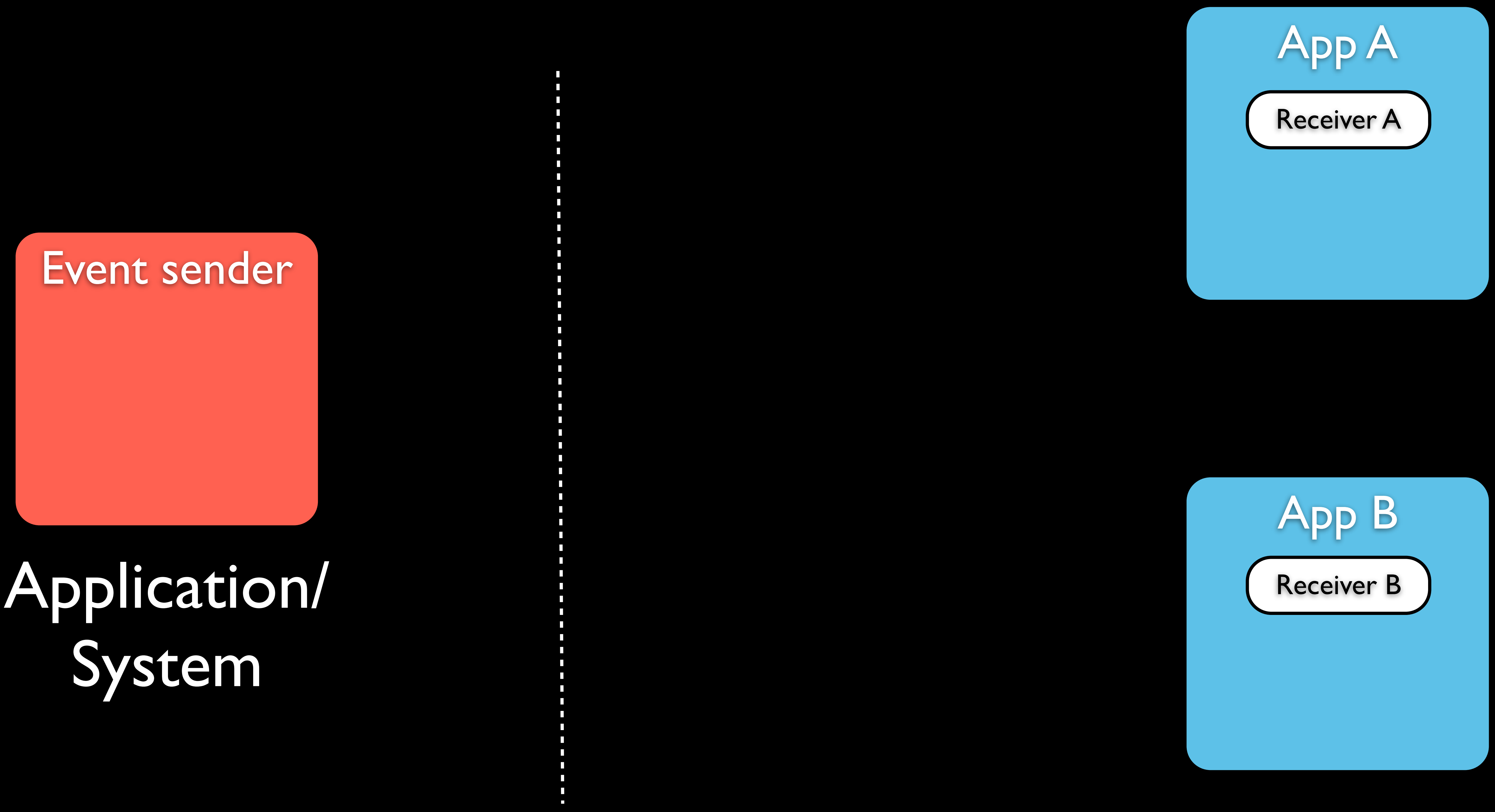
- Receive broadcasted events
  - System and Apps
- Subclass
  - onReceive() ← UI Thread
- Registration
  - Static
  - Dynamic
    - Context.registerReceiver()

```
<manifest>
<application>

  <receiver android:name="com.jayway.MyReceiver"/>

</application>
</manifest>
```

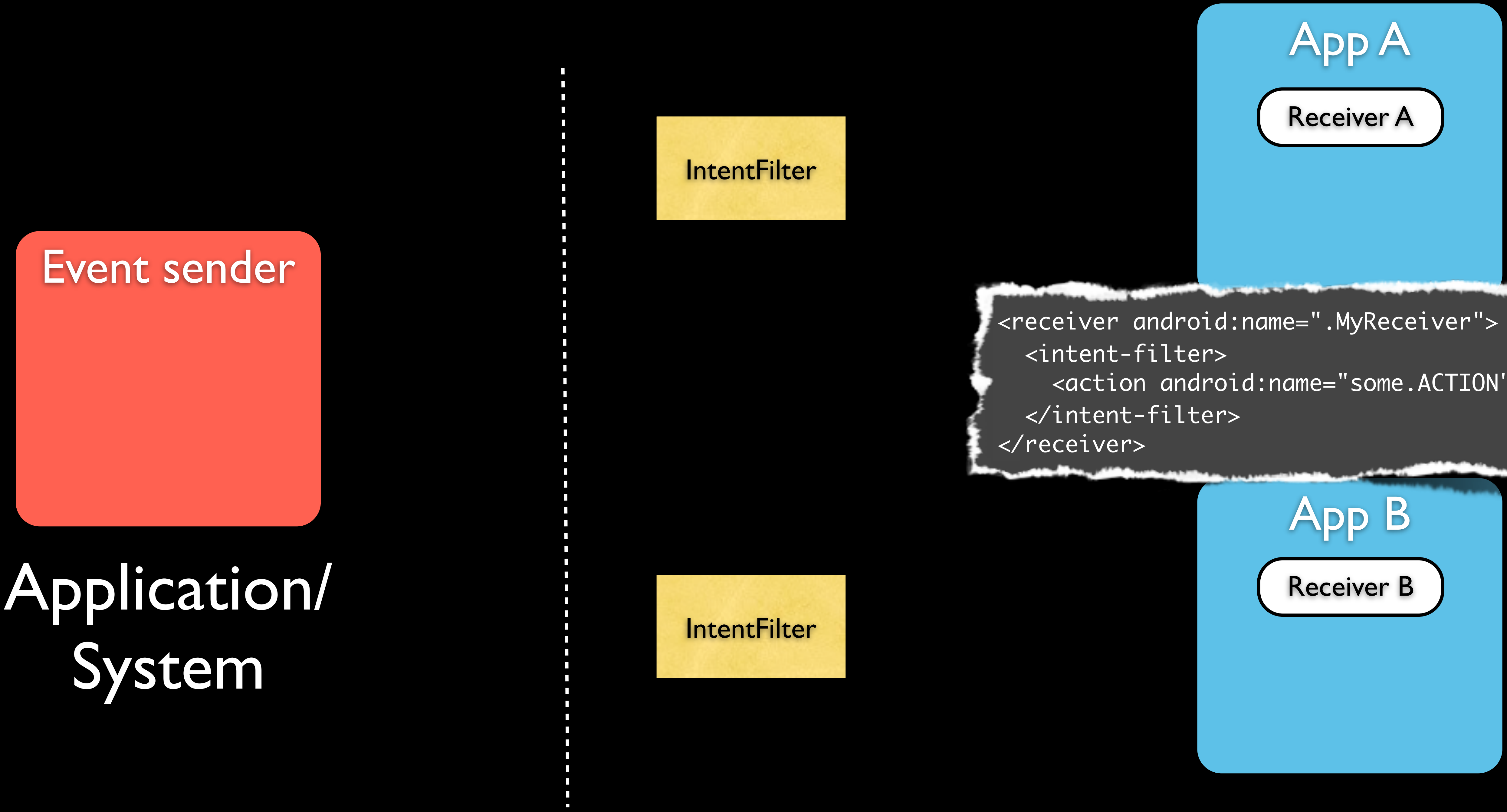
Lifecycle



Normal Broadcasts



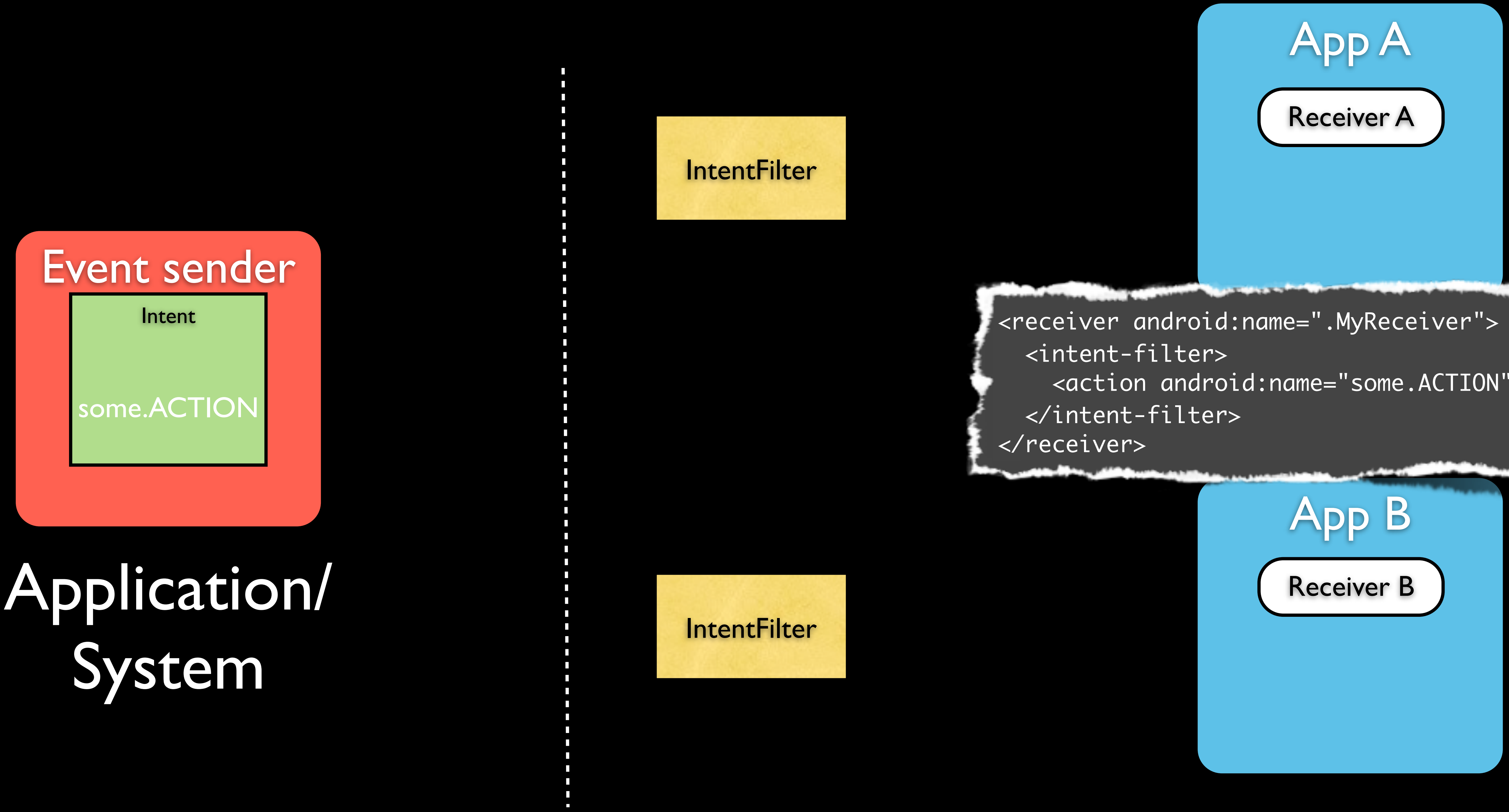
Normal Broadcasts



```
<receiver android:name=".MyReceiver">
  <intent-filter>
    <action android:name="some.ACTION" />
  </intent-filter>
</receiver>
```

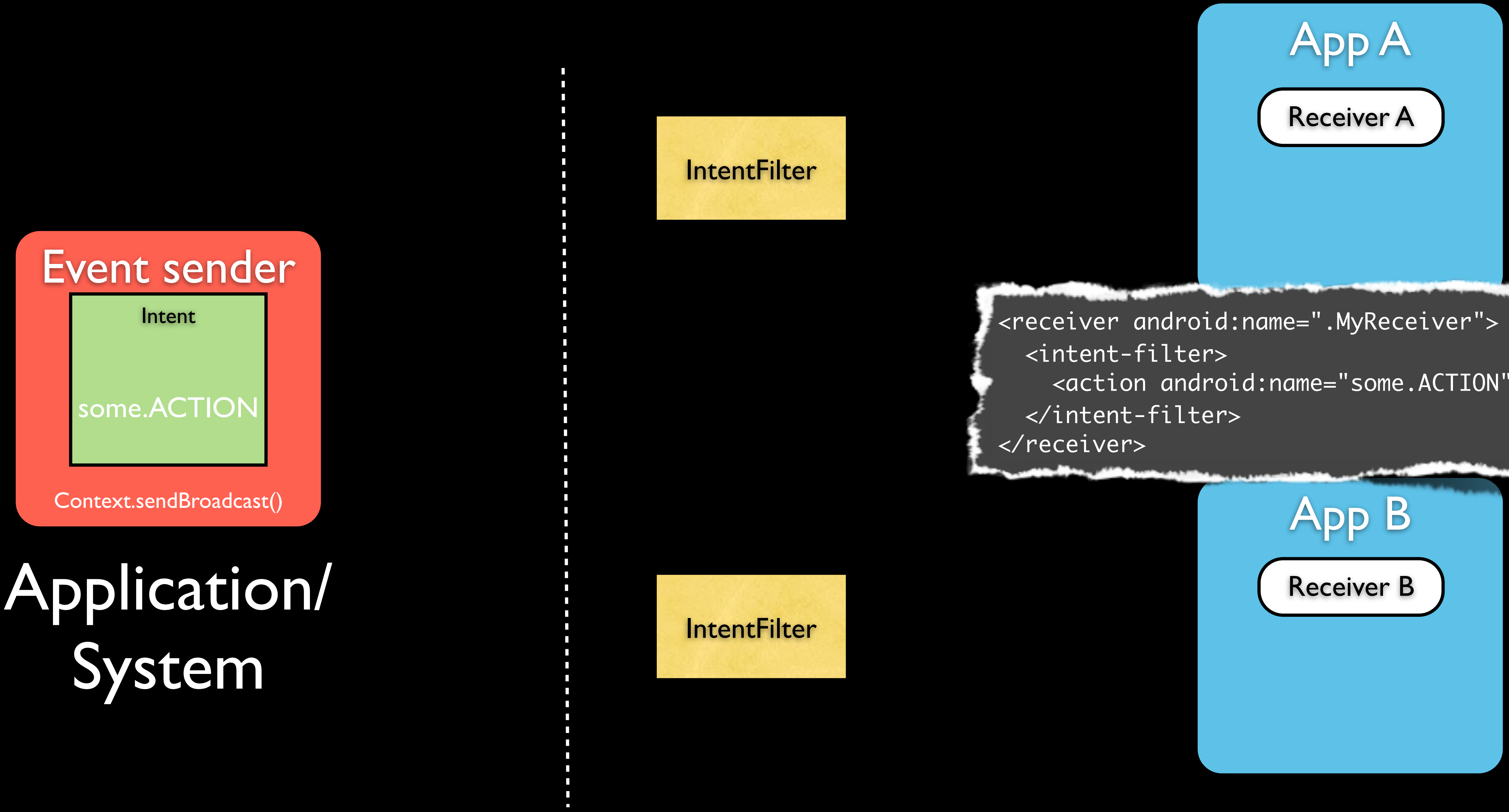
JAYWAY

Normal Broadcasts



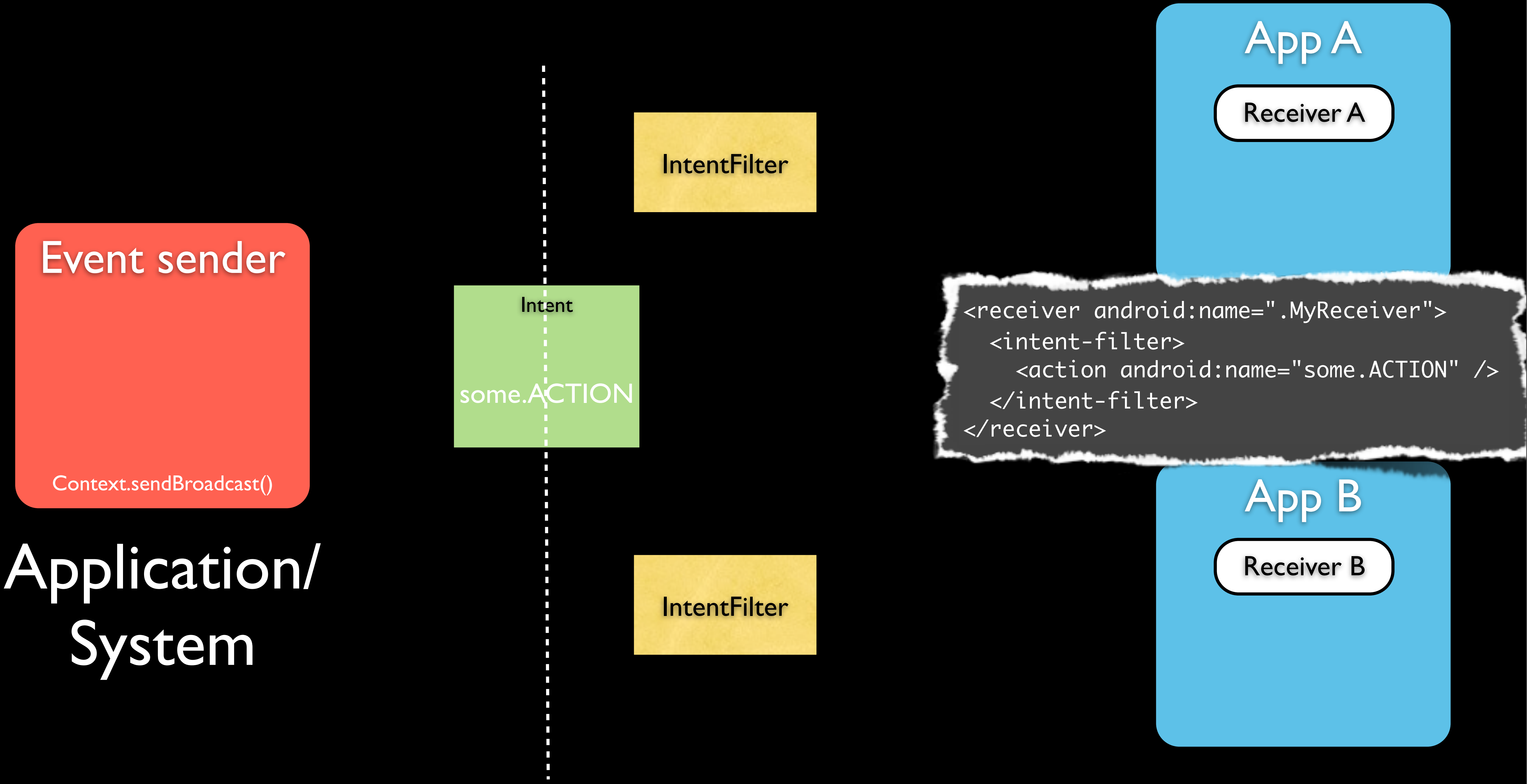
```
<receiver android:name=".MyReceiver">
  <intent-filter>
    <action android:name="some.ACTION" />
  </intent-filter>
</receiver>
```

Normal Broadcasts



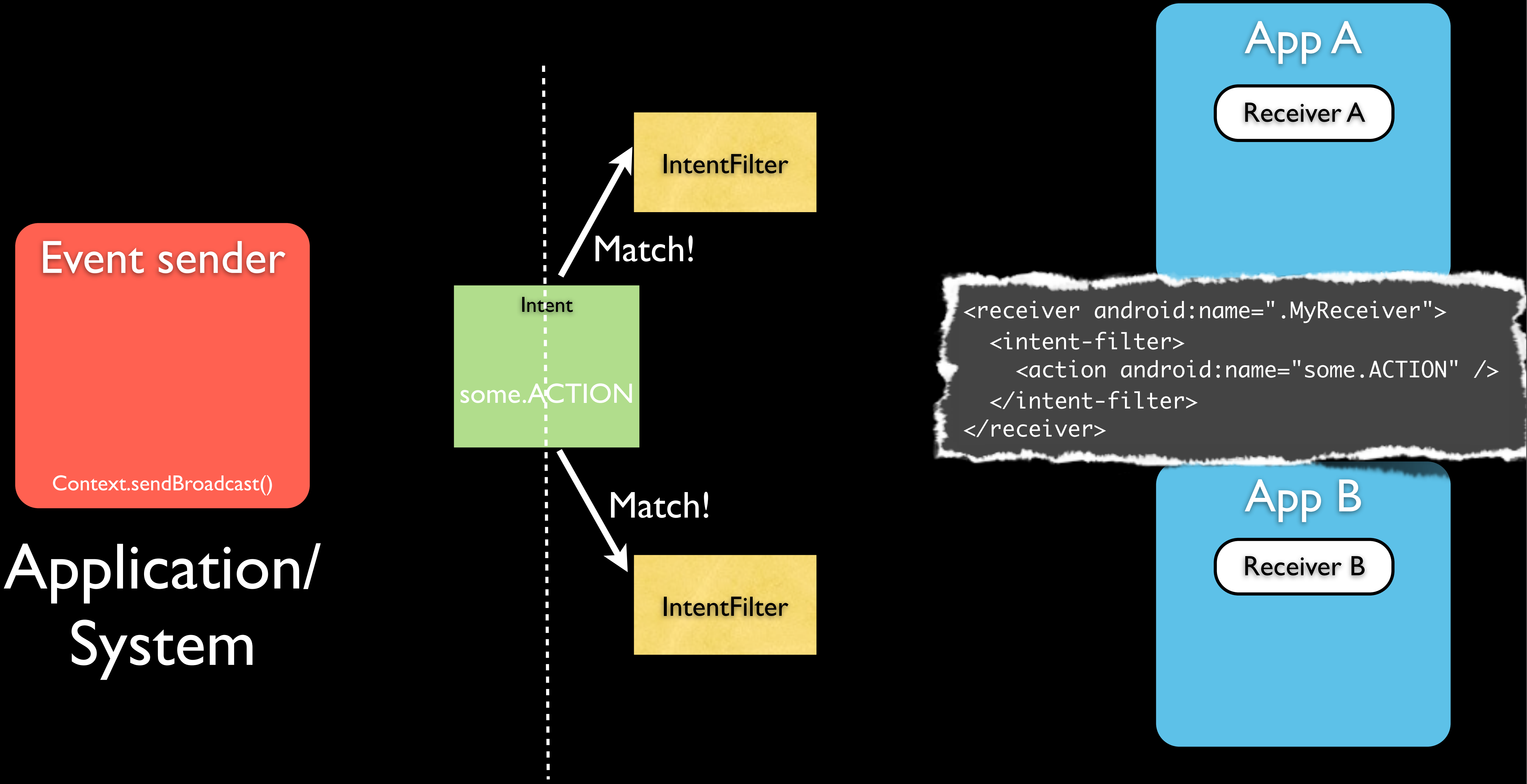
```
<receiver android:name=".MyReceiver">
  <intent-filter>
    <action android:name="some.ACTION" />
  </intent-filter>
</receiver>
```

Normal Broadcasts



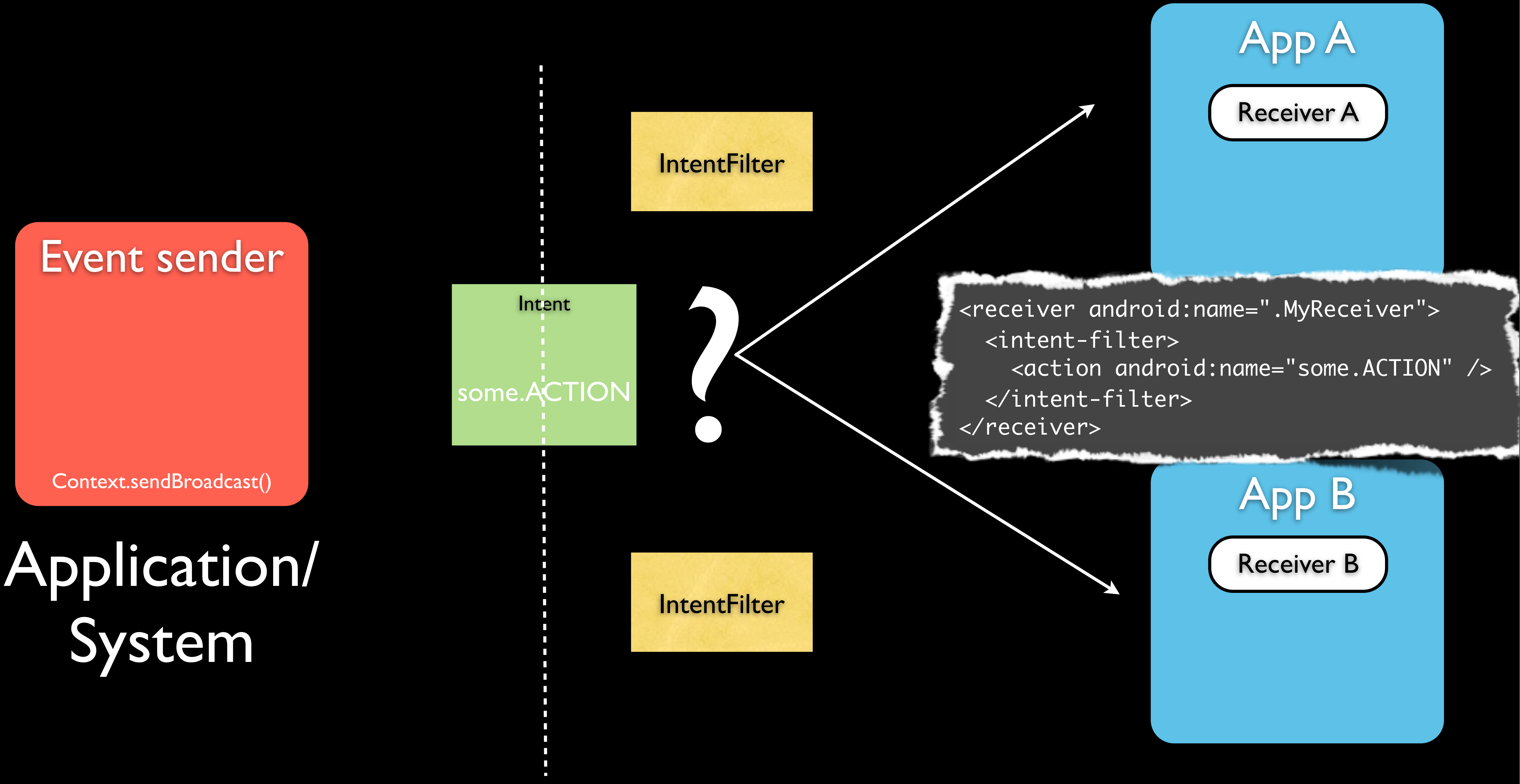
```
<receiver android:name=".MyReceiver">
  <intent-filter>
    <action android:name="some.ACTION" />
  </intent-filter>
</receiver>
```

Normal Broadcasts



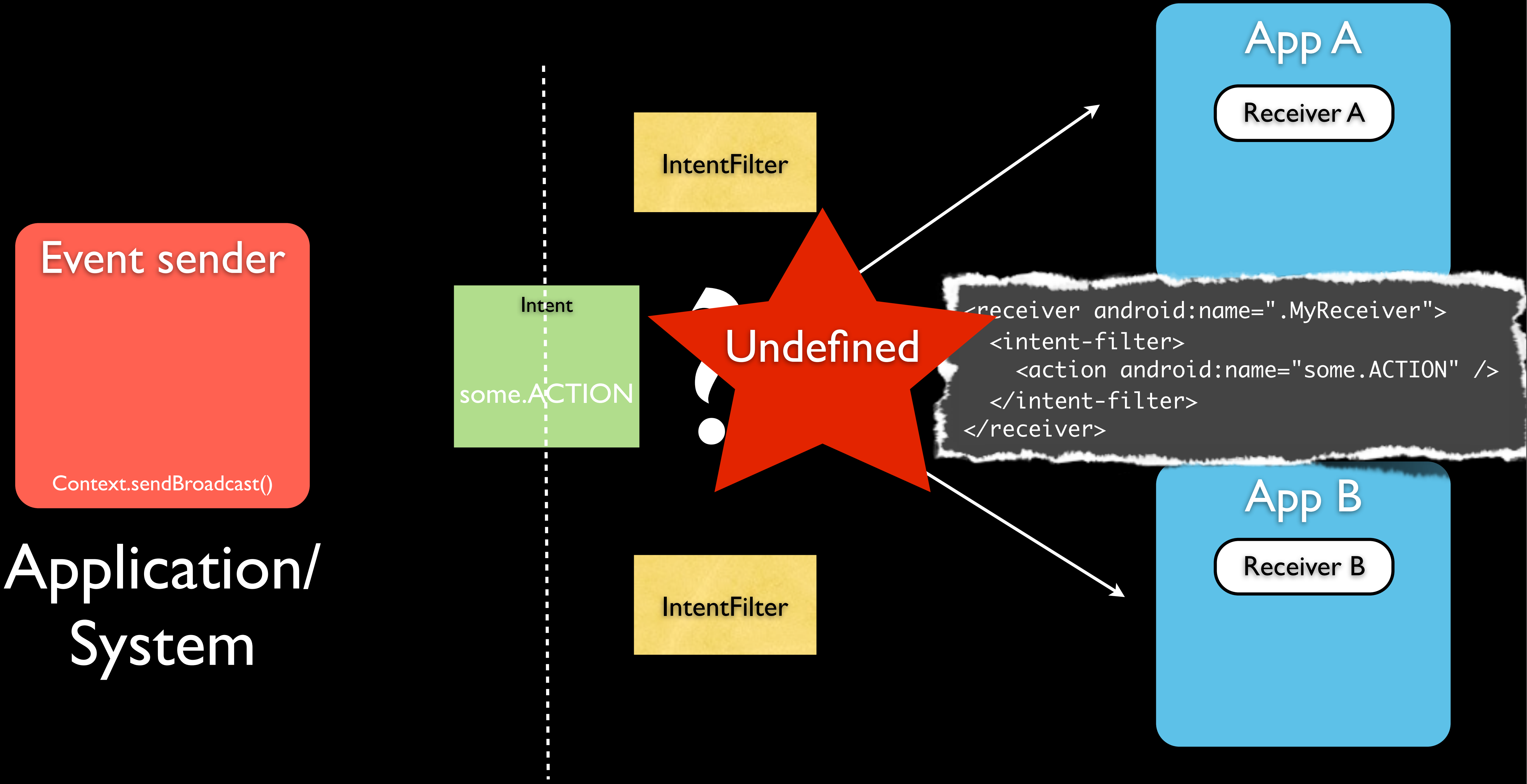
```
<receiver android:name=".MyReceiver">
  <intent-filter>
    <action android:name="some.ACTION" />
  </intent-filter>
</receiver>
```

Normal Broadcasts



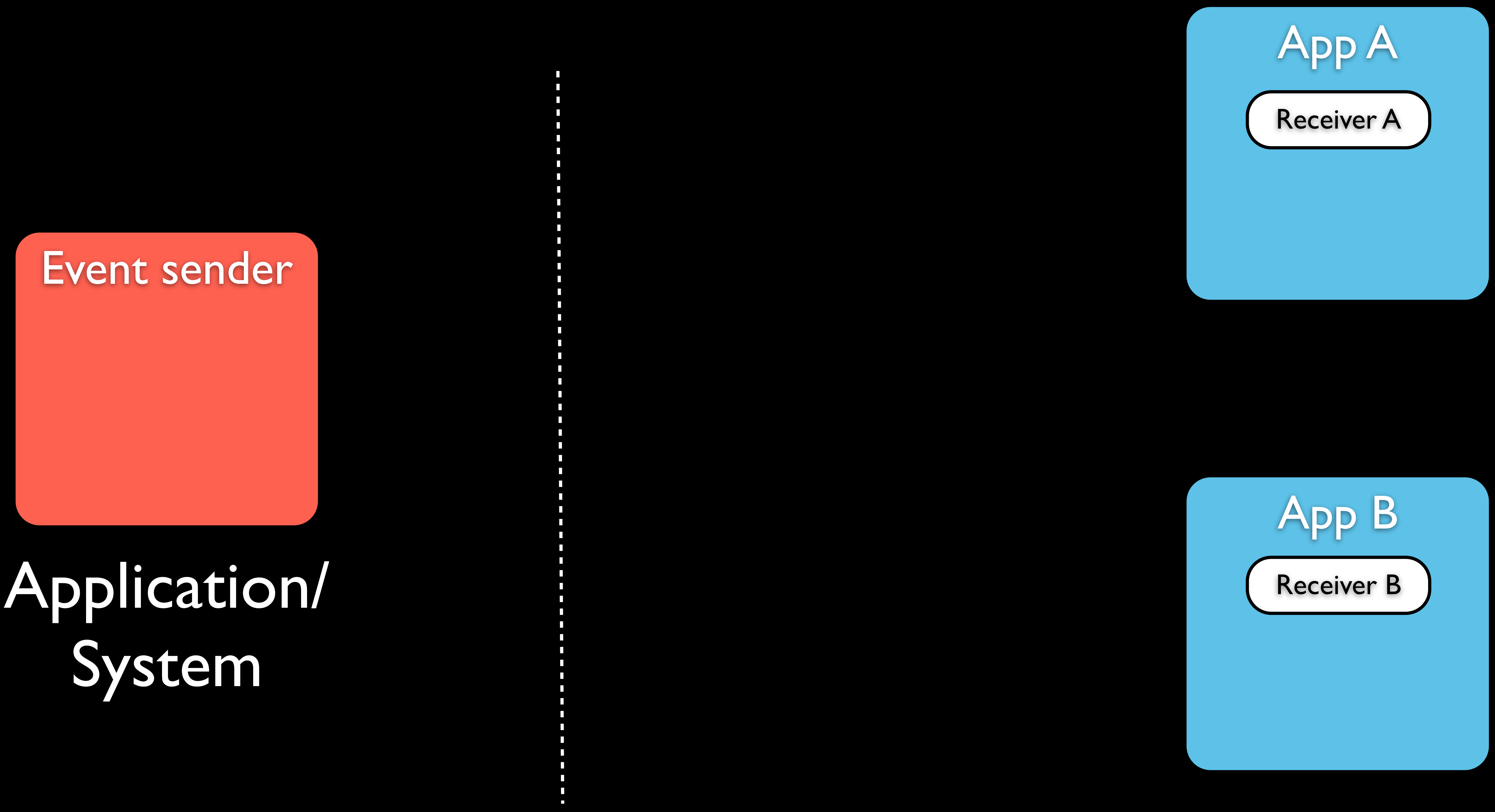
```
<receiver android:name=".MyReceiver">
  <intent-filter>
    <action android:name="some.ACTION" />
  </intent-filter>
</receiver>
```

Normal Broadcasts

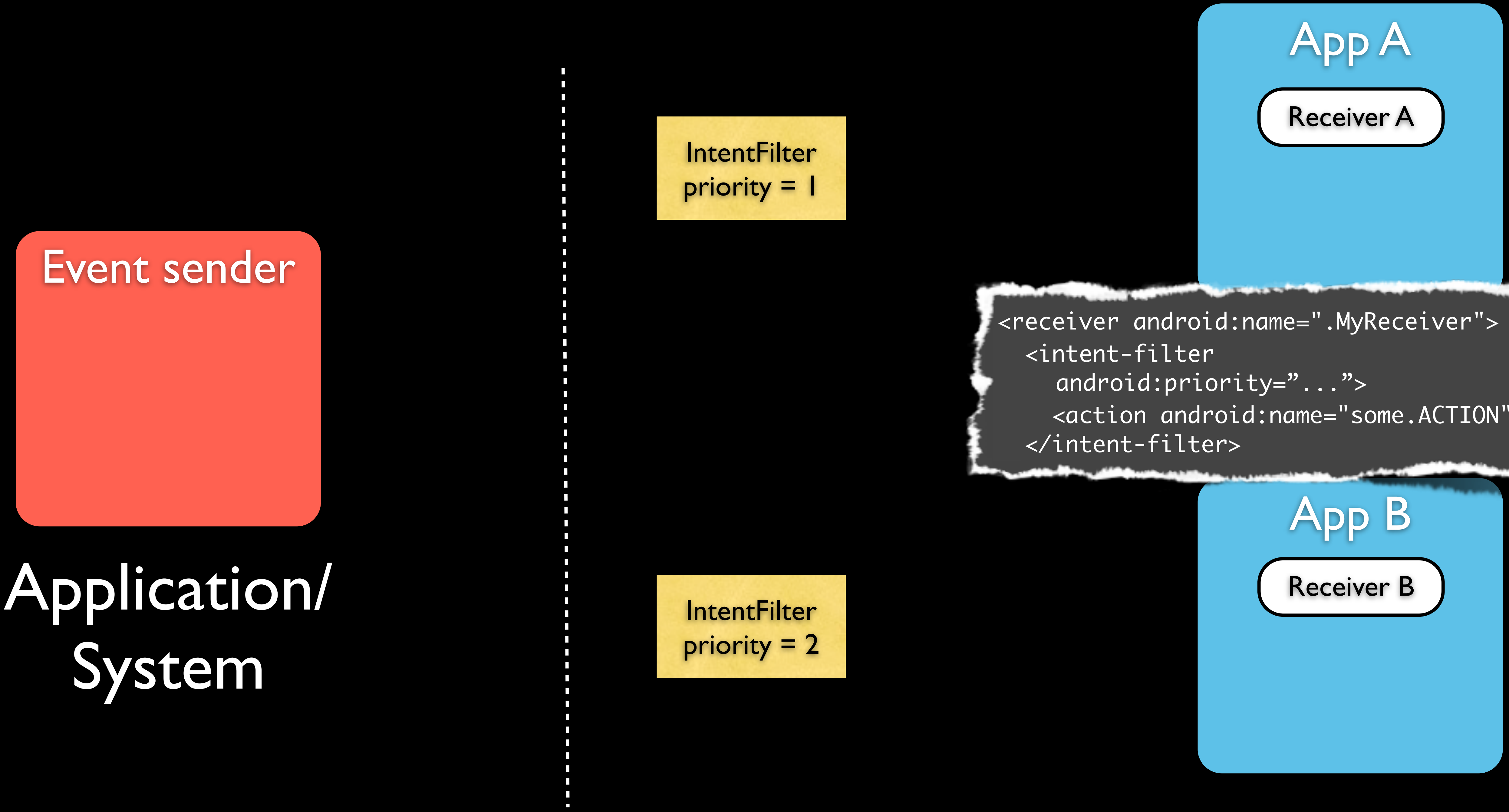


```
<receiver android:name=".MyReceiver">
  <intent-filter>
    <action android:name="some.ACTION" />
  </intent-filter>
</receiver>
```

Ordered Broadcasts

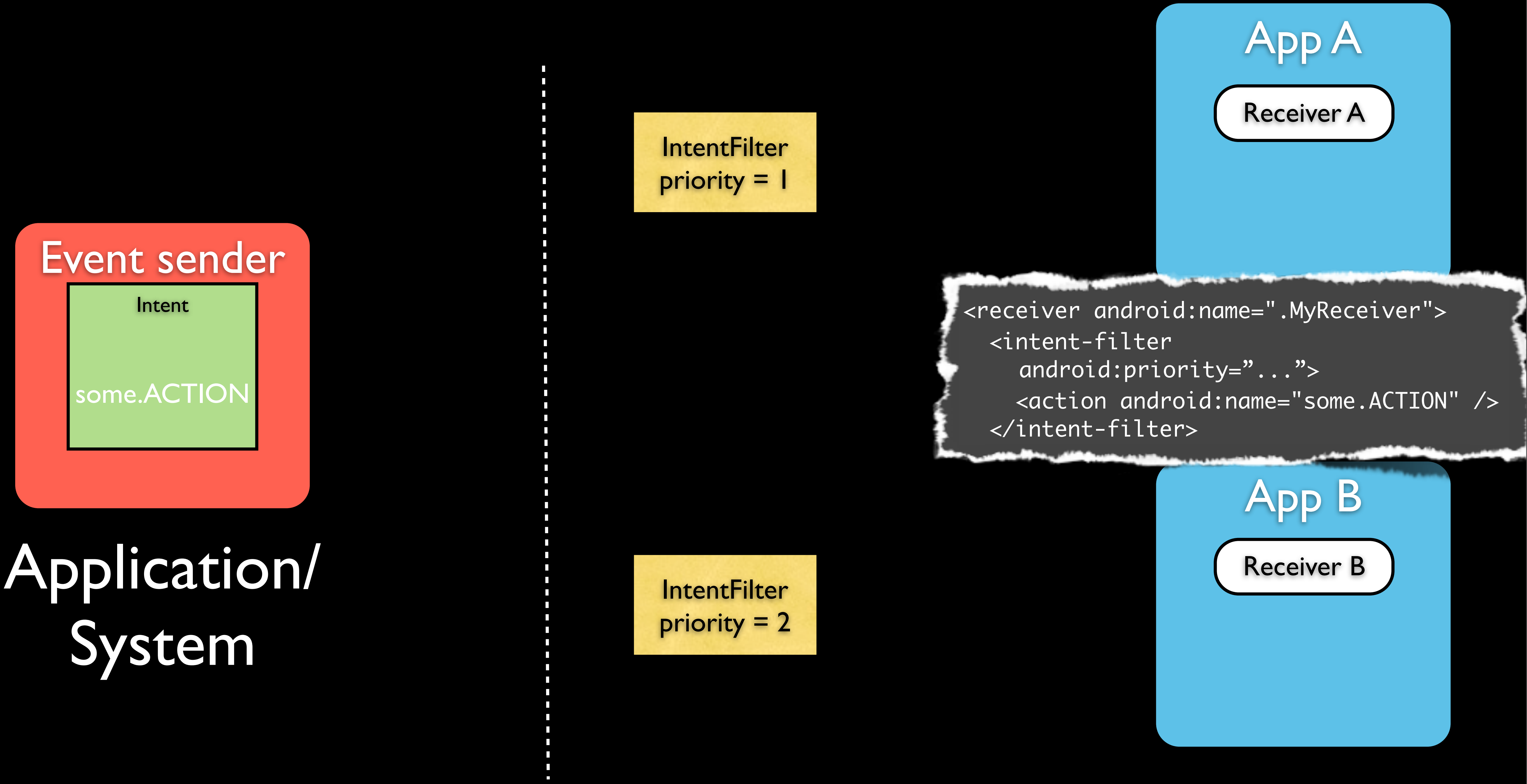


Ordered Broadcasts



```
<receiver android:name=".MyReceiver">
  <intent-filter
     android:priority=”...”>
    <action android:name="some.ACTION" />
  </intent-filter>
```

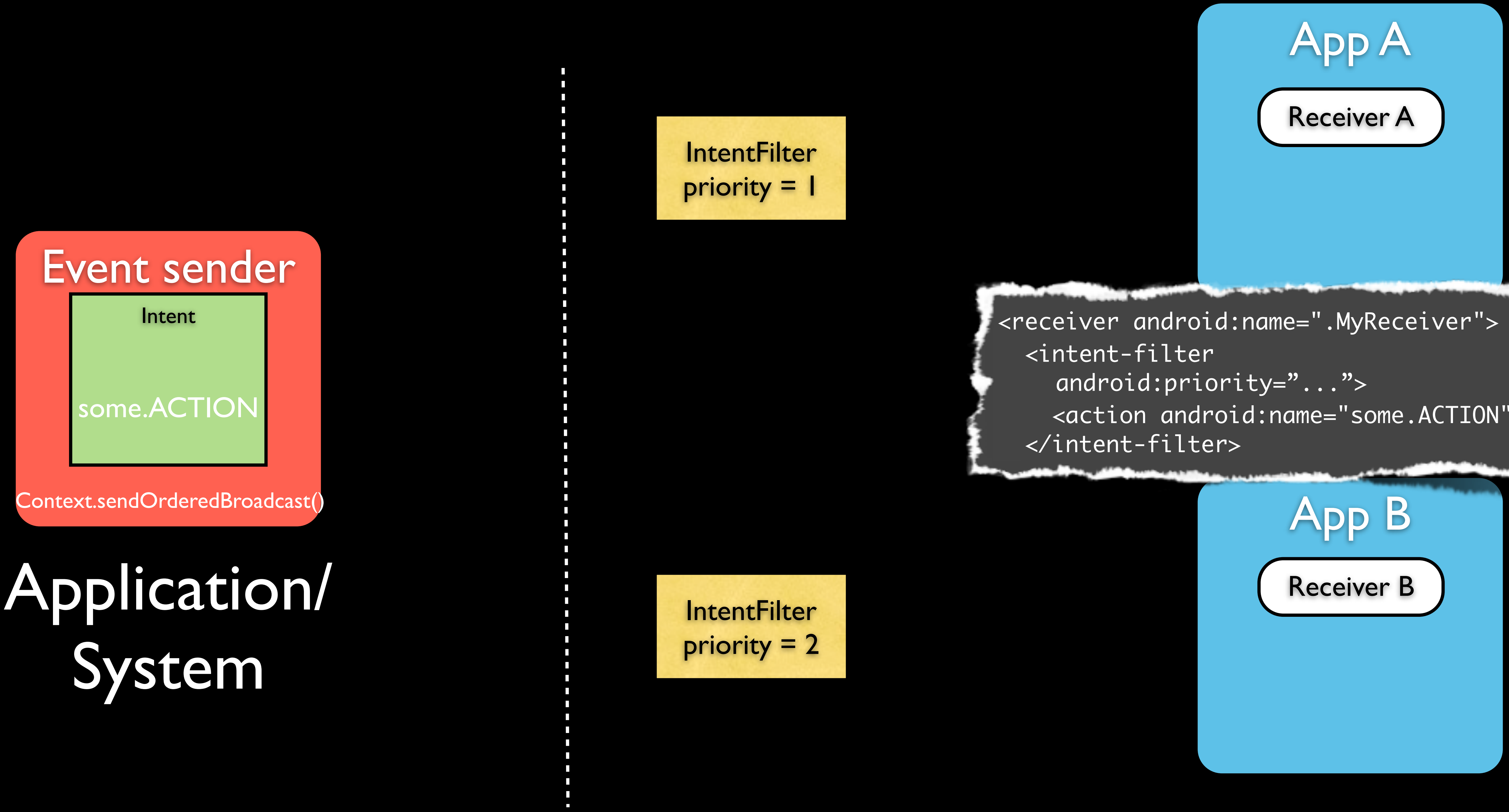
Ordered Broadcasts



```
<receiver android:name=".MyReceiver">
  <intent-filter
    android:priority=”...”>
    <action android:name="some.ACTION" />
  </intent-filter>
```

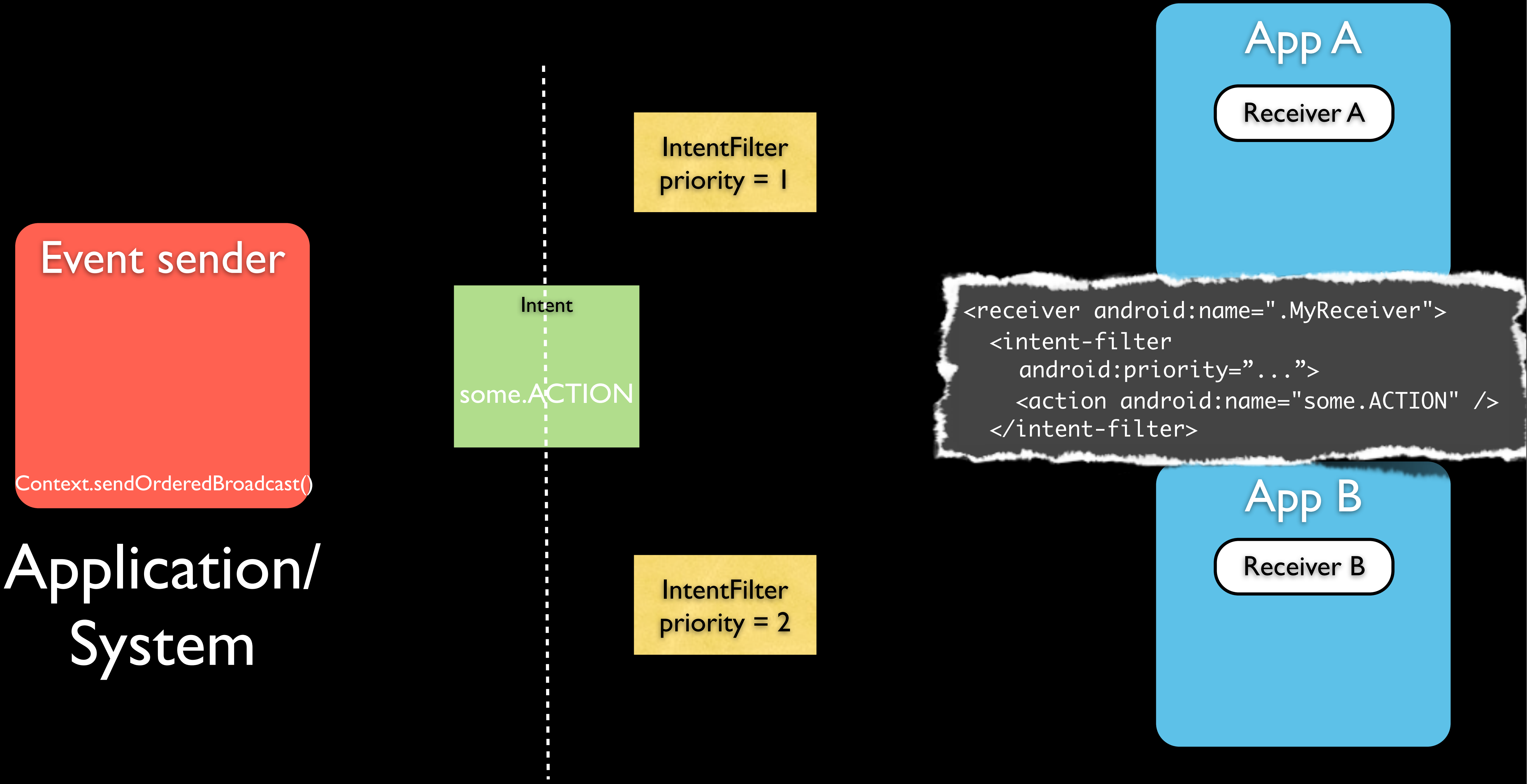
JAYWAY

Ordered Broadcasts


Event sender
Intent
some.ACTION
Context.sendOrderedBroadcast()
Application/ System
IntentFilter priority = 1
IntentFilter priority = 2
App A
Receiver A
App B
Receiver B

```
<receiver android:name=".MyReceiver">
  <intent-filter
    android:priority=”...”>
    <action android:name="some.ACTION" />
  </intent-filter>
```

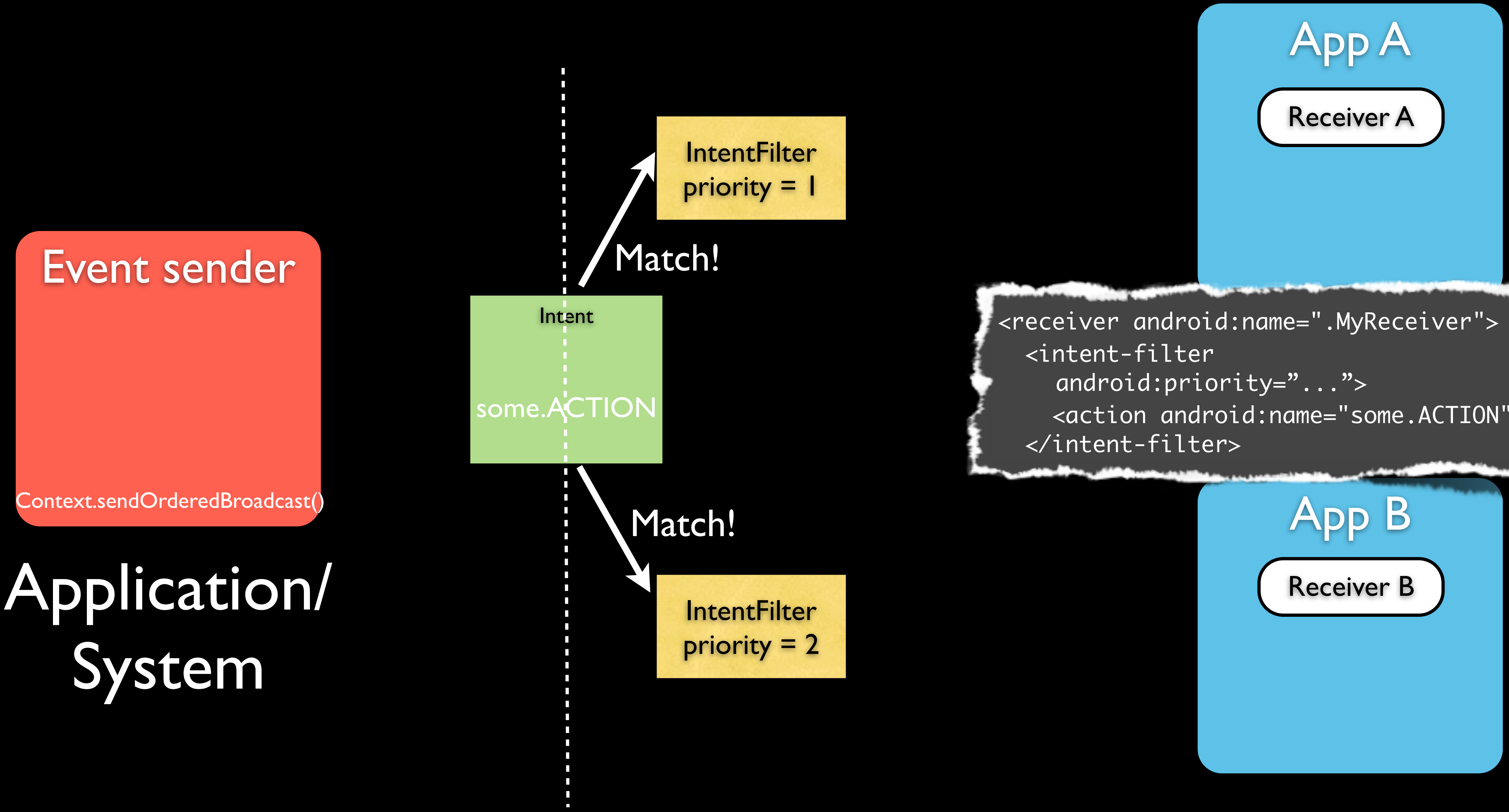
Ordered Broadcasts



```
<receiver android:name=".MyReceiver">
  <intent-filter
    android:priority=”...”>
    <action android:name="some.ACTION" />
  </intent-filter>
```

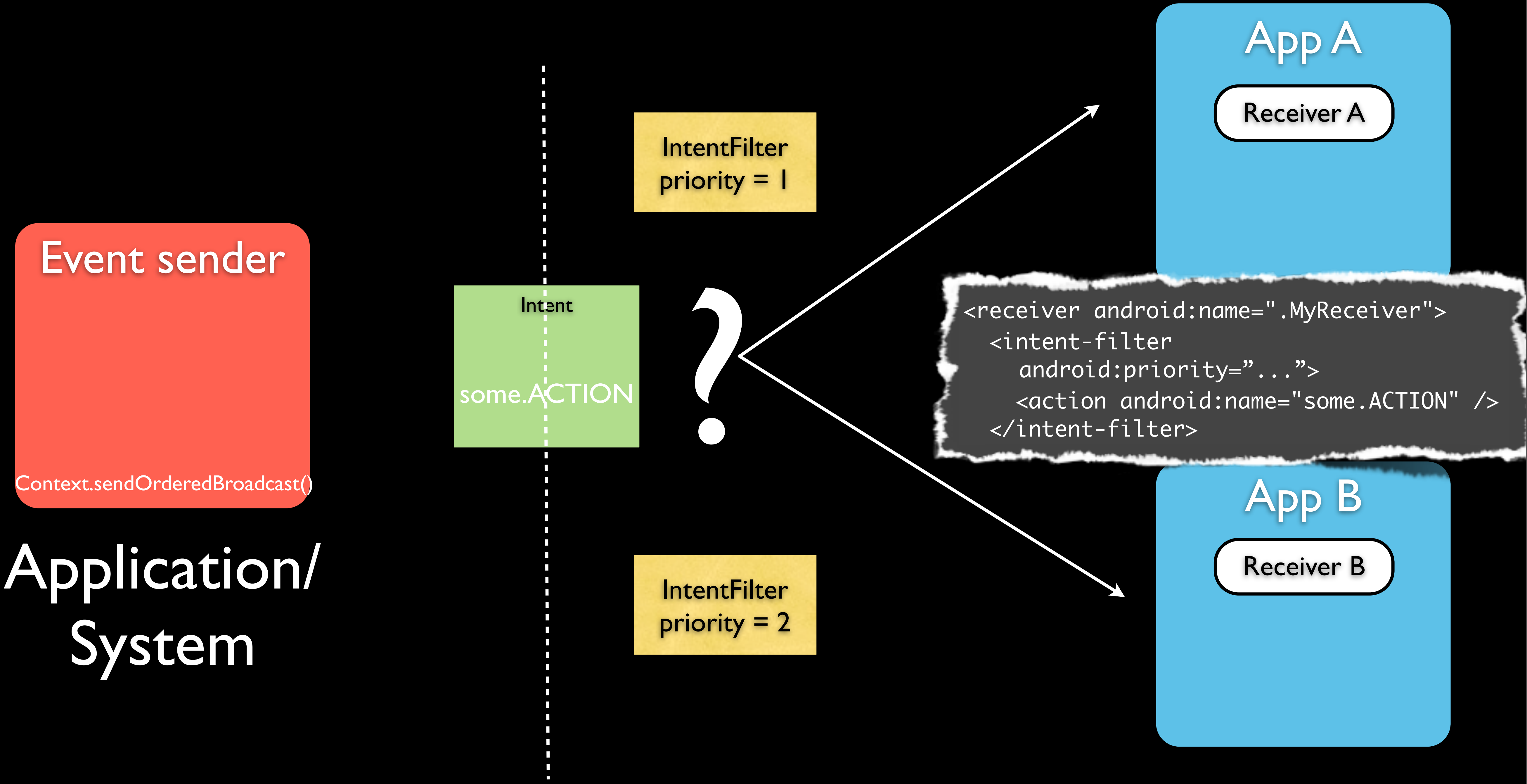
JAYWAY

Ordered Broadcasts



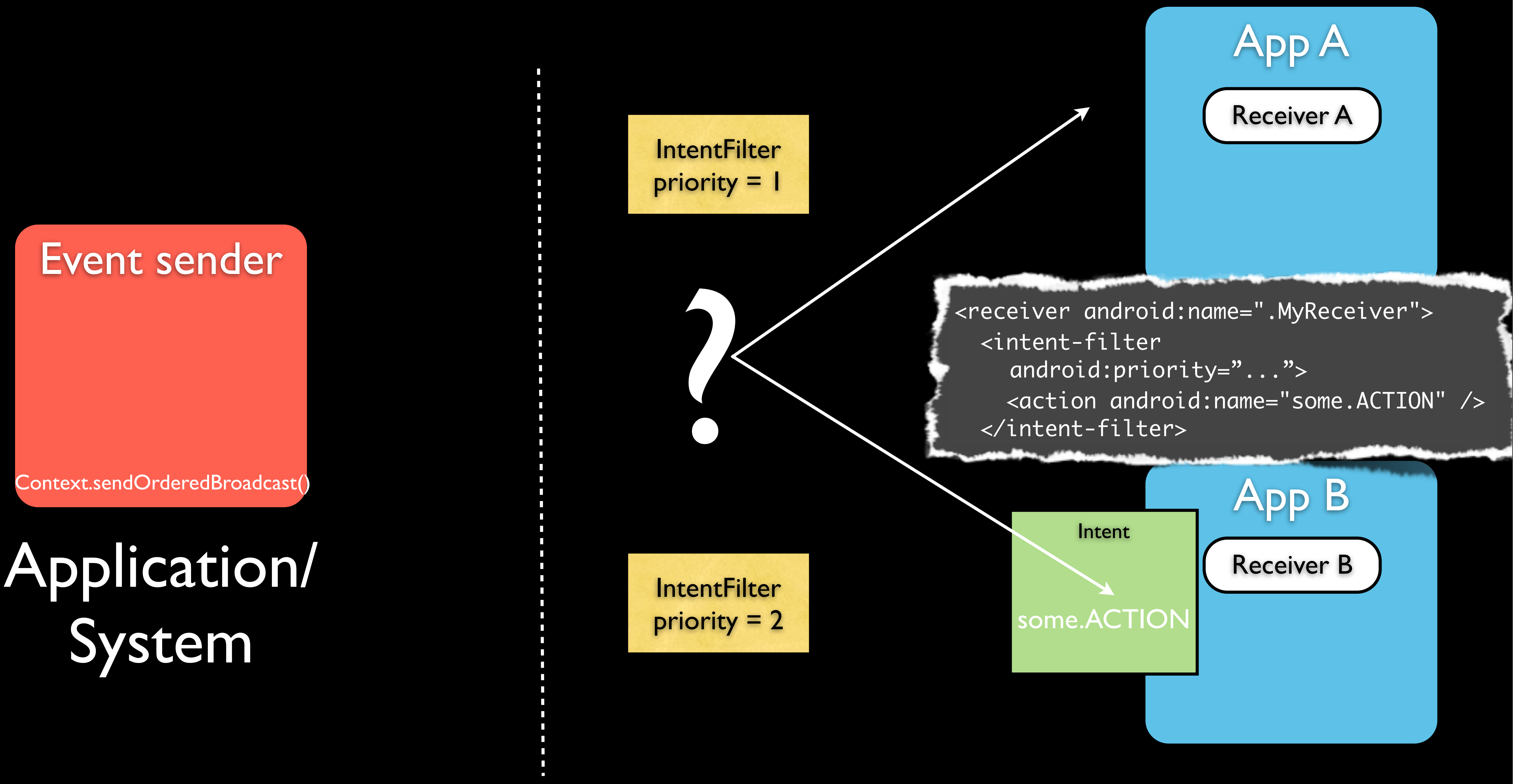
```
<receiver android:name=".MyReceiver">
  <intent-filter
    android:priority=”...”>
    <action android:name="some.ACTION" />
  </intent-filter>
```

Ordered Broadcasts



```
<receiver android:name=".MyReceiver">
  <intent-filter
    android:priority=”...”>
    <action android:name="some.ACTION" />
  </intent-filter>
```

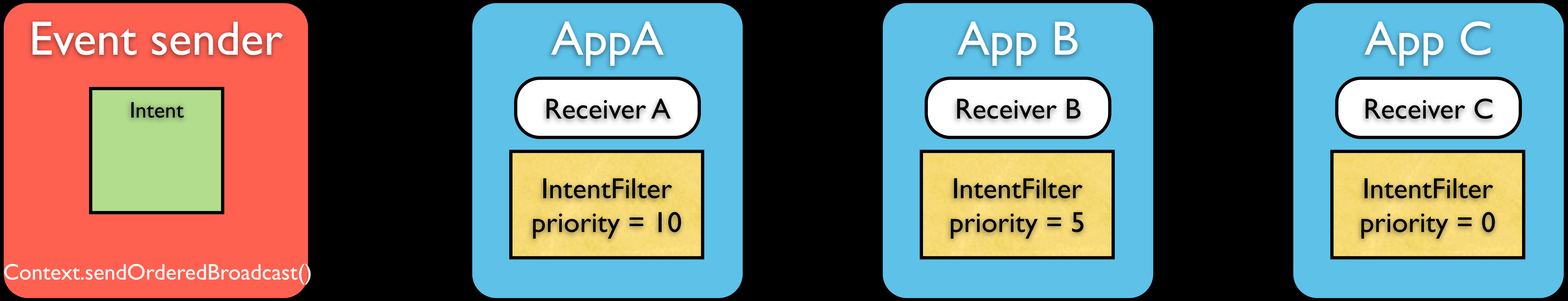
Ordered Broadcasts



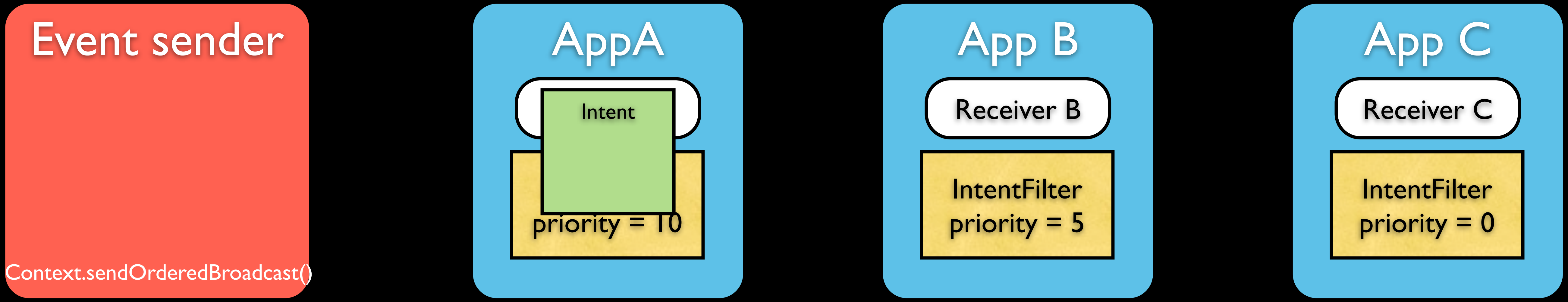
```
<receiver android:name=".MyReceiver">
  <intent-filter
    android:priority=”...”>
    <action android:name="some.ACTION" />
  </intent-filter>
```

JAYWAY

Ordered Broadcasts

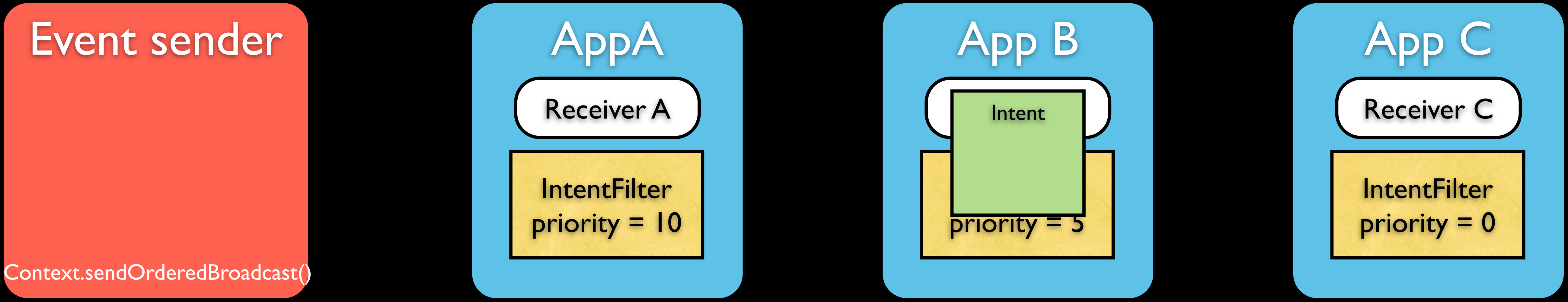


Ordered Broadcasts



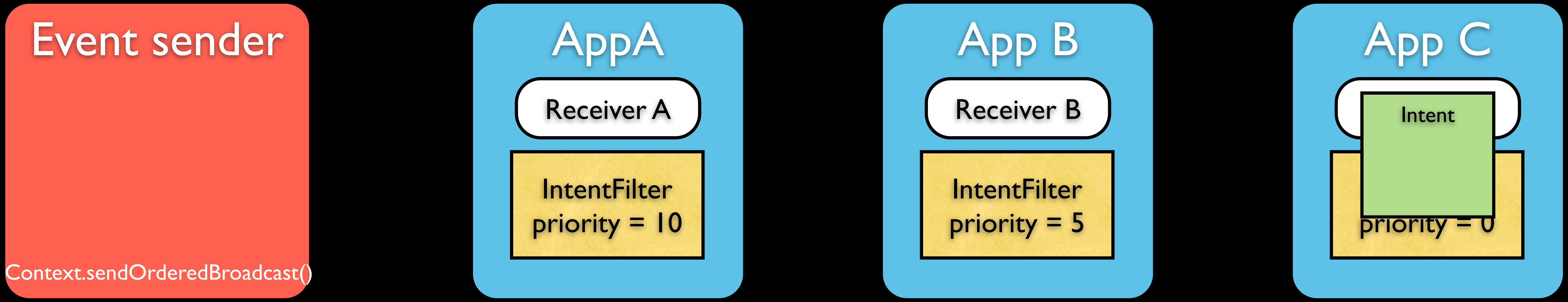
JAYWAY

Ordered Broadcasts



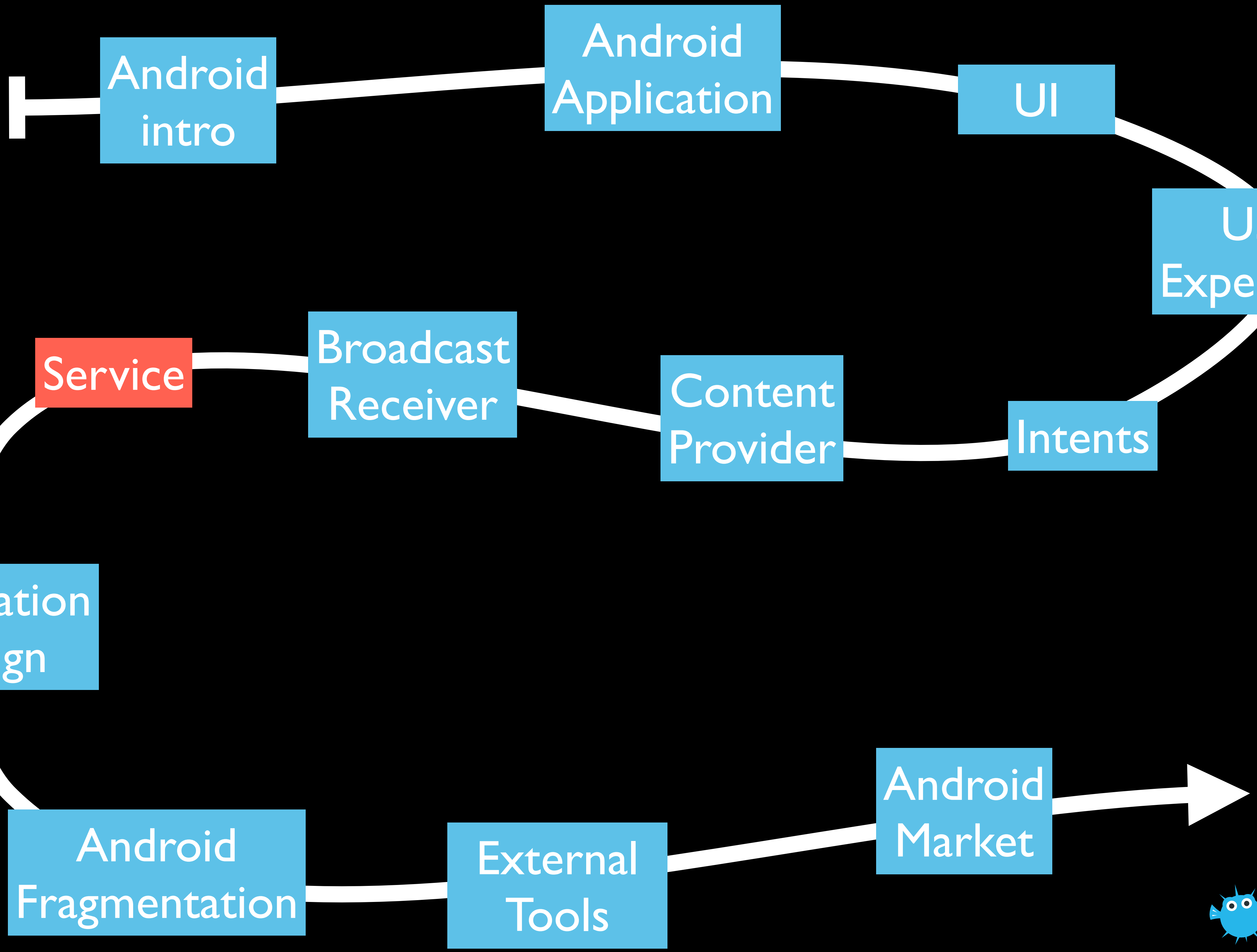
JAYWAY

Ordered Broadcasts



JAYWAY

Agenda



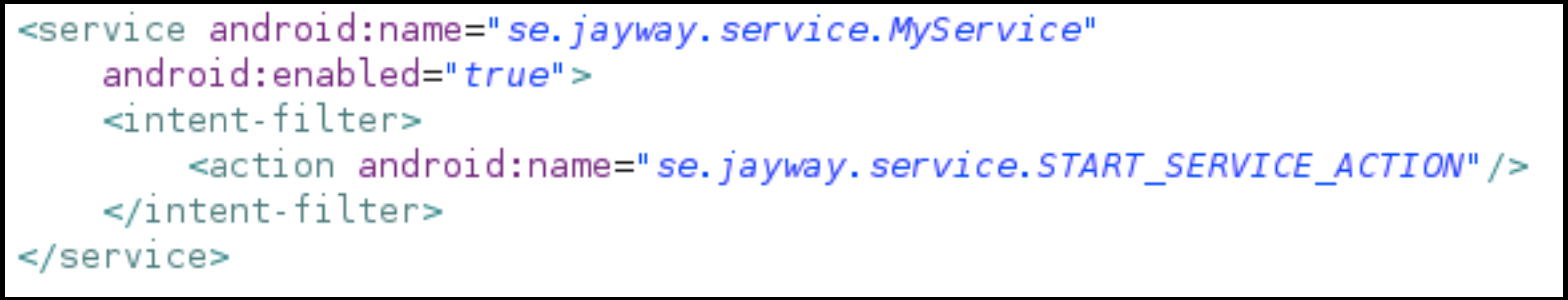
JAYWAY

Service

- Background operation that isn’t interacting with the user
- Less likely to be shut down by the system
- Started explicitly or Bound to
  - startService
  - bindService
- Expose functionality to other apps

AndroidMainifest.xml

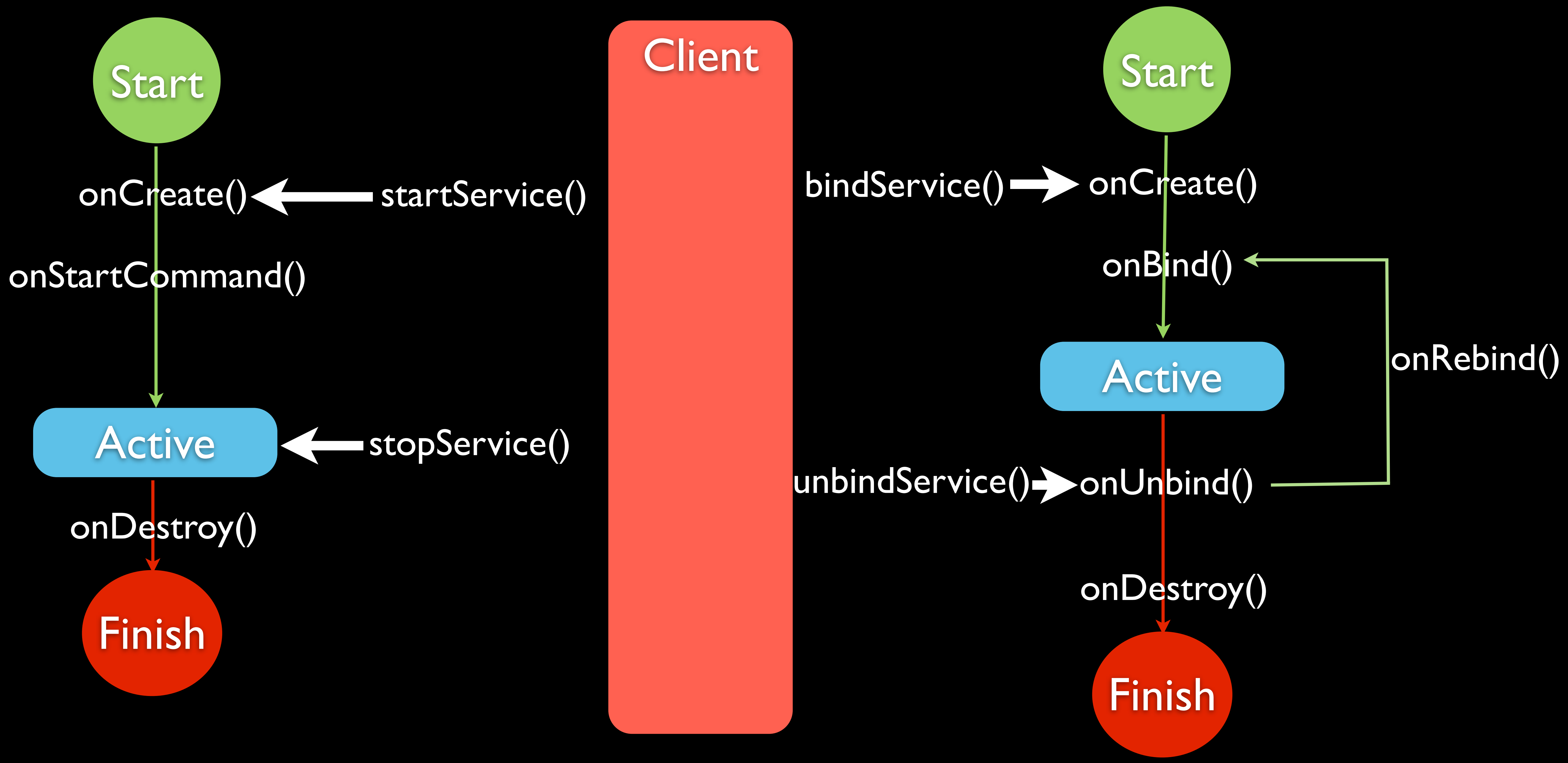
- Must be enabled to be started
- Permissions limit access to service
- android:process=”processName”
  - Run service in own service
    - “:a” -> Private process
    - “”:A -> Global process



```xml
<service android:name="se.jayway.service.MyService"
    android:enabled="true">
    <intent-filter>
        <action android:name="se.jayway.service.START_SERVICE_ACTION"/>
    </intent-filter>
</service>
```

JAYWAY

Lifecycle


Start
onCreate()
startService()
onStartCommand()
Active
stopService()
onDestroy()
Finish
Client
bindService()
onCreate()
Start
onBind()
Active
onRebind()
unbindService()
onUnbind()
onDestroy()
Finish

Start Modes

- Explicitly started services (startService())
- onStartCommand defines start mode
    - START_STICKY
    - START_NOT_STICKY

```
public int onStartCommand(Intent intent, int flags, int startId) {
        return START_STICKY;
    }
```

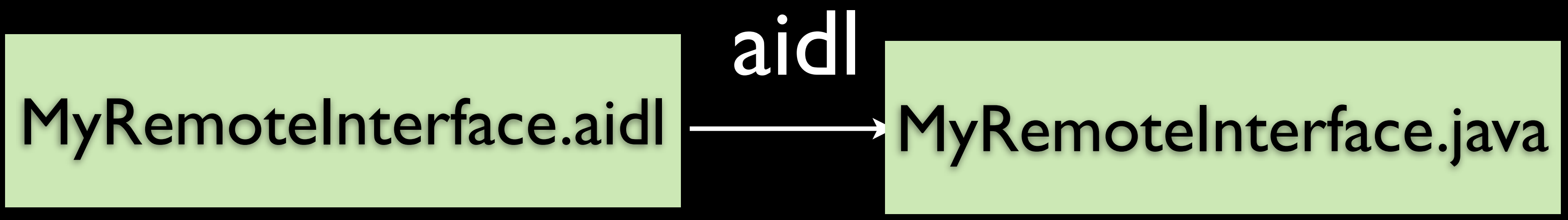
Remote Service

- Service that doesn’t run in the same process as the client application
- Communication between Linux processes through IPC
- Synchronous calls
- Android provides an abstraction mechanism for IPC
  - AIDL

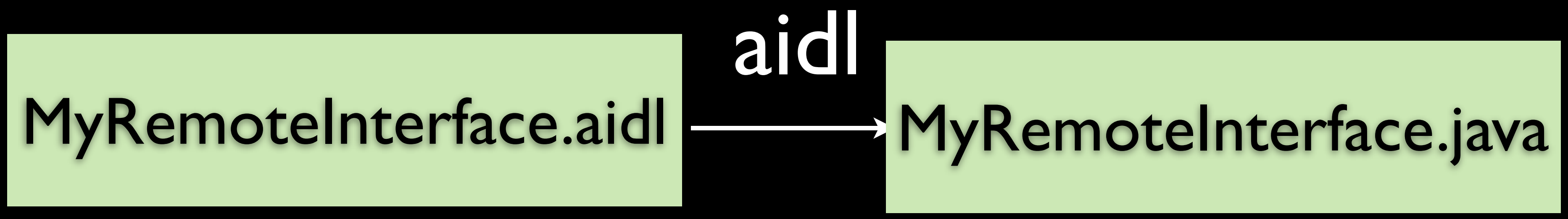
Android IPC Abstraction

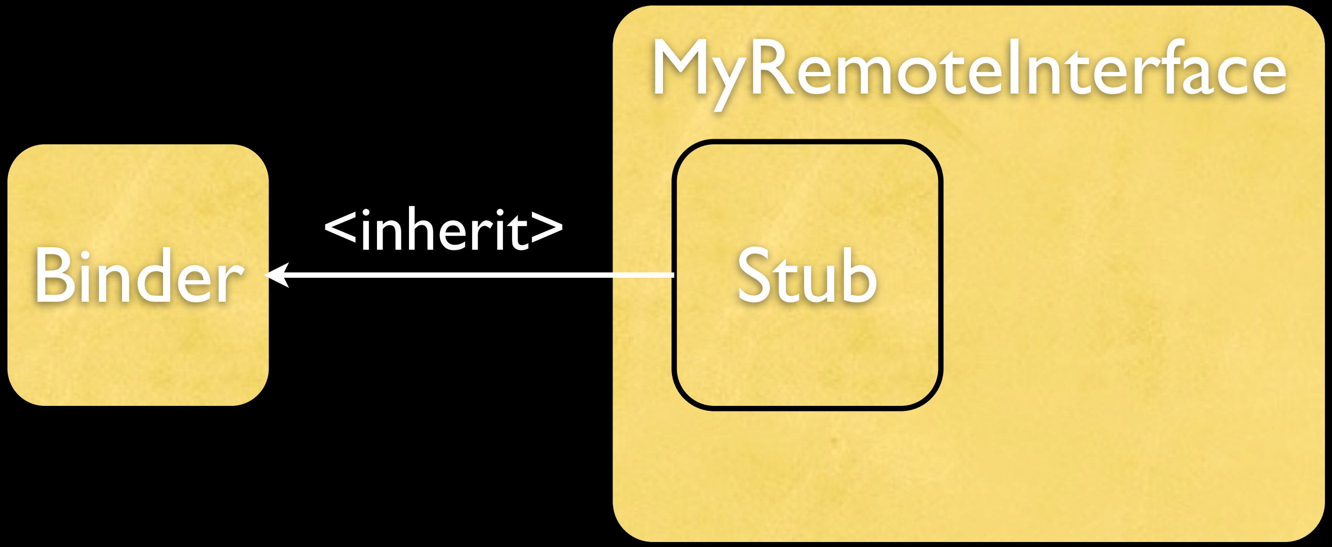
- Service defines the interface in an AIDL-file
    - Limited data set
        - Primitives, String, List, Map, CharSequence
        - Parcelables - Custom classes
    - aidl: Compiles aidl-files
- Implement AIDL interface
- Expose the interface

Android IPC Abstraction



Android IPC Abstraction





System Services

- Context.getSystemService()
- adb shell service list

WINDOW_SERVICE

LAYOUT_INFLATER_SERVICE

ACTIVITY_SERVICE

POWER_SERVICE

ALARM_SERVICE

NOTIFICATION_SERVICE

KEYGUARD_SERVICE

LOCATION_SERVICE

VIBRATOR_SERVICE

CONNECTIVITY_SERVICE

WIFI_SERVICE

System Services

- Con
- adb

WIND

LAYO

ACTIV

POWE

ALAR

NOTIFICATION_SERVICE

KEYGUARD_SERVICE

LOCATION_SERVICE

VIBRATOR_SERVICE

CONNECTIVITY_SERVICE

WIFI_SERVICE

Watch out!

The system services may be closely related to the current context. Only use it within the current context.

JAYWAY

NotificationManager

- Notify the user that something has happened
    - Put message in status bar
    - Flash backlight
    - Flash LED’s
    - Play sound

NotificationManager

- Noti ed
    - P
    - F
    - F
    - Play sound

User experience!

Use notifications when informing the user of events in the background. Don't show Activity or Dialog.

JAYWAY

NotificationManager

```
String ns = Context.NOTIFICATION_SERVICE;
NotificationManager mNotificationManager = (NotificationManager) getSystemService(ns);

int icon = R.drawable.notification_icon;
CharSequence tickerText = "Hello";
long when = System.currentTimeMillis();
Notification notification = new Notification(icon, tickerText, when);

Context context = getApplicationContext();
CharSequence contentTitle = "My notification";
CharSequence contentText = "Hello World!";
Intent notificationIntent = new Intent(this, MyClass.class);
PendingIntent contentIntent = PendingIntent.getActivity(this, 0, notificationIntent, 0);
notification.setLatestEventInfo(context, contentTitle, contentText, contentIntent);

private static final int HELLO_ID = 1;
mNotificationManager.notify(HELLO_ID, notification);
```

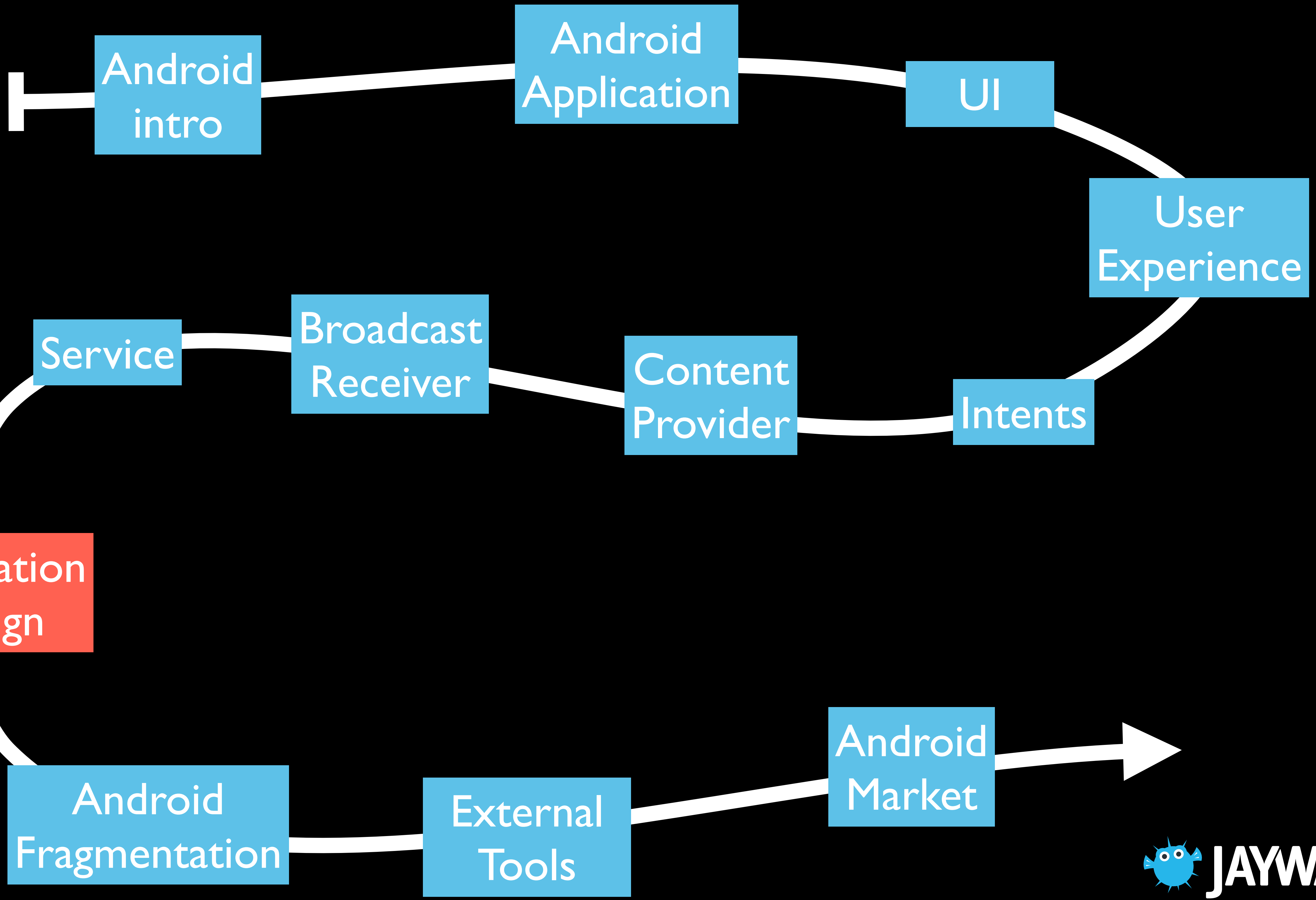
System service

Create Notification

Create Intent to send

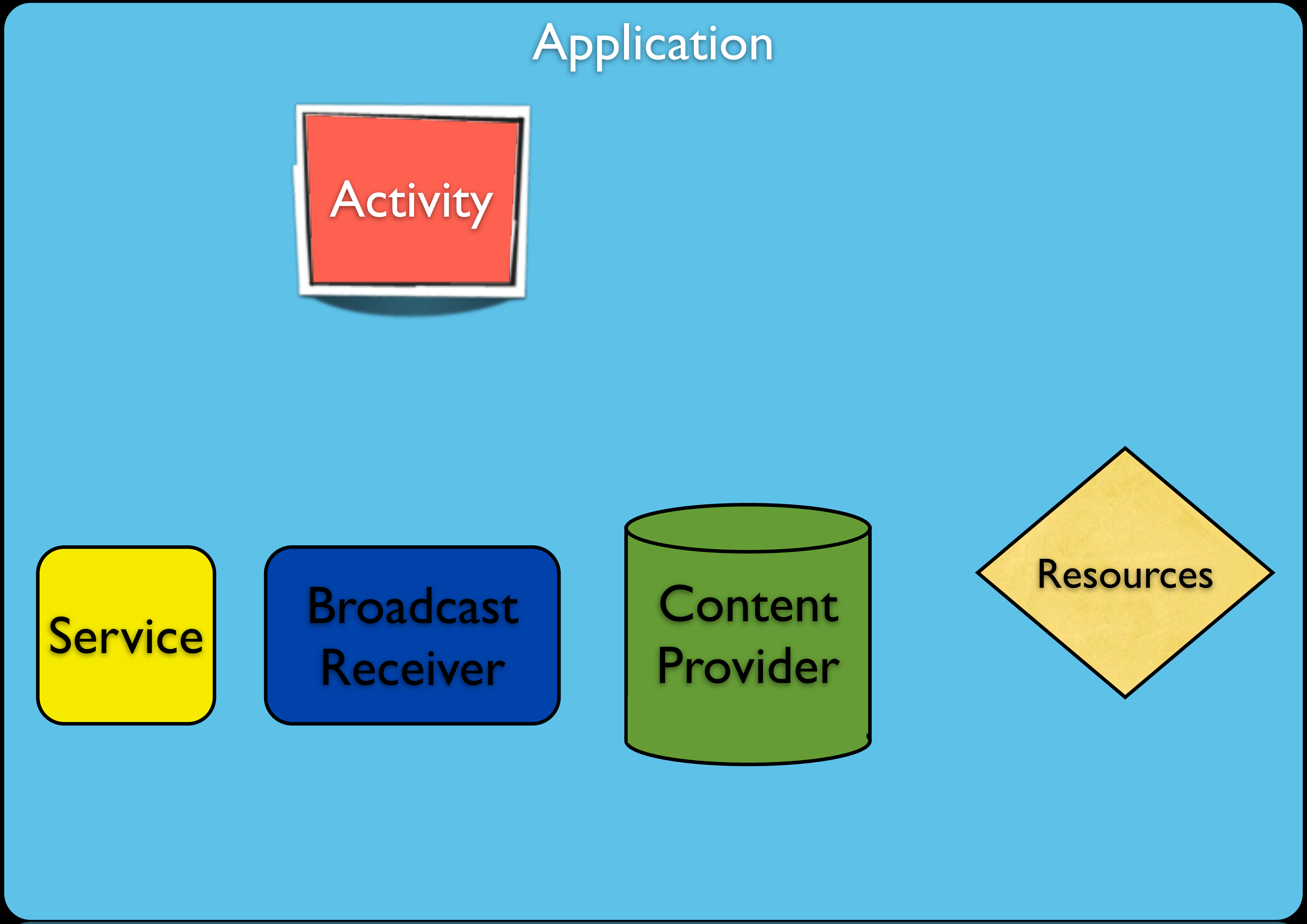
Let the platform have the notification

Agenda



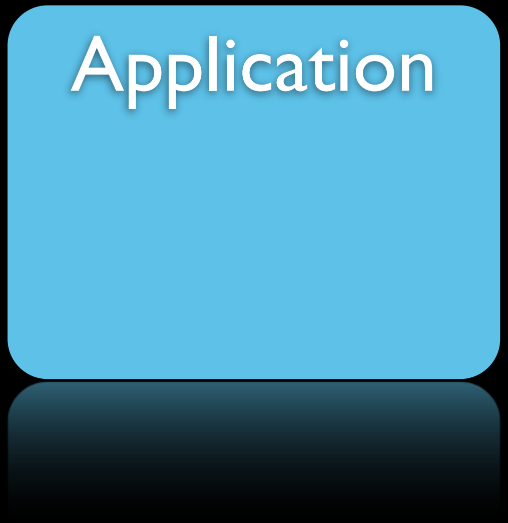
JAYWAY

Application Design



JAYWAY

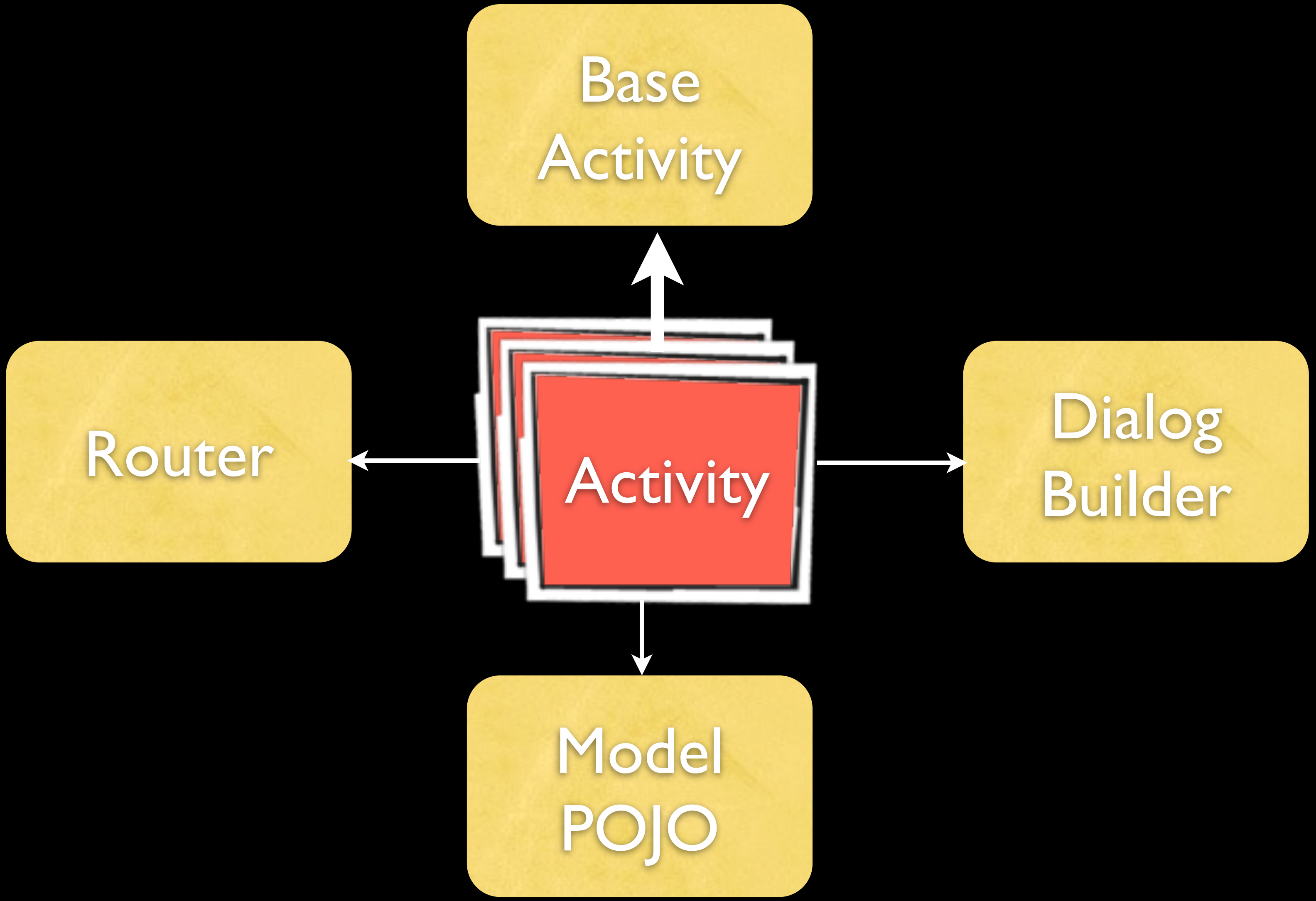
Application



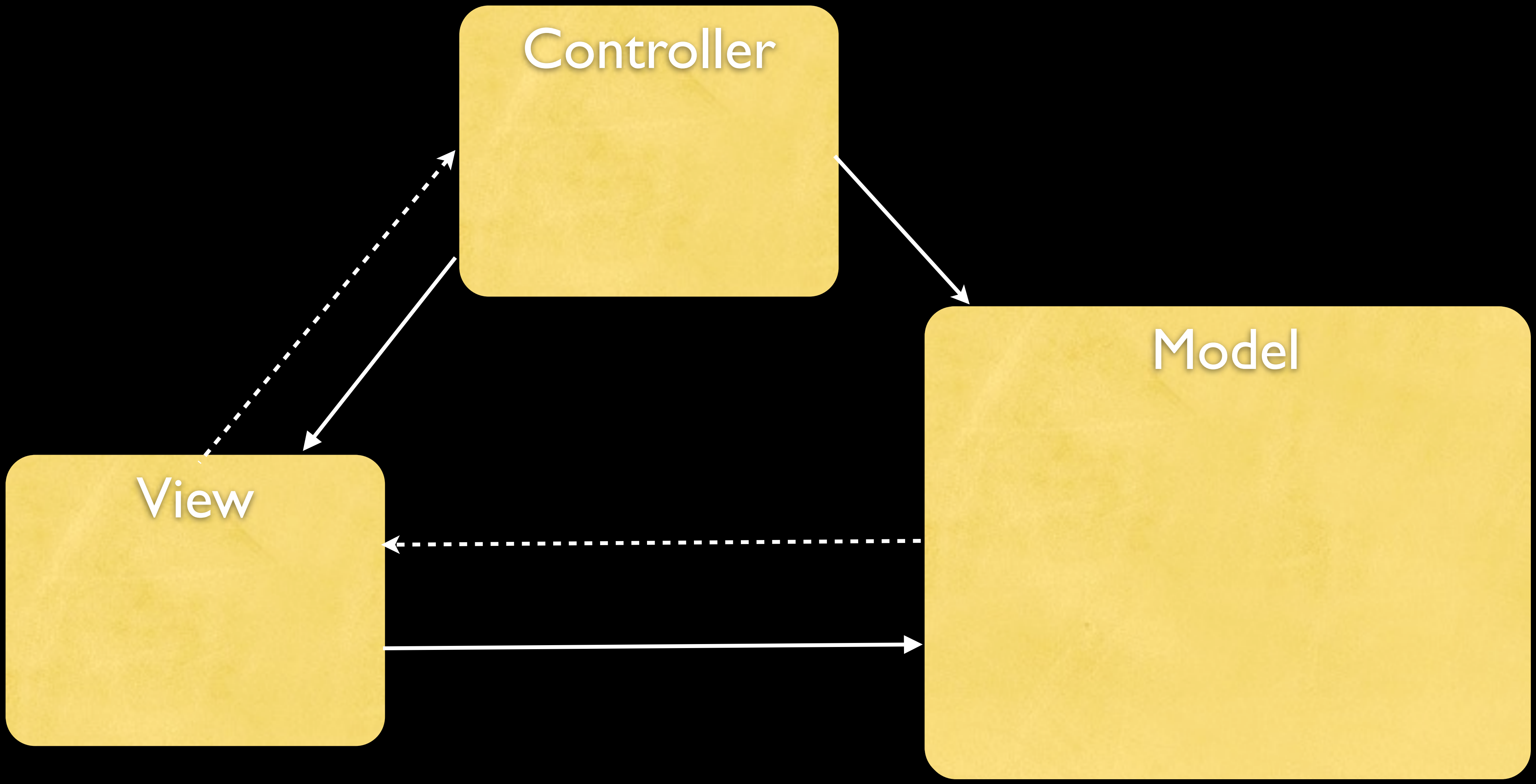
- Global state
  - Reference to domain model data
  - Accessible from all components

Activity

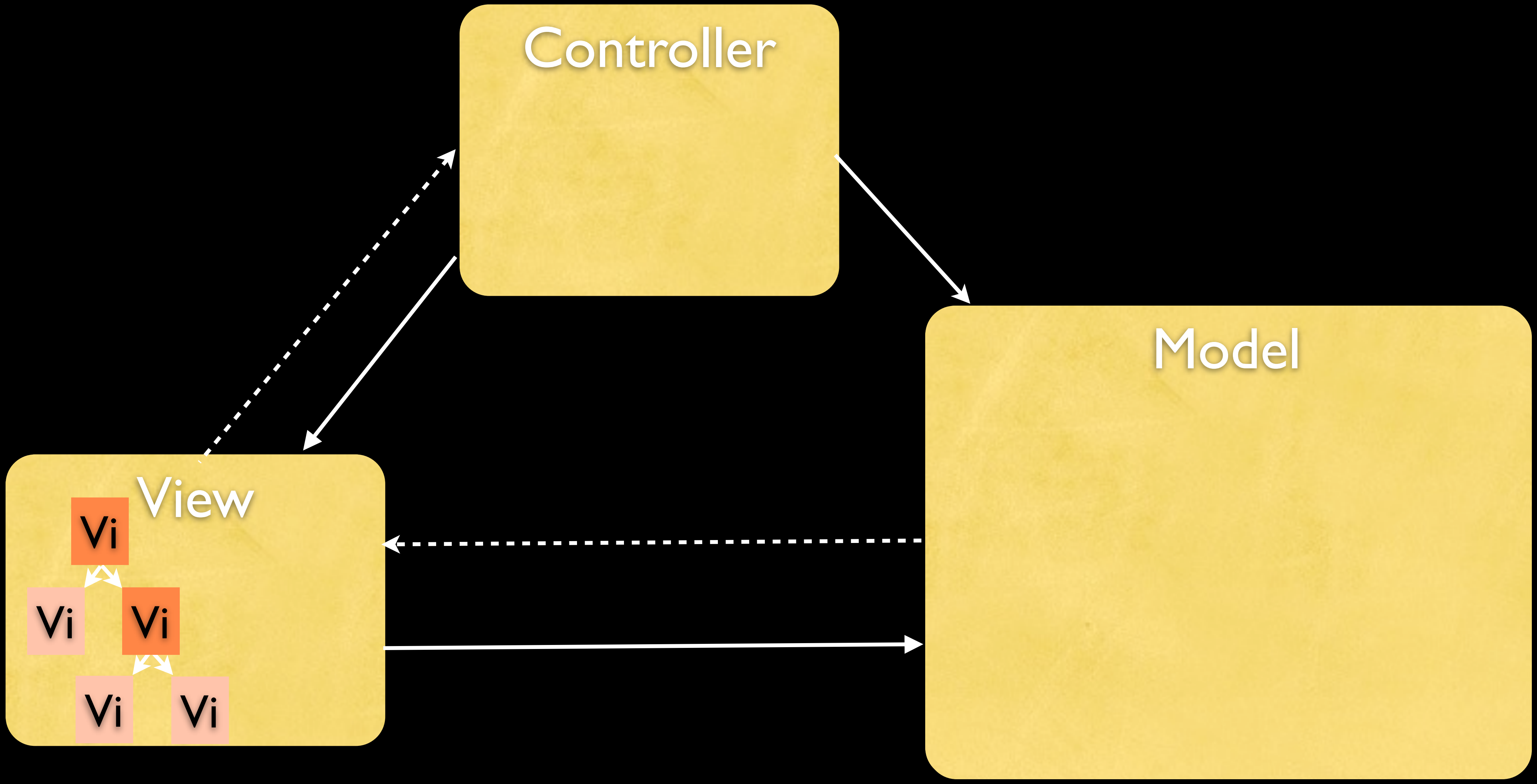
- Easily gets fat
- Move functionality out of the Activity



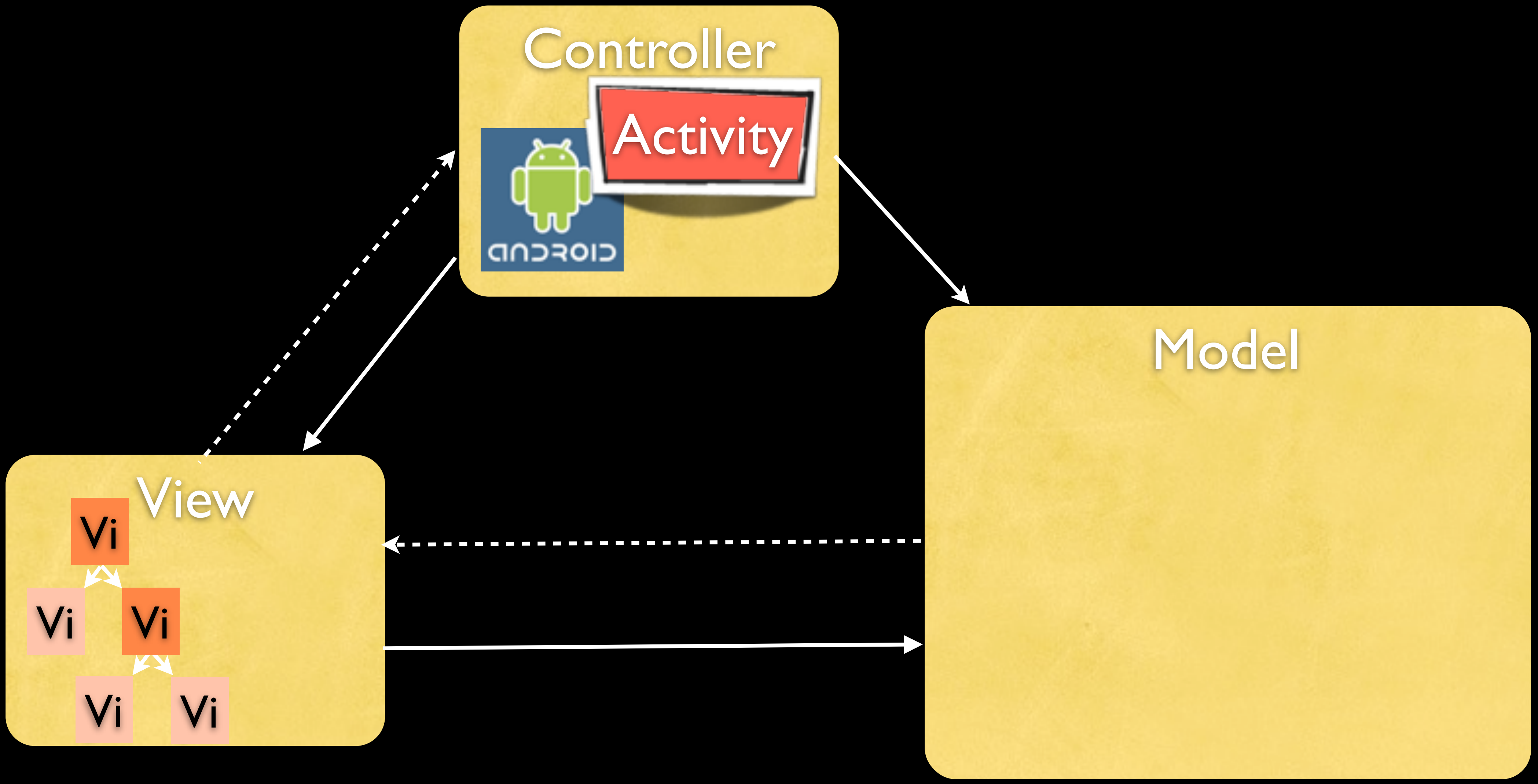
Android MVC



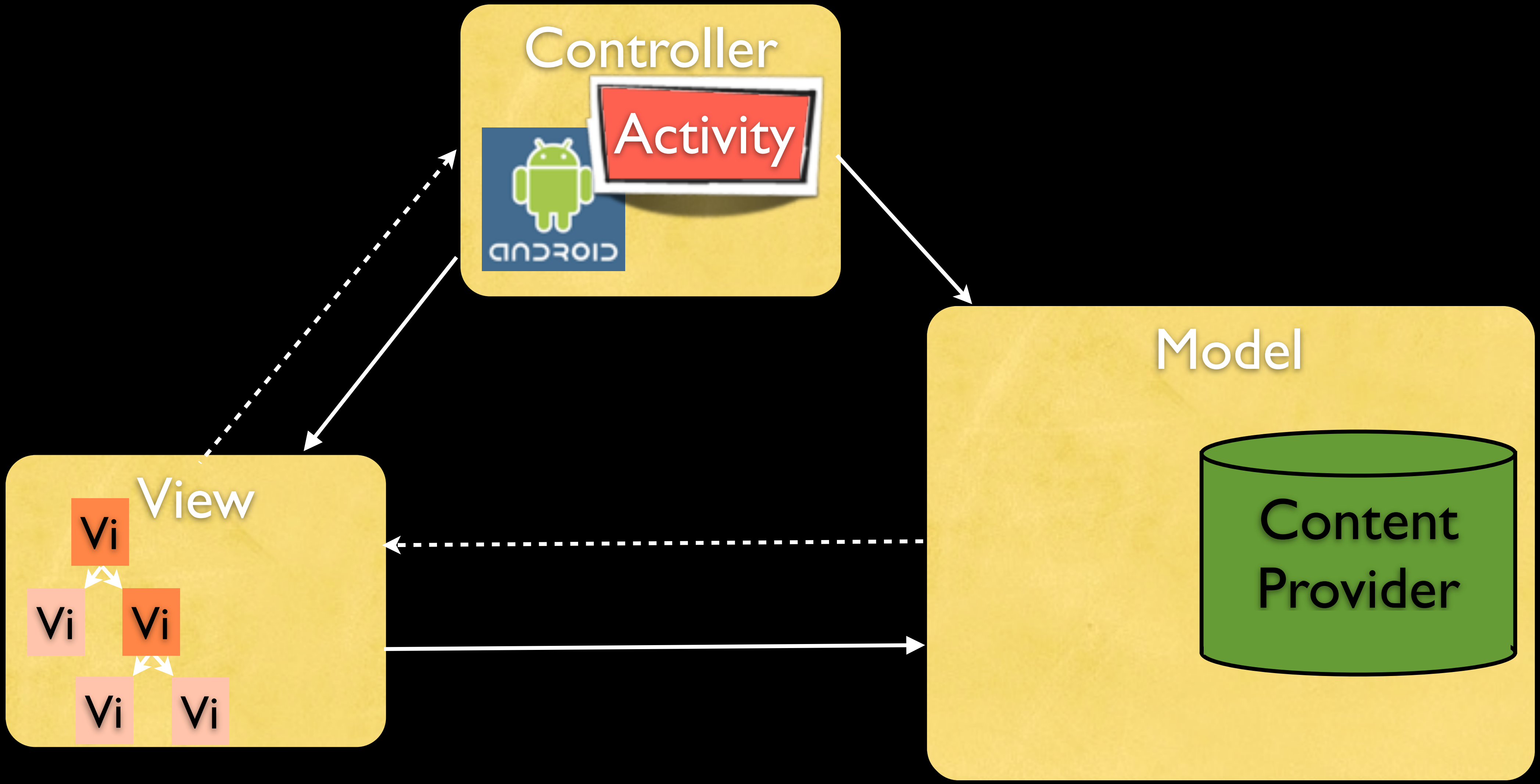
Android MVC



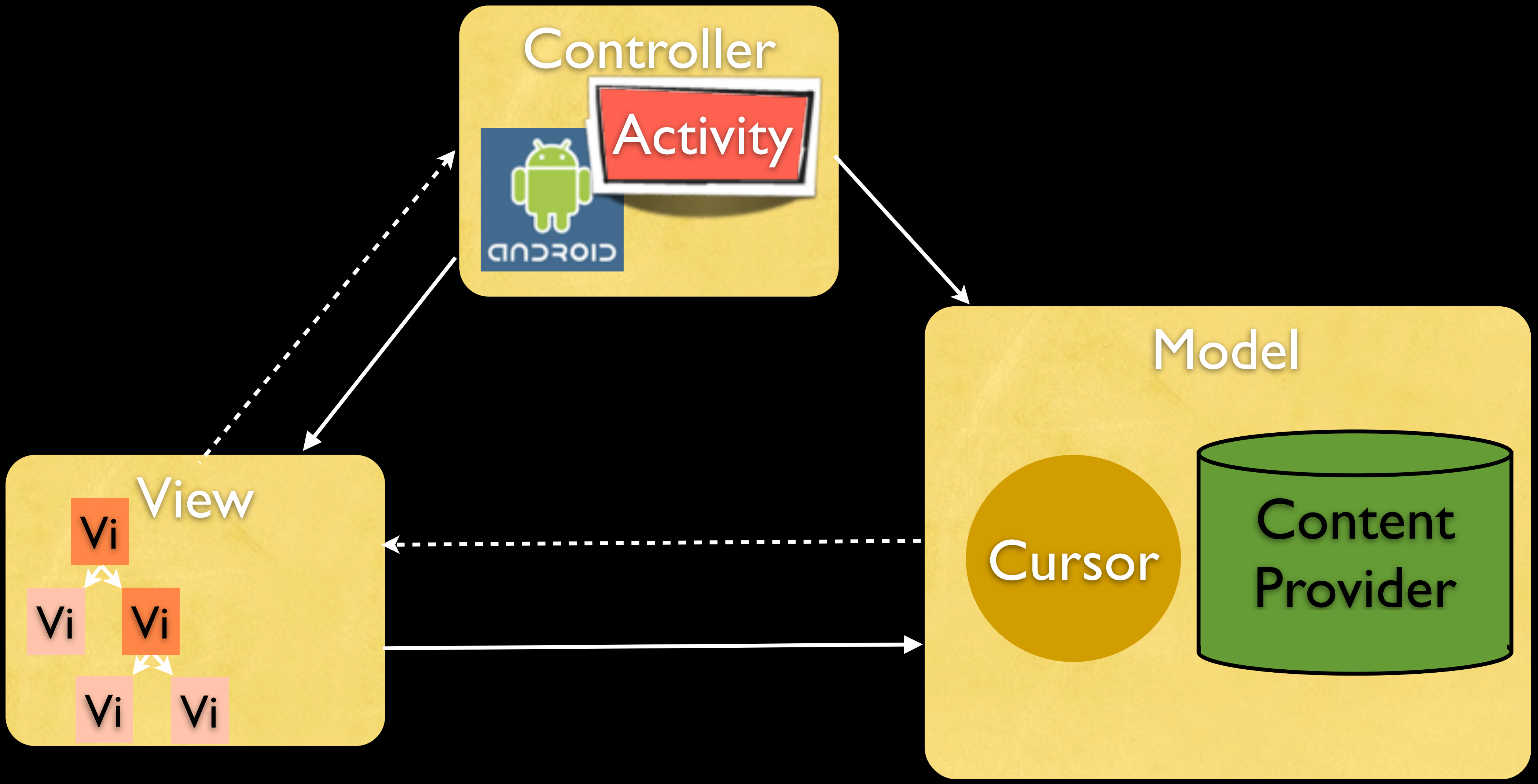
Android MVC



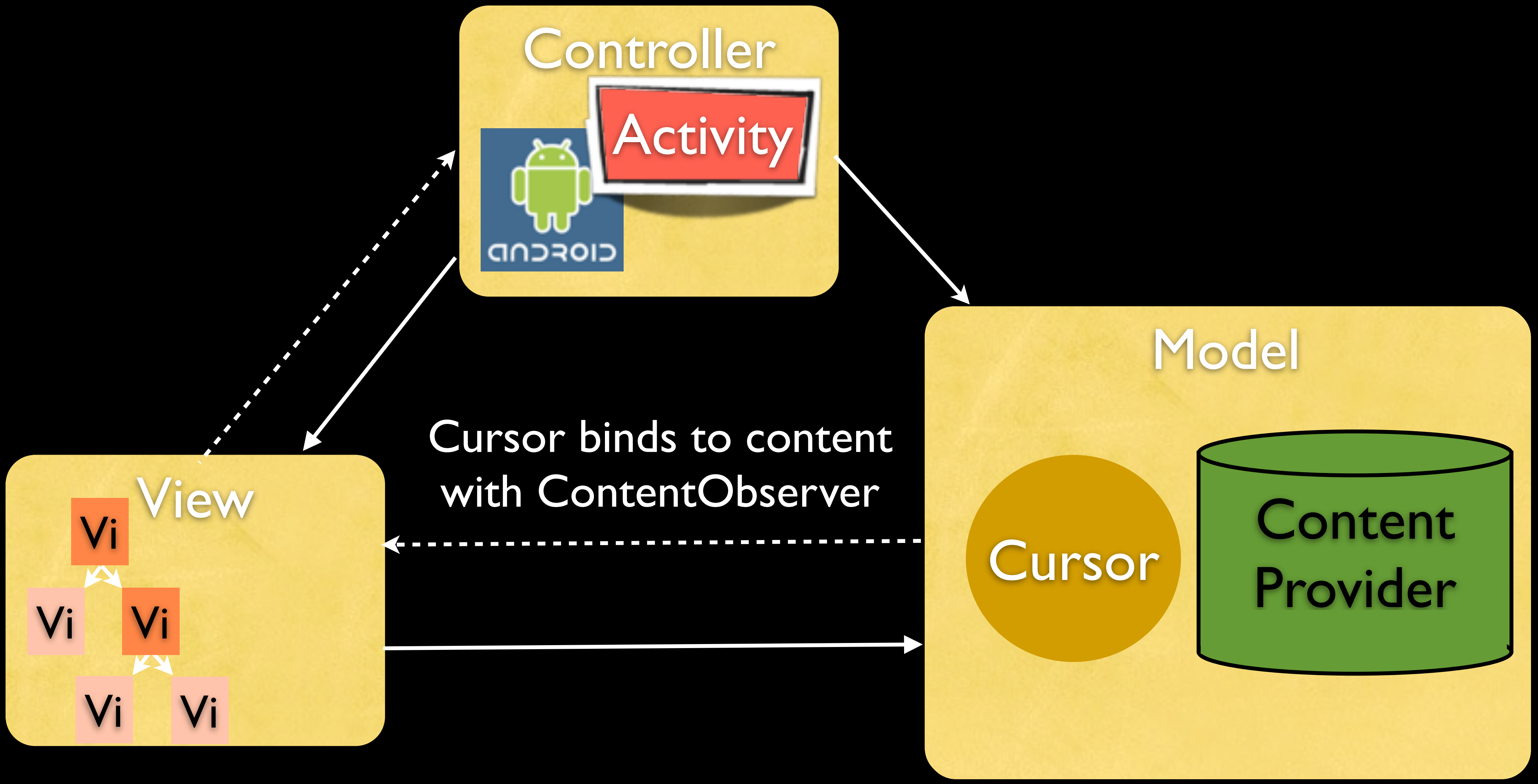
Android MVC


Controller
Activity
android
Model
Content Provider
View
Vi
Vi
Vi
Vi
Vi

Android MVC


Controller
Activity
android
Model
Cursor
Content Provider
View
Vi
Vi
Vi
Vi
Vi

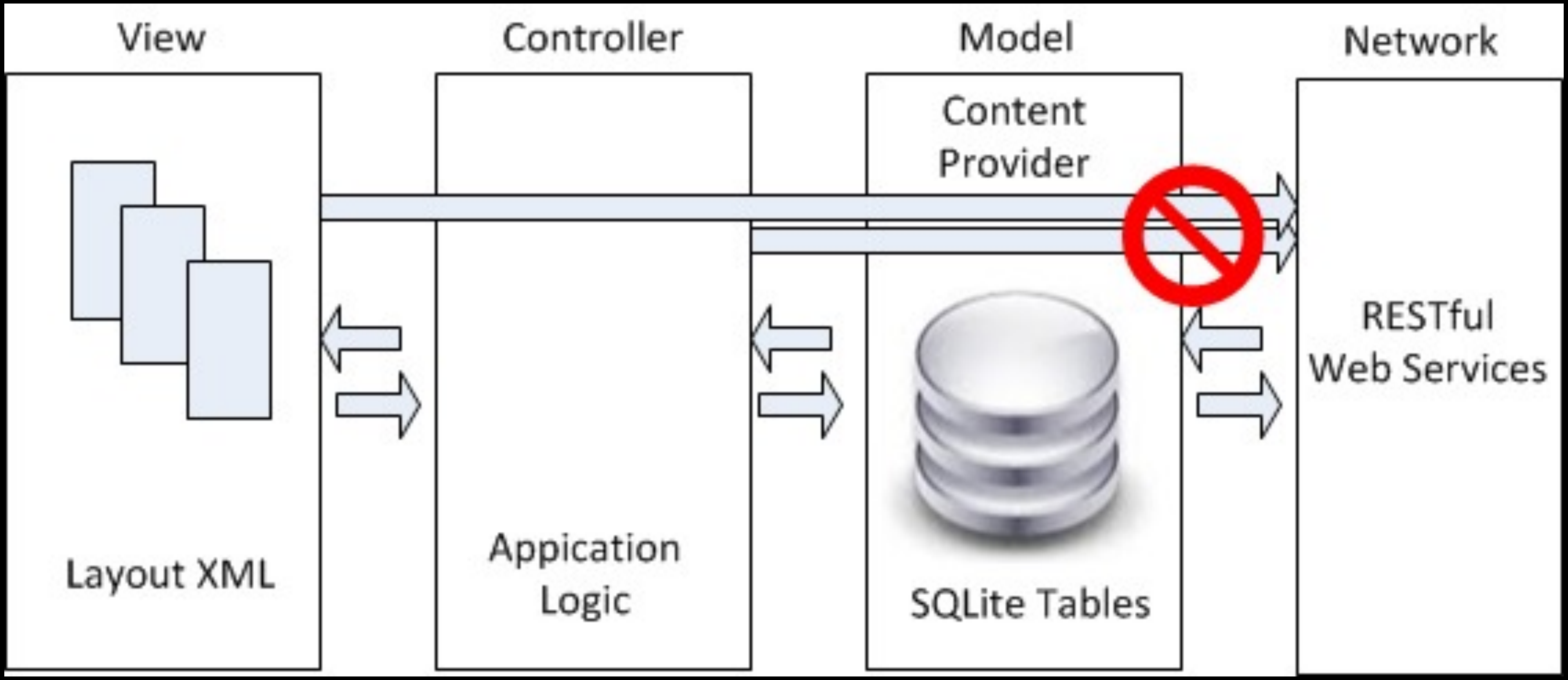
Android MVC



`ContentObserver.onChange()`

`getContext().getContentResolver().notifyChange()`

Android Network MVC

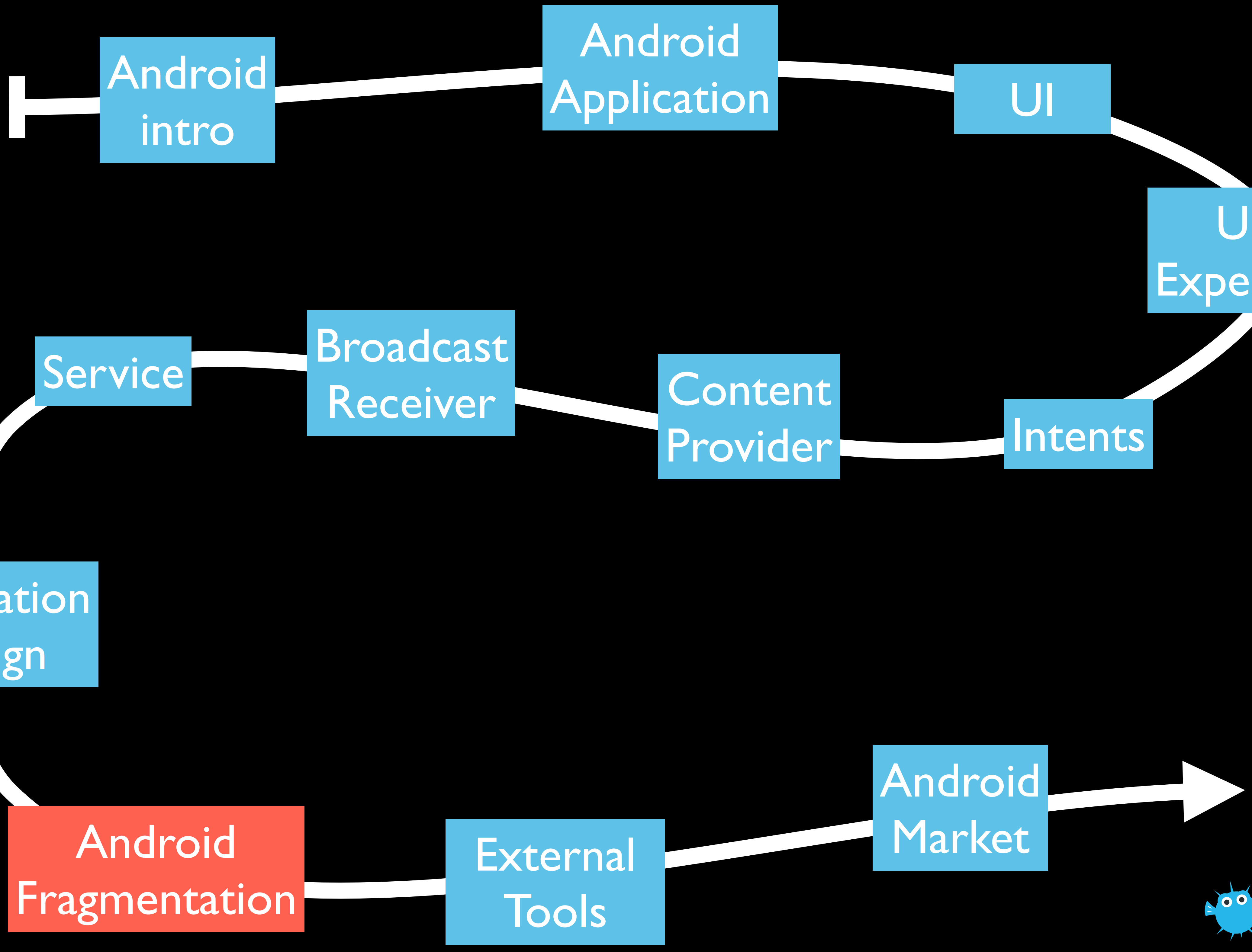


Google IO: http://www.google.com/events/io/2010/sessions/developing-RESTful-android-apps.html

Book: Programming Android, O'Reilly

JAYWAY

Agenda



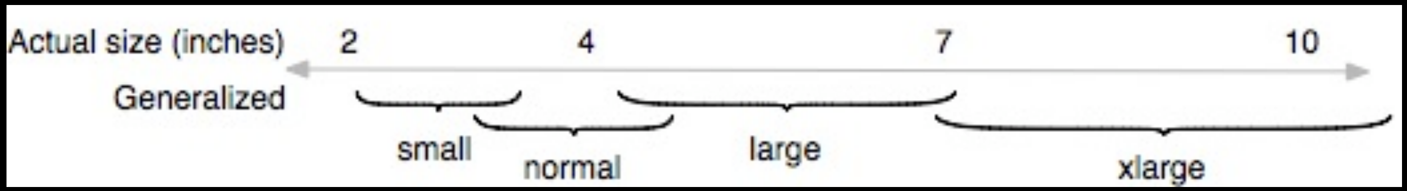
JAYWAY

Android Fragmentation

- Screen size
- Screen pixel density
- Manufacturers
    - Sony Ericsson, HTC, Samsung, LG, Motorola, etc.
- Keyboard
    - Touch, QWERTY, 12-key, etc.

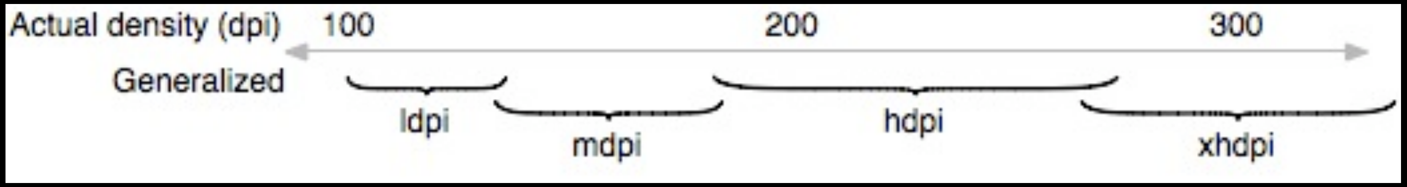
Screen Size

- Small
- Normal
- Large
- XLarge (2.2/Froyo)

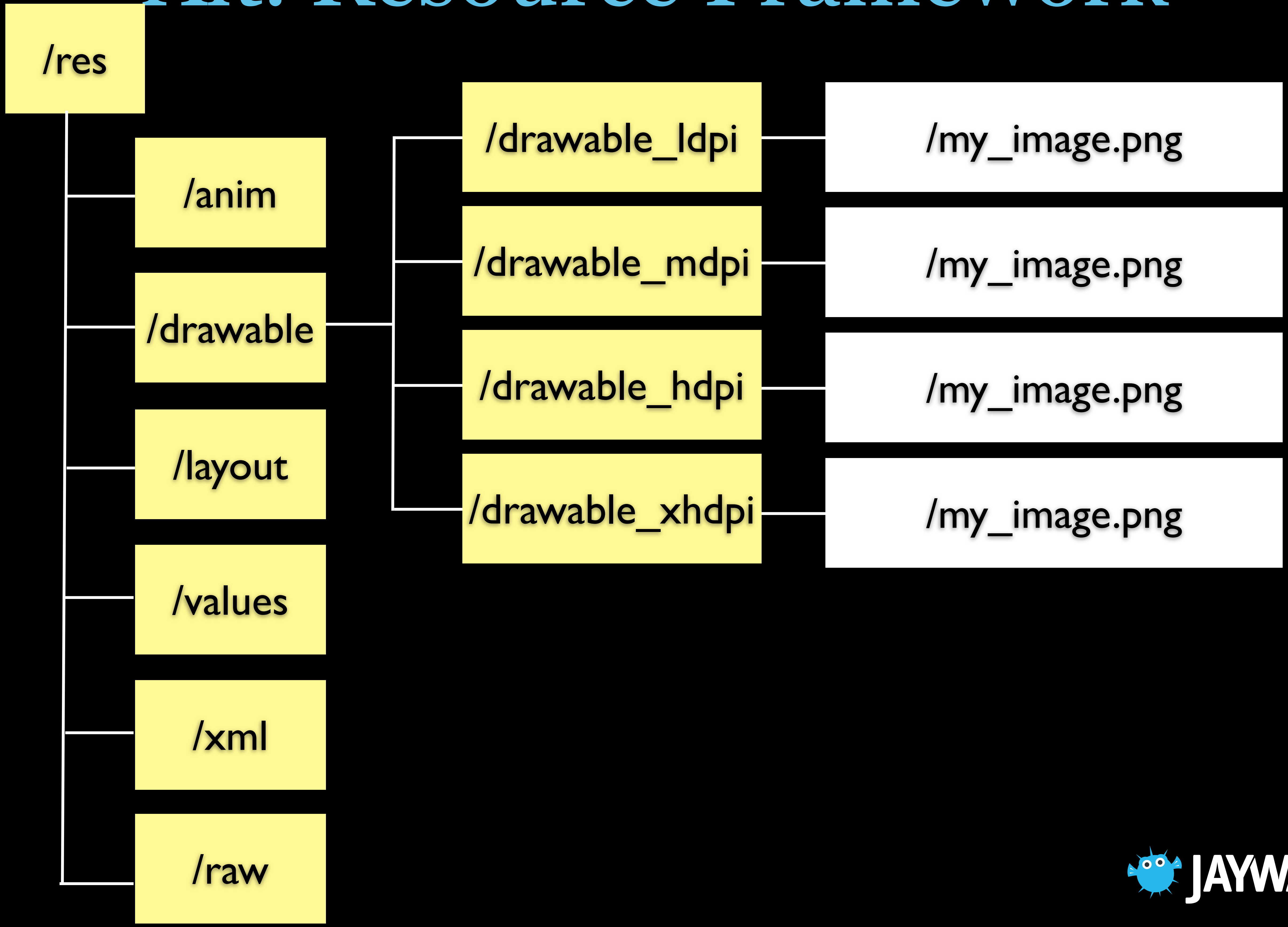


Screen Pixel Density

- Low (120dpi)
- Medium (160dpi)
- High (240dpi)
- Extra high (320dpi)



Alt. Resource Framework

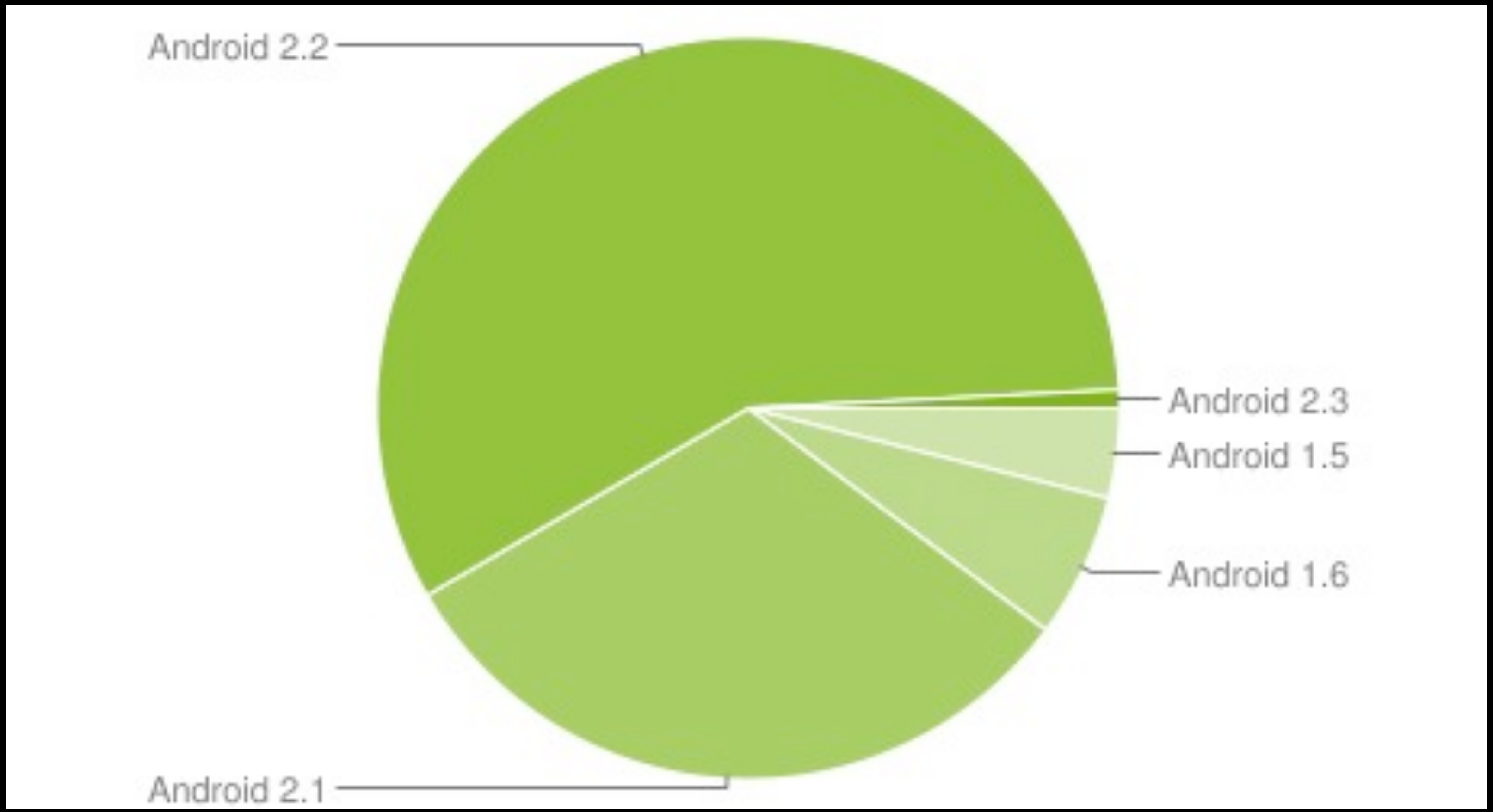


JAYWAY

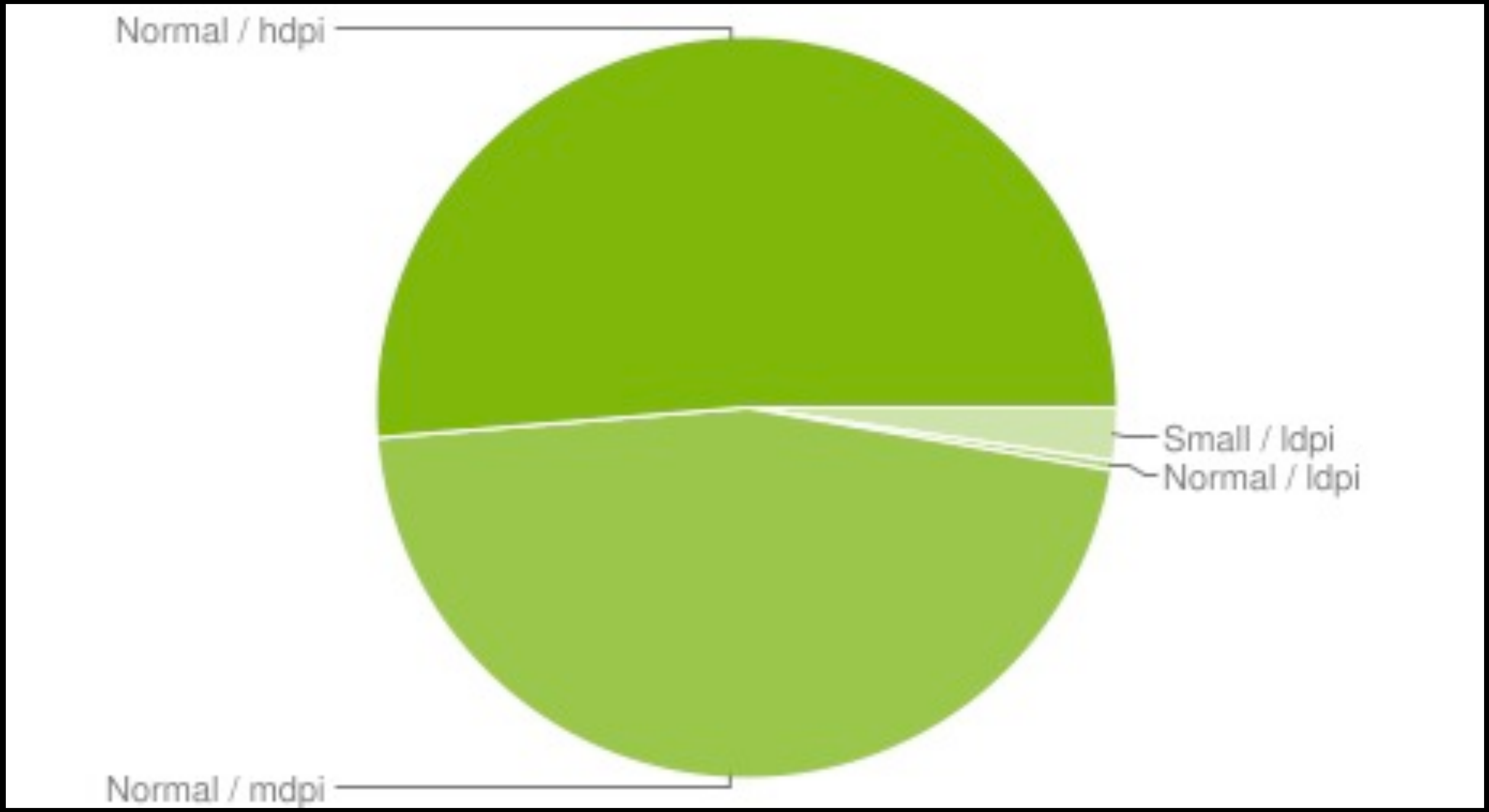
Recommendations

- Leave as much as possible to the platform to decide
    - wrap_content, match_parent
- Avoid defining absolute pixel values
    - “px”
- Use pixel dependent values
    - “dp”
    - “sp”

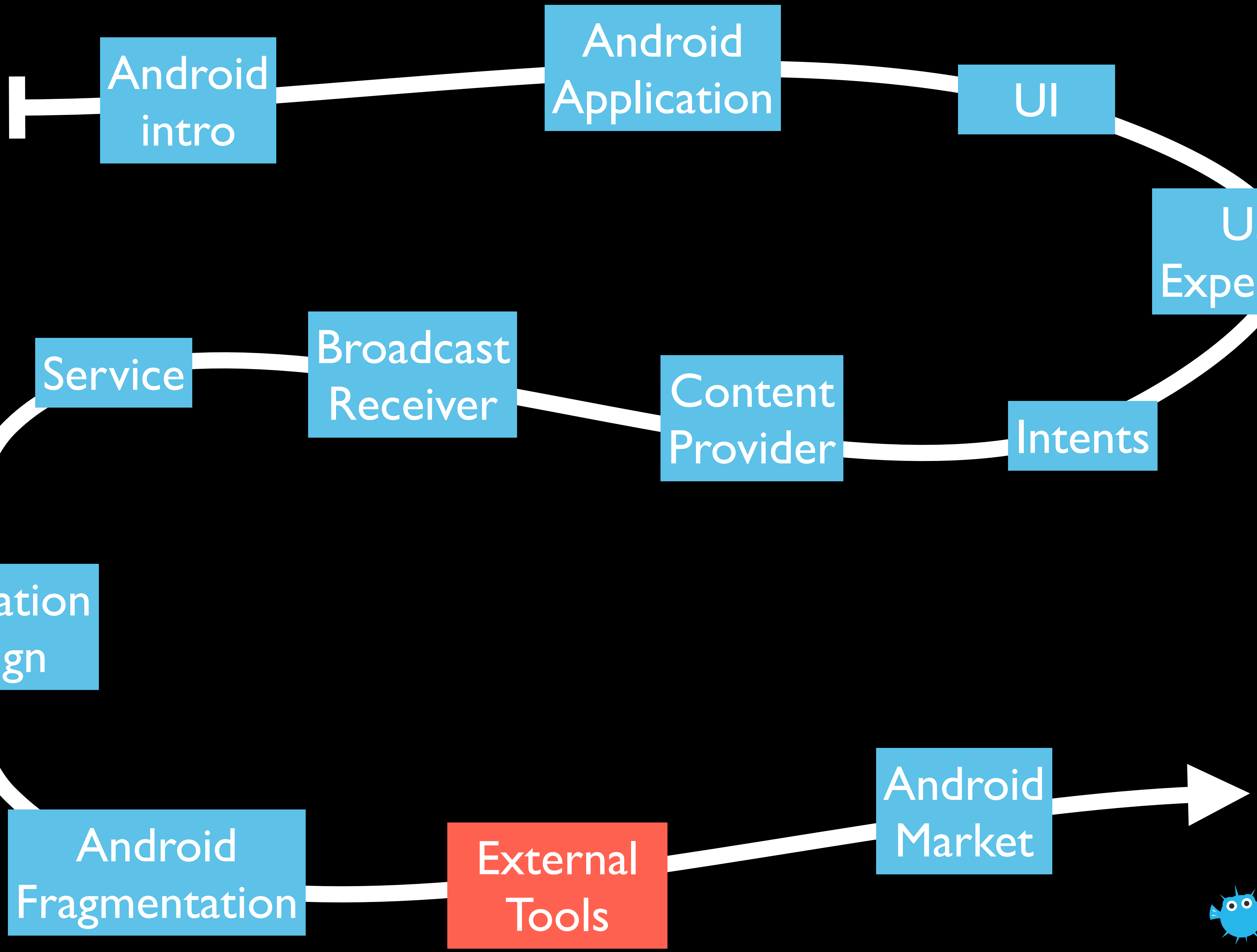
Platform versions



Screens



Agenda



JAYWAY

External Tools

- Large Android community
- Lot’s of good, helpful third party tools

Roboguice

- Dependency injection framework for Android
- http://code.google.com/p/roboguice/

```
class RoboWay extends RoboActivity {
    @InjectView(R.id.name)                  TextView name;
    @InjectView(R.id.thumbnail)             ImageView thumbnail;
    @InjectResource(R.drawable.icon)        Drawable icon;
    @InjectResource(R.string.app_name)      String myName;
    @Inject                                 LocationManager loc;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        name.setText( "Hello, " + myName );
    }
}
```

```
class AndroidWay extends Activity {
    TextView name;
    ImageView thumbnail;
    LocationManager loc;
    Drawable icon;
    String myName;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        name      = (TextView) findViewById(R.id.name);
        thumbnail = (ImageView) findViewById(R.id.thumbnail);
        loc       = (LocationManager) getSystemService(Activity.LOCATION_SERVICE);
        icon      = getResources().getDrawable(R.drawable.icon);
        myName    = getString(R.string.app_name);
        name.setText( "Hello, " + myName );
    }
}
```

Robolectric

- Unit test framework
- Runs tests on computer and not on device
- Tests are run in the computer JVM
- Makes mocking frameworks obsolete
- http://pivotal.github.com/robolectric/

Robotium

- Black box functional test framework
- Source code access not needed
- Write test scenarios spanning over multiple activities
- Open source
- http://code.google.com/p/robotium/

Google GSON

- Converts Java objects to their JSON representation

```
BagOfPrimitives obj = new BagOfPrimitives();
Gson gson = new Gson();
String json = gson.toJson(obj);
```

- Converts JSON representation to Java objects

```
BagOfPrimitives obj2 = gson.fromJson(json, BagOfPrimitives.class)
```

microlog4android

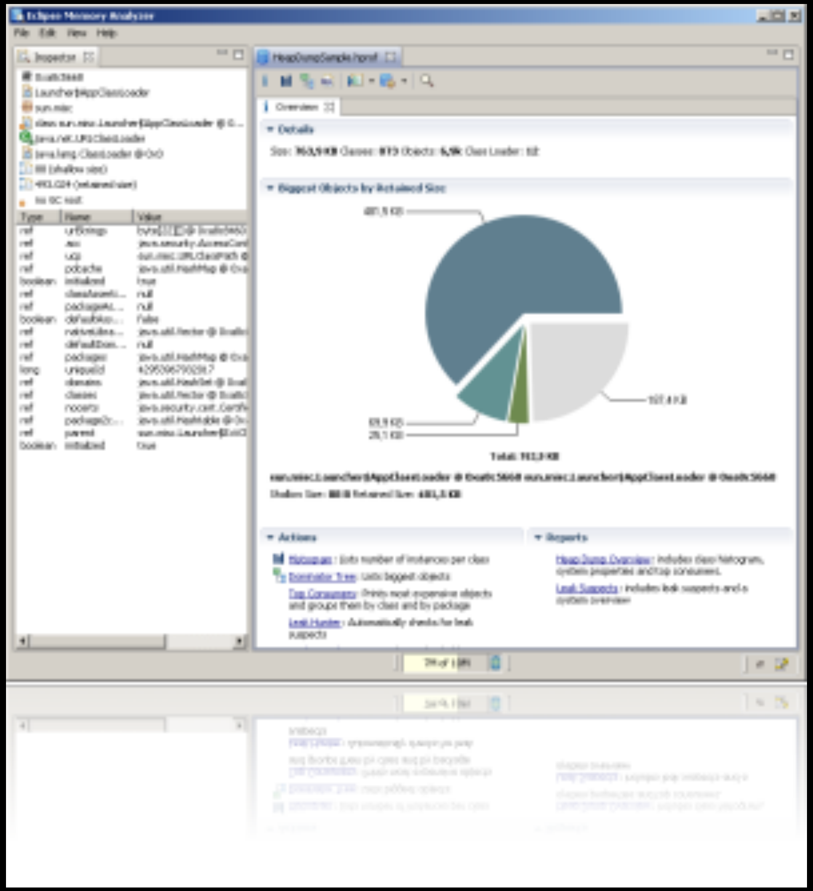
- Log4j API
- File logging (Internal/SD Card)
- http://code.google.com/p/microlog4android/

Maven

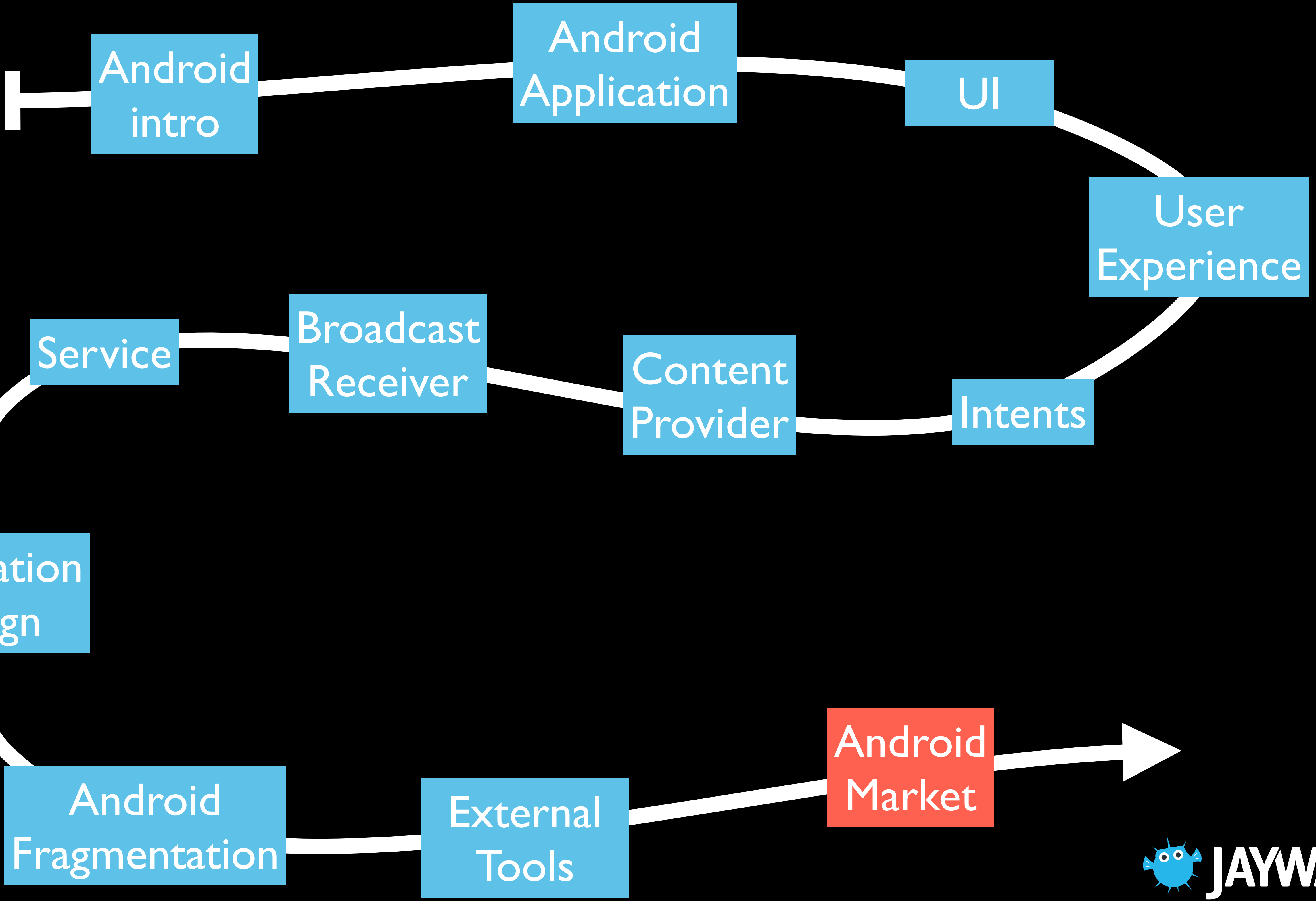
- Maven Android Plugin
- m2eclipse
- http://code.google.com/p/maven-android-plugin/

Memory Analyzer Tool

- Analyzes memory dumps
- Find memory leaks
- Eclipse plugin
- http://www.eclipse.org/mat/

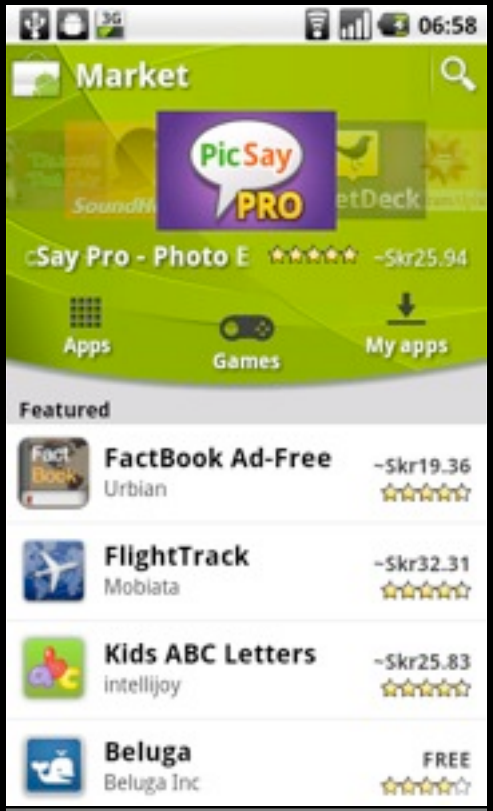


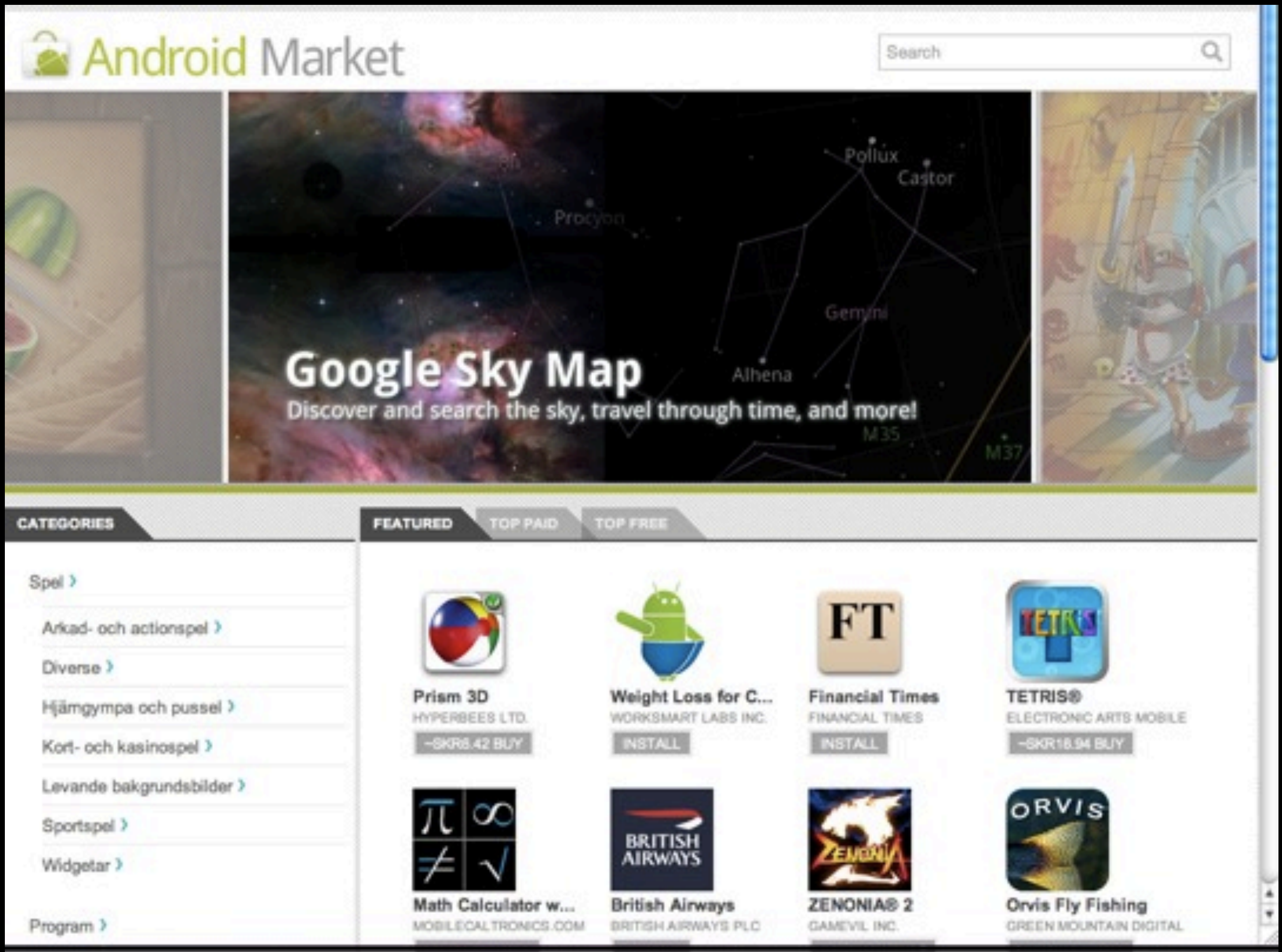
Agenda



JAYWAY

Android Market


06:58
Market
PicSay PRO
PicSay Pro - Photo E ~Skr25.94
Apps
Games
My apps
Featured
FactBook Ad-Free Urbian ~Skr19.36
FlightTrack Mobiata ~Skr32.31
Kids ABC Letters intellijoy ~Skr25.83
Beluga Beluga Inc FREE


Android Market
Search
Google Sky Map
Discover and search the sky, travel through time, and more!
CATEGORIES
FEATURED
TOP PAID
TOP FREE
Spel
Arkad- och actionspel
Diverse
Hjämgympa och pussel
Kort- och kasinospel
Levande bakgrundsbilder
Sportspel
Widgetar
Program
Prism 3D HYPERBEES LTD. ~SKR6.42 BUY
Weight Loss for C... WORKSMART LABS INC. INSTALL
Financial Times FINANCIAL TIMES INSTALL
TETRIS® ELECTRONIC ARTS MOBILE ~SKR18.94 BUY
Math Calculator w... MOBILECALTRONICS.COM
British Airways BRITISH AIRWAYS PLC
ZENONIA® 2 GAMEVIL INC.
Orvis Fly Fishing GREEN MOUNTAIN DIGITAL

Market Filters

- Restricts what Apps are shown to user in Market
- Filter
  - Device type
  - Publishing status
  - Priced status
  - Native platform
  - Forward lock

JAYWAY

Market Filters

- Restricts what Apps are shown to user in Market
- Filter
  - Device type
  - Publishing status
  - Priced status
  - Native platform
  - Forward lock

Device Type

↔

AndroidManifest.xml

<supports-screens>

<uses-configuration>

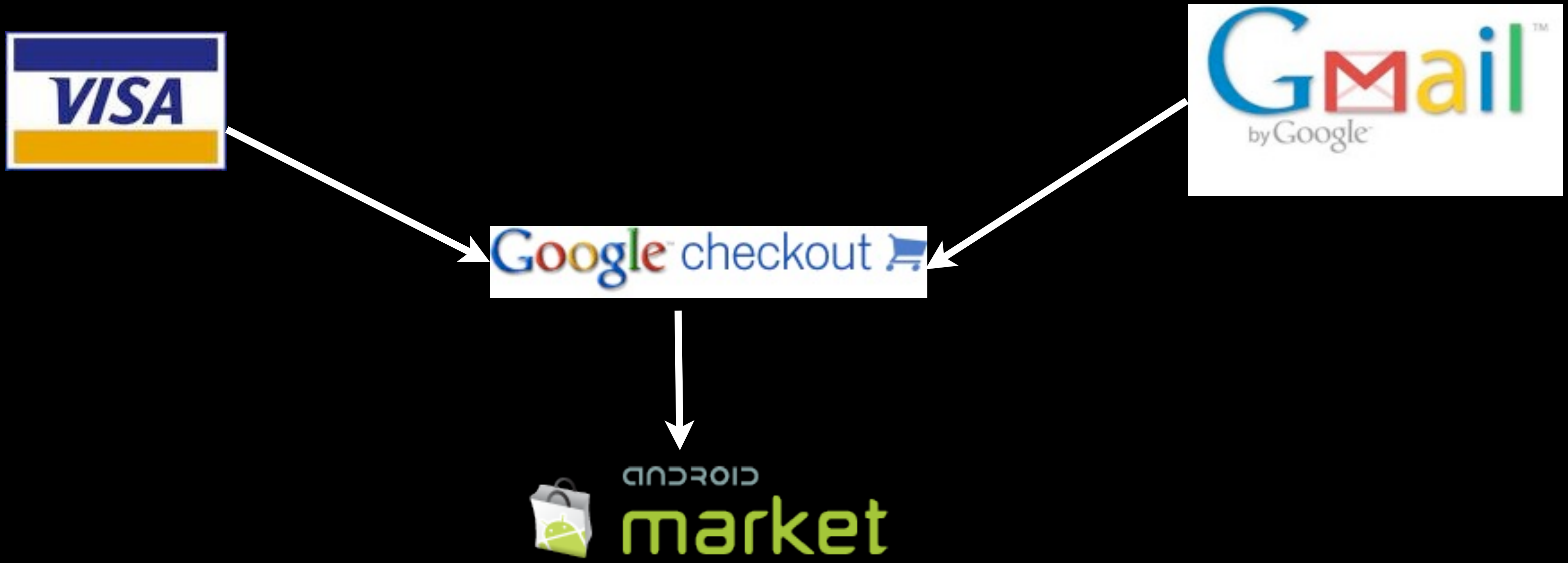
<uses-feature>

<uses-library>

<uses-sdk>

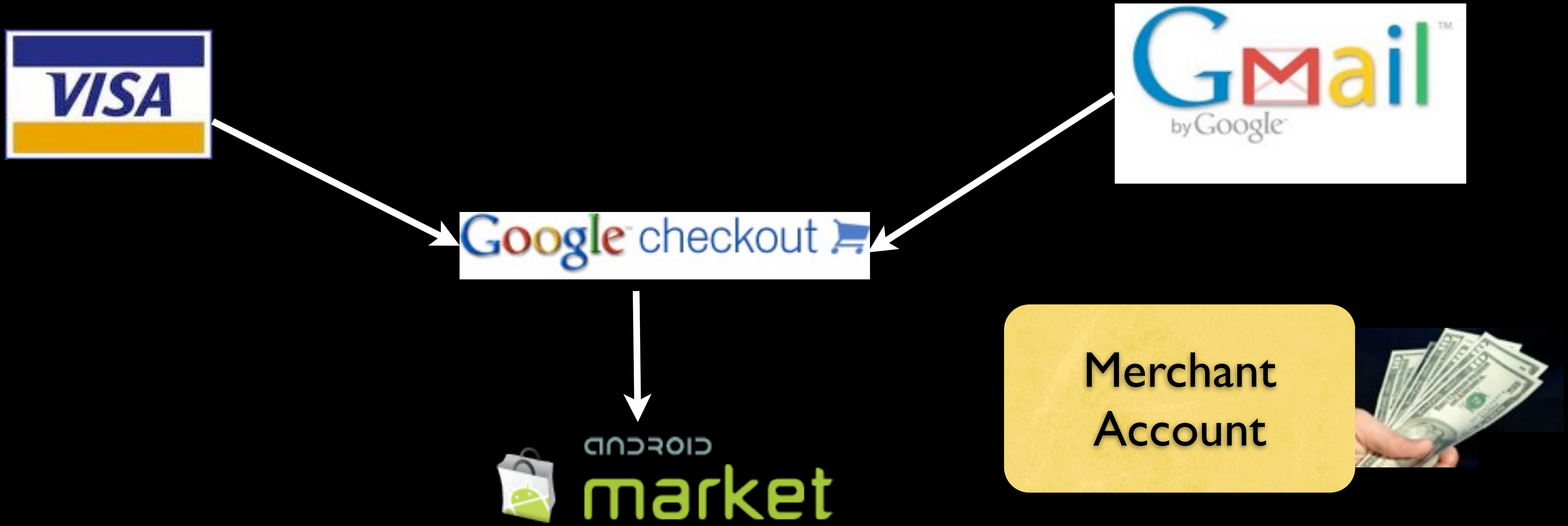
Register

- One time fee: $25



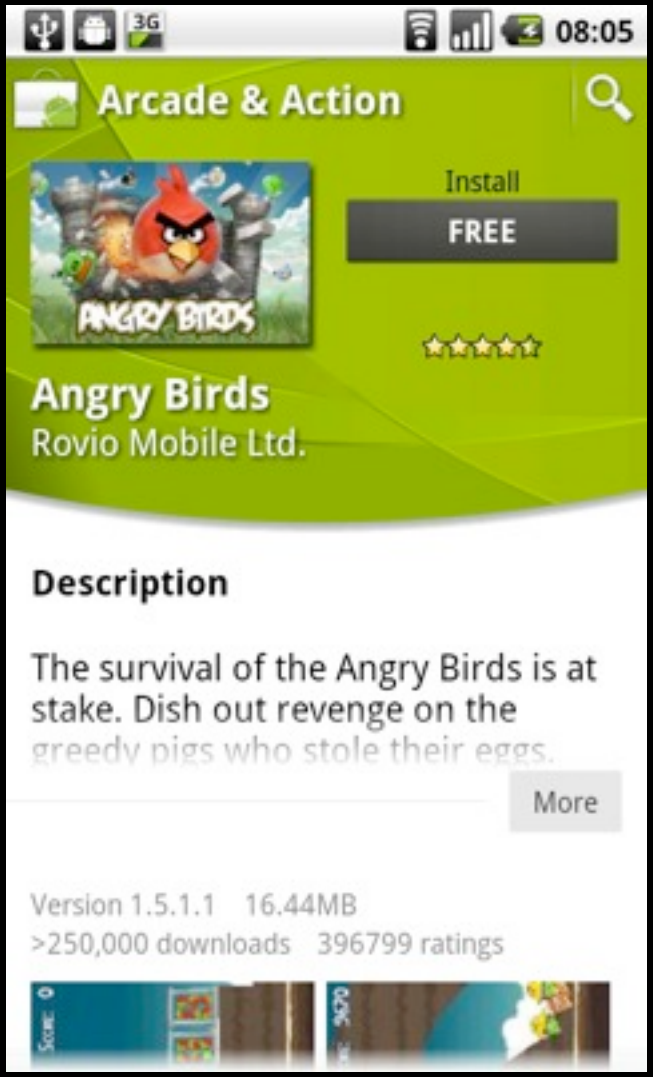
Register

- One time fee: $25



Upload

- APK (< 50 MB)
- Screenshots (2 Required + 6 Optional)
- Application Icon (Required)
- Promotional video (Optional)
- Listing details
- Contact information (One of WWW, Mail, Phone)



Angry Birds
Rovio Mobile Ltd.

Description

The survival of the Angry Birds is at stake. Dish out revenge on the greedy pigs who stole their eggs.

Version 1.5.1.1 16.44MB
>250,000 downloads 396799 ratings

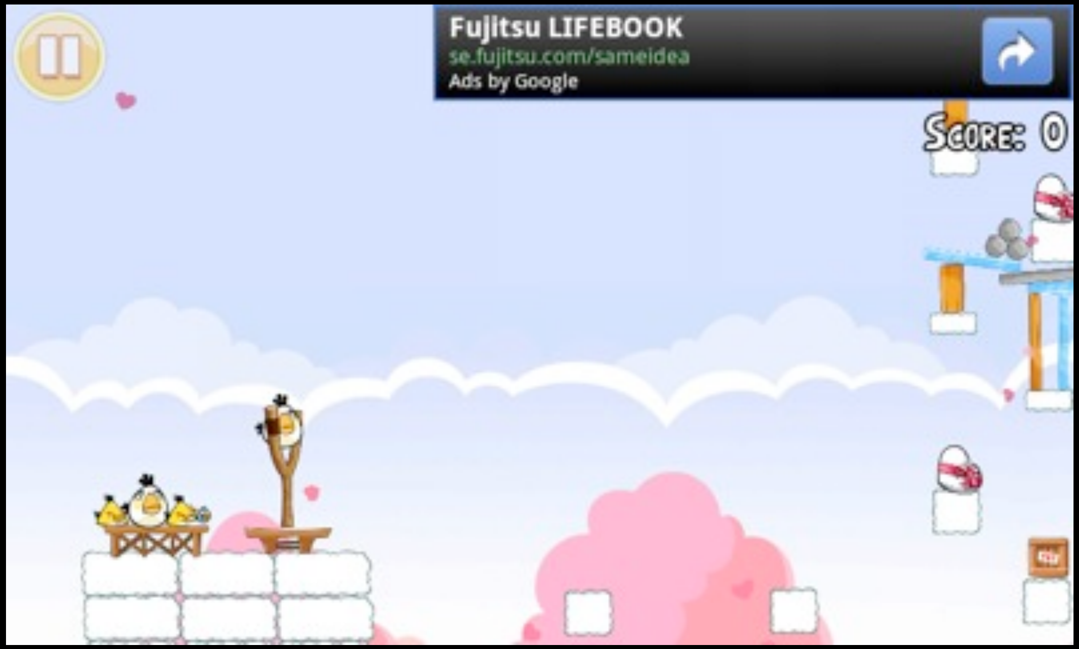
Paid Apps

Argentina*
- Australia
- Austria
- Belgium
- Brazil*
- Canada
- Denmark
- Finland
- France
- Germany
- Hong Kong
- Ireland
- Israel*
- Italy
- Japan
- Mexico*
- Netherlands
- New Zealand
- Norway
- Portugal
- Russia*
- Singapore
- Spain
- South Korea*
- Sweden
- Switzerland
- Taiwan*
- United Kingdom
- United States

- Transaction fee: 30%
- Pricing
  - SEK: 7 SEK - 1500 SEK

Advertising

- AdMob
  - AdSense content
- Add market-url
  - market://<your_app_url>



AdMob SDK

- `admob-sdk-android.jar`

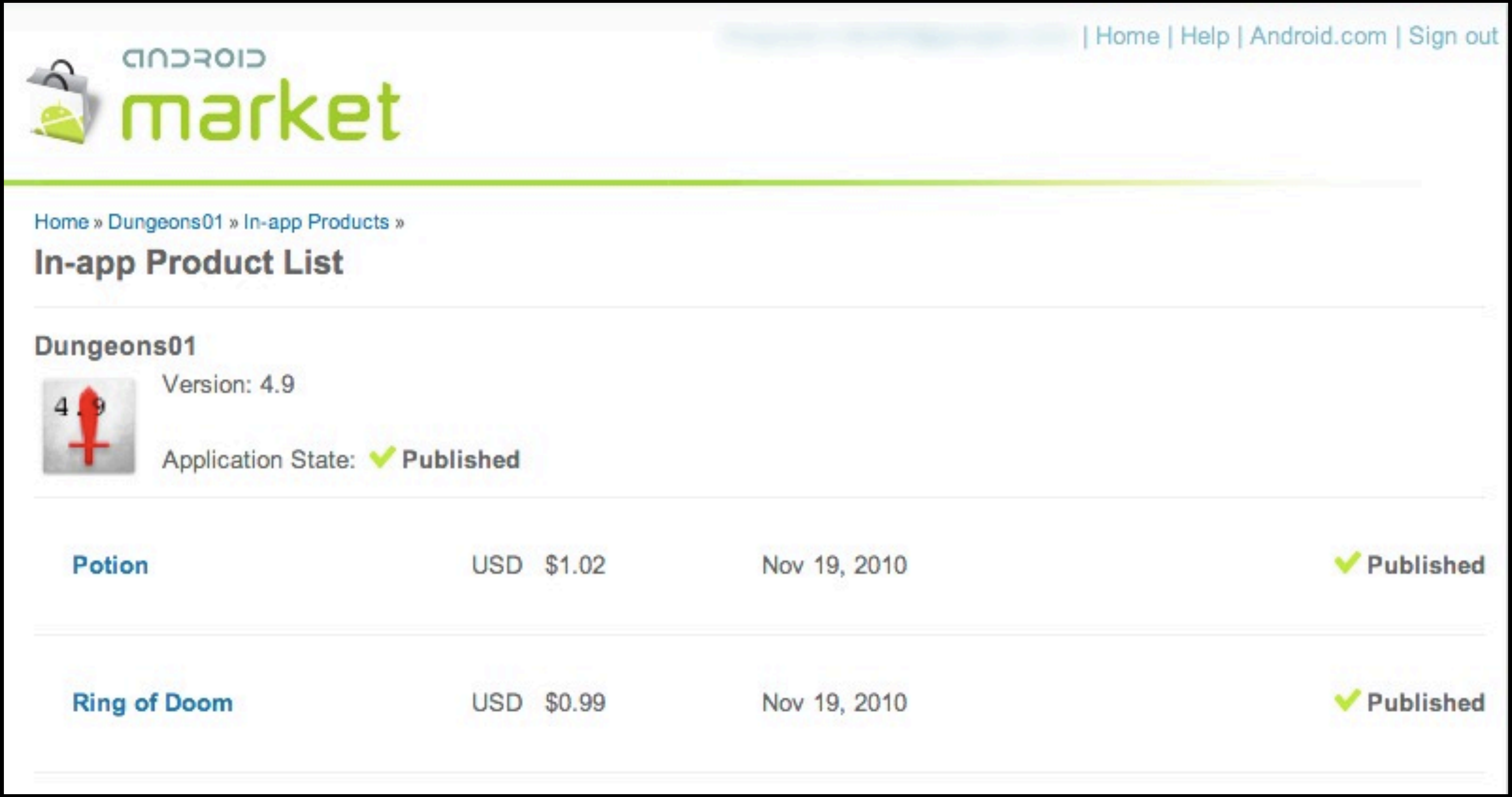
```
<!-- The application's publisher ID assigned by AdMob -->
<meta-data android:value="YOUR_ID_HERE" android:name="ADMOB_PUBLISHER_ID" />

<!-- AdMobActivity definition -->
   <activity android:name="com.admob.android.ads.AdMobActivity"
   android:theme="@android:style/Theme.NoTitleBar.Fullscreen"
   android:configChanges="orientation|keyboard|keyboardHidden" />

   <!-- Track Market installs -->
   <receiver android:name="com.admob.android.ads.analytics.InstallReceiver"
   android:exported="true">

      <intent-filter> <action android:name="com.android.vending.INSTALL_REFERRER" />
      </intent-filter>
   </receiver>
```

In-App Billing

- Sell content in application
  - Android Market publisher account
  - Google Checkout merchant account
  - No Android API
- Android Market handles the transaction

JAYWAY

In-App Billing


| Home | Help | Android.com | Sign out
android market
Home » Dungeons01 » In-app Products »

In-app Product List

Dungeons01

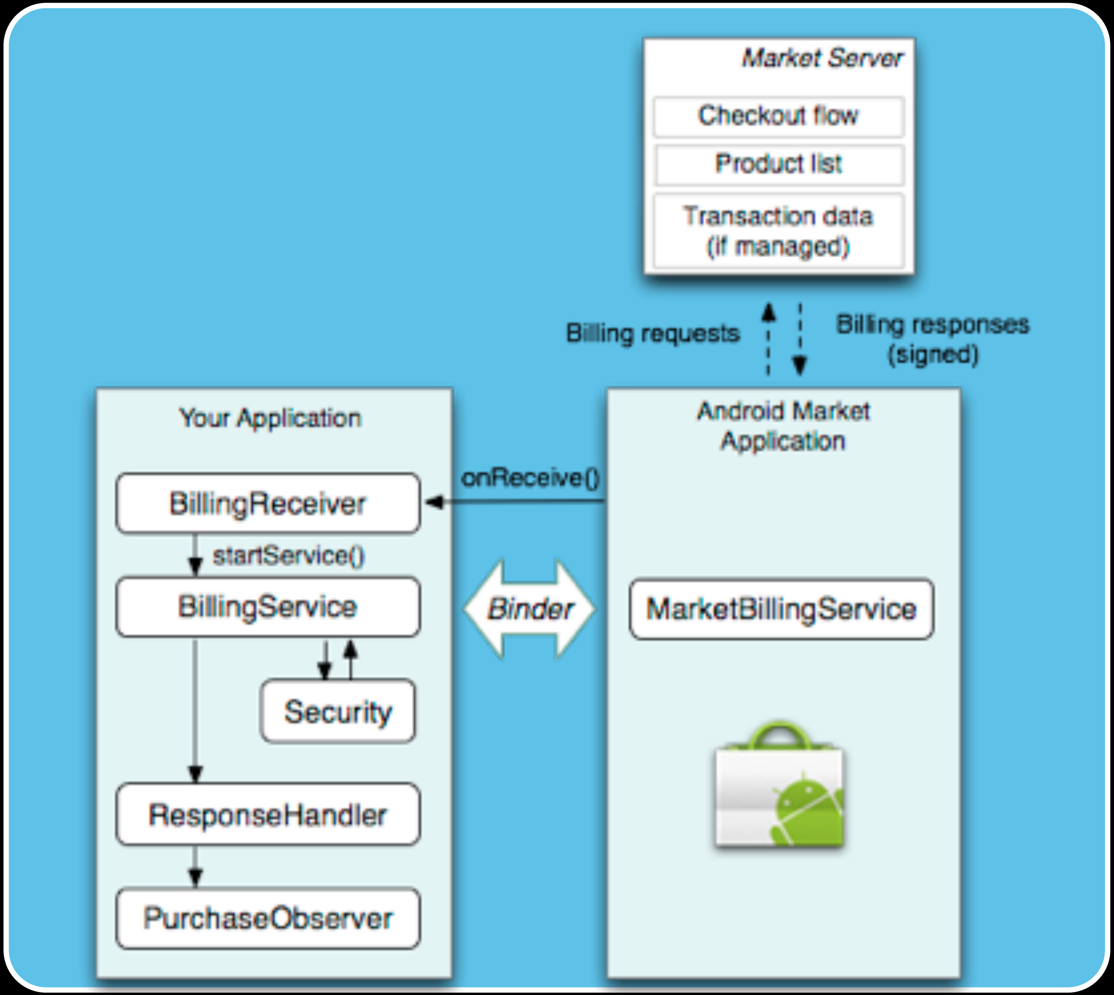
Version: 4.9

Application State: ✓ Published

Potion	USD	$1.02	Nov 19, 2010	✓ Published
Ring of Doom	USD	$0.99	Nov 19, 2010	✓ Published

Implementation overview

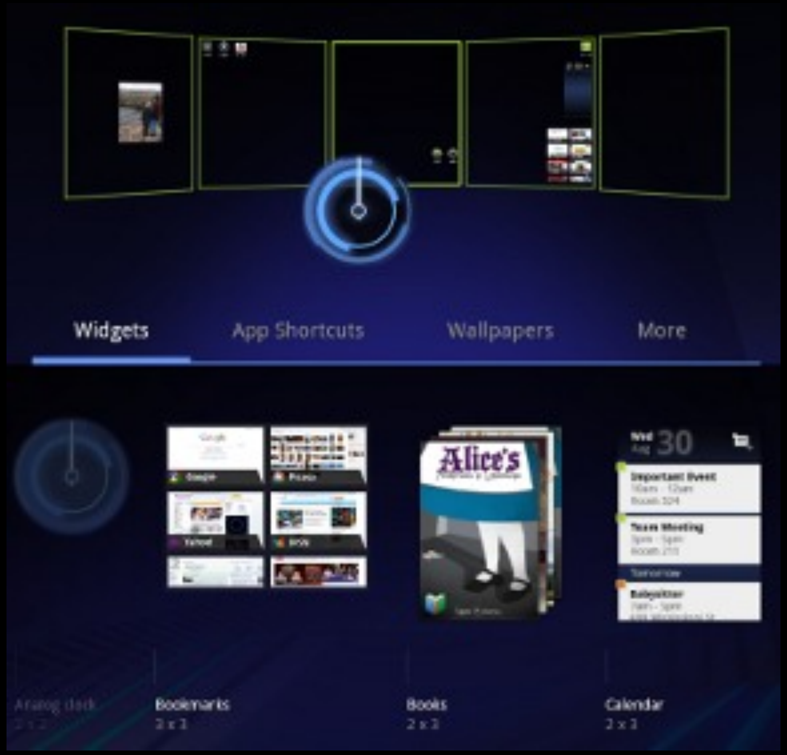
- Market-app exposes API
  - `IMarketBillingService.aidl`
- Service
  - Send info to Market-app
- BroadcastReceiver
  - Billing info from Market-app
  - Response codes
- Security-component
  - Integrity of transaction



JAYWAY

Android - What are we waiting for?

- 3.0
- Tablet version of Android
- New UI
- Richer widgets
- New application set
- Fragments API



Thank you for listening!

JAYWAY