Intent

https://developer.android.com/reference/android/content/Intent

Cont ...

- An intent is an abstract description of an operation to be performed.
- It can be used with startActivity to launch an Activity,
- broadcastIntent to send it to any interested BroadcastReceiver components, and
- Context.startService(Intent) or Context.bindService(Intent, ServiceConnection, int) to communicate with a background Service.

Cont ...

- An Intent provides a facility for performing late runtime binding between the code in different applications.
- Its most significant use is in the launching of activities, where it can be thought of as the glue between activities.
- It is basically a passive data structure holding an abstract description of an action to be performed.

Intent Structure

- The primary pieces of information in an intent are:
  - action -- The general action to be performed, such as ACTION_VIEW, ACTION_EDIT, ACTION_MAIN, etc.
  - data -- The data to operate on, such as a person record in the contacts database, expressed as a Uri.
- In addition to these primary attributes, there are a number of secondary attributes that you can also include with an intent:

Cont ...

- category -- Gives additional information about the action to execute. For example, CATEGORY_LAUNCHER means it should appear in the Launcher as a top-level application, while CATEGORY_ALTERNATIVE means it should be included in a list of alternative actions the user can perform on a piece of data.
- type -- Specifies an explicit type (a MIME type) of the intent data. Normally the type is inferred from the data itself. By setting this attribute, you disable that evaluation and force an explicit type.

Cont ...

- component -- Specifies an explicit name of a component class to use for the intent. Normally this is determined by looking at the other information in the intent (the action, data/type, and categories) and matching that with a component that can handle it. If this attribute is set then none of the evaluation is performed, and this component is used exactly as is. By specifying this attribute, all of the other Intent attributes become optional.
- extras -- This is a Bundle of any additional information. This can be used to provide extended information to the component. For example, if we have a action to send an e-mail message, we could also include extra pieces of data here to supply a subject, body, etc.

Cont ...

- Put together, the set of actions, data types, categories, and extra data defines a language for the system allowing for the expression of phrases such as "call john smith's cell".
- As applications are added to the system, they can extend this language by adding new actions, types, and categories, or they can modify the behavior of existing phrases by supplying their own activities that handle them.

Intent Resolution

- There are two primary forms of intents you will use.
  - Explicit Intents have specified a component (via setComponent(ComponentName) or setClass(Context, Class)), which provides the exact class to be run. Often these will not include any other information, simply being a way for an application to launch various internal activities it has as the user interacts with the application.
  - Implicit Intents have not specified a component; instead, they must include enough information for the system to determine which of the available components is best to run for that intent.

Cont ...

- When using implicit intents, given such an arbitrary intent we need to know what to do with it.
- This is handled by the process of Intent resolution, which maps an Intent to an Activity, BroadcastReceiver, or Service (or sometimes two or more activities/receivers) that can handle it.

Cont ...

- The intent resolution mechanism basically revolves around matching an Intent against all of the <intent-filter> descriptions in the installed application packages.
- (Plus, in the case of broadcasts, any BroadcastReceiver objects explicitly registered with Context.registerReceiver(BroadcastReceiver, IntentFilter).) More details on this can be found in the documentation on the IntentFilter class.

Cont ...

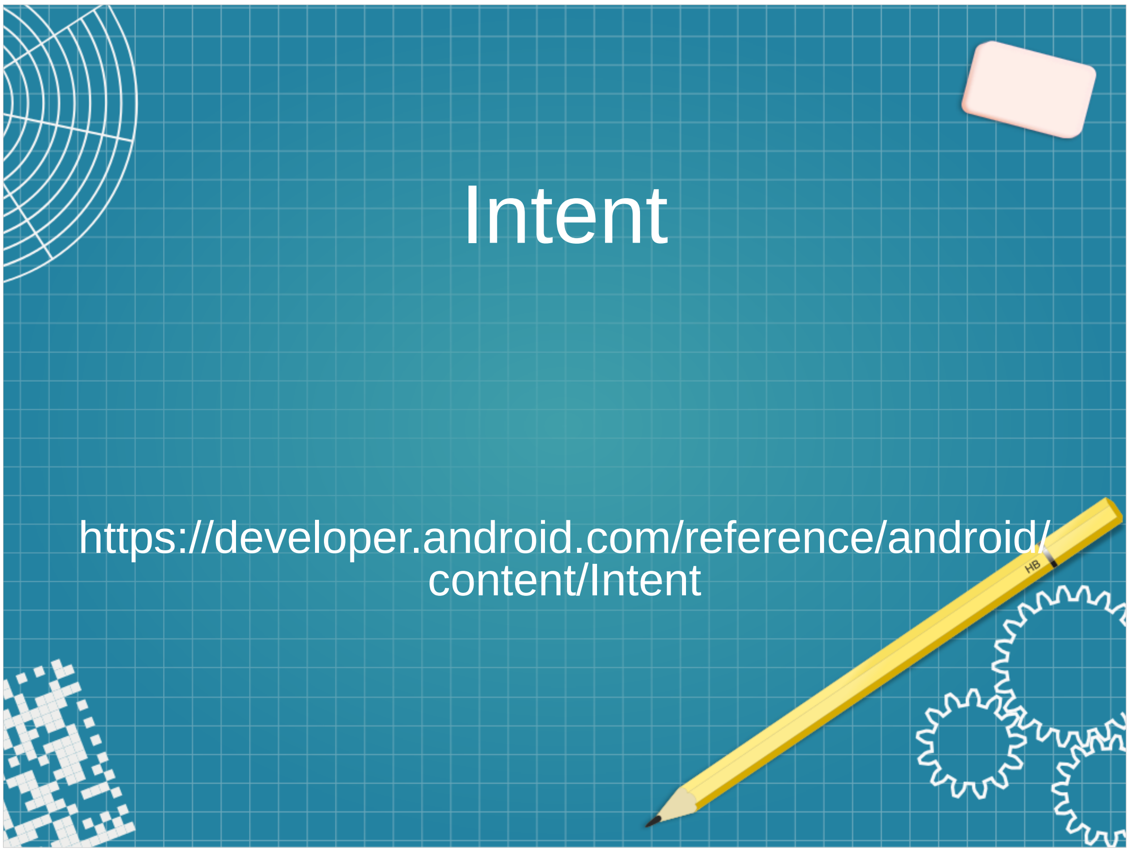
- There are three pieces of information in the Intent that are used for resolution:
  - the action,
  - type, and
  - category.
- Using this information, a query is done on the PackageManager for a component that can handle the intent.

Cont ...

- The appropriate component is determined based on the intent information supplied in the AndroidManifest.xml file as follows:
  - The **action**, if given, must be listed by the component as one it handles.
  - The **type** is retrieved from the Intent's data, if not already supplied in the Intent. Like the action, if a type is included in the intent (either explicitly or implicitly in its data), then this must be listed by the component as one it handles.

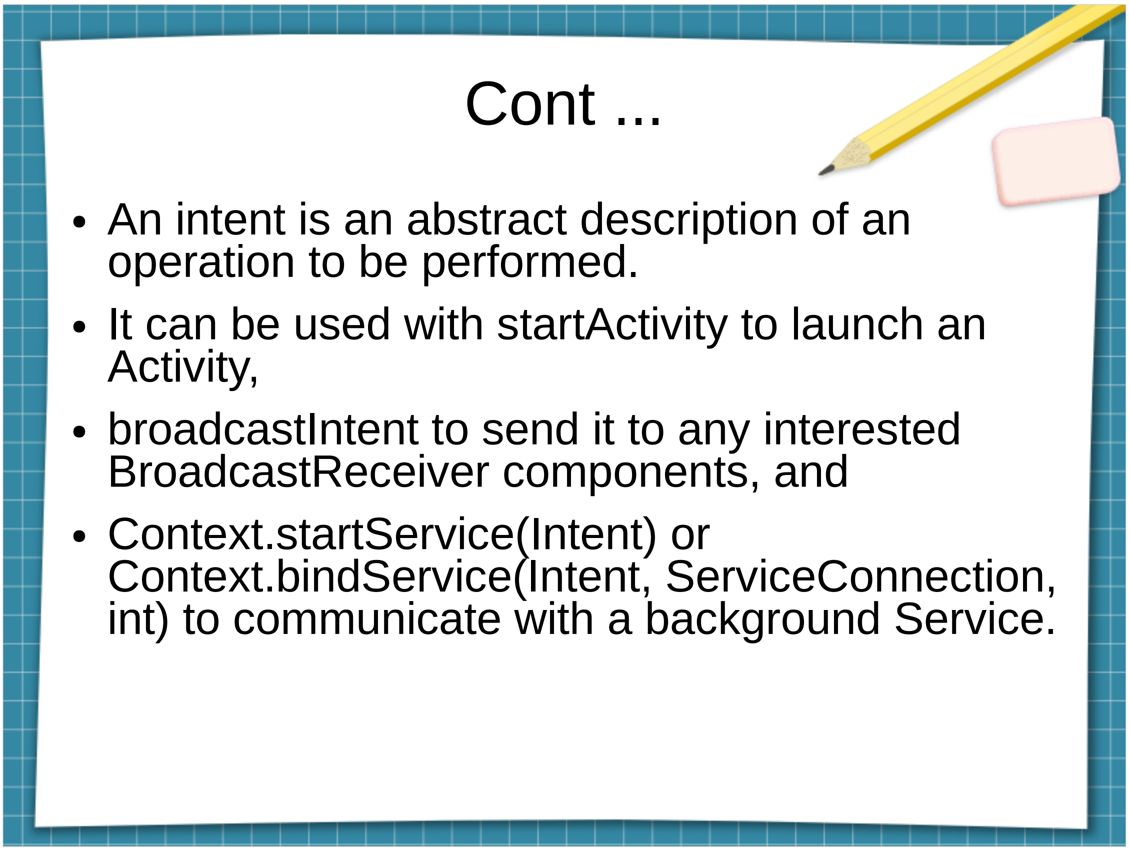
Cont ...

- For data that is not a content: URI and where no explicit type is included in the Intent, instead the **scheme** of the intent data (such as http: or mailto:) is considered. Again like the action, if we are matching a scheme it must be listed by the component as one it can handle.
- The **categories**, if supplied, must all be listed by the activity as categories it handles. That is, if you include the categories CATEGORY_LAUNCHER and CATEGORY_ALTERNATIVE, then you will only resolve to components with an intent that lists both of those categories. Activities will very often need to support the CATEGORY_DEFAULT so that they can be found by Context.startActivity().



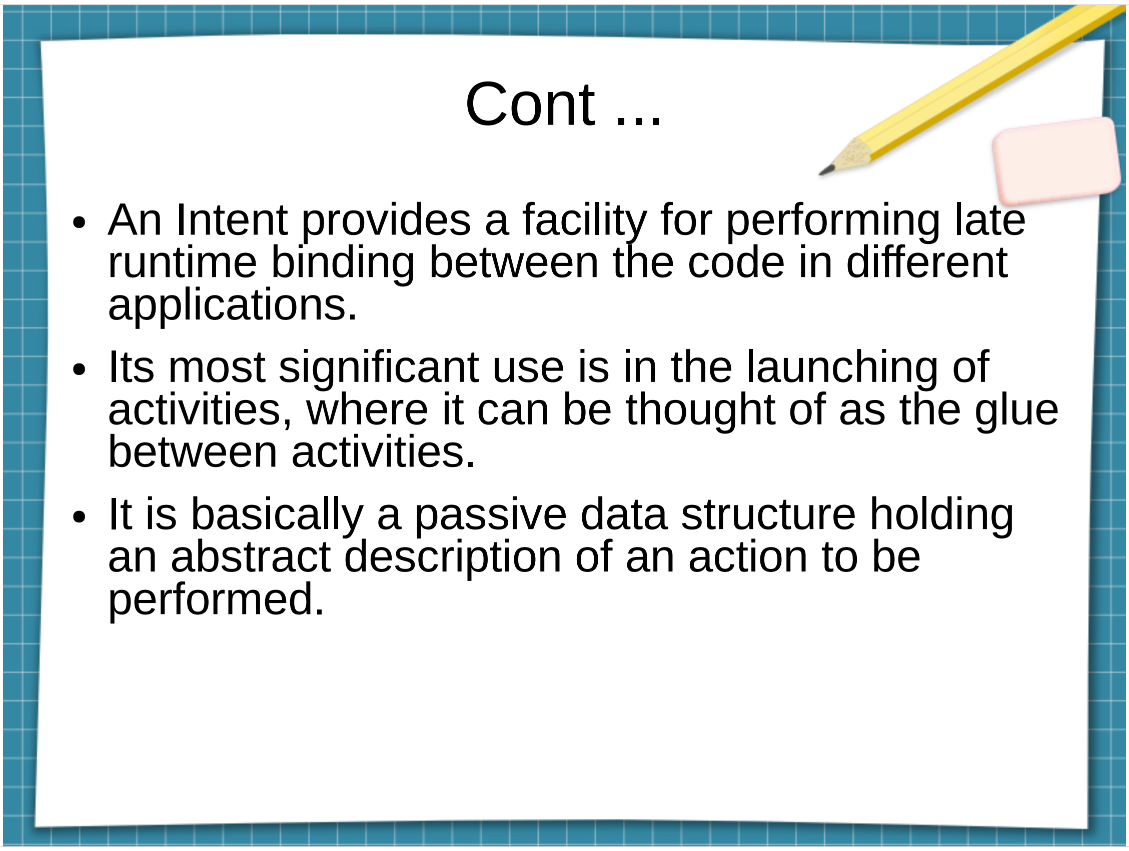
Intent

https://developer.android.com/reference/android/content/Intent



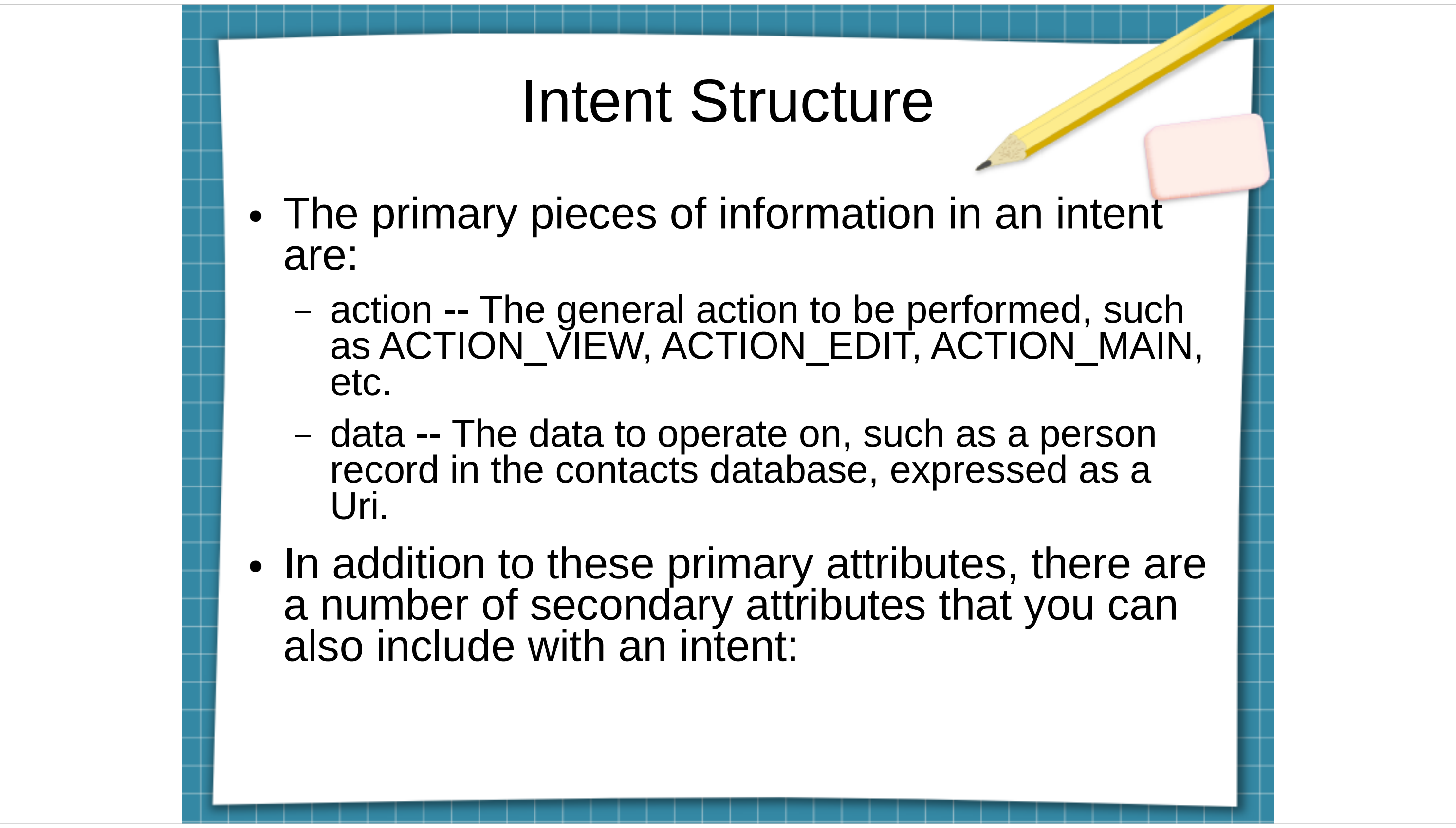
Cont ...

- An intent is an abstract description of an operation to be performed.
- It can be used with startActivity to launch an Activity,
- broadcastIntent to send it to any interested BroadcastReceiver components, and
- Context.startService(Intent) or Context.bindService(Intent, ServiceConnection, int) to communicate with a background Service.



Cont ...

- An Intent provides a facility for performing late runtime binding between the code in different applications.
- Its most significant use is in the launching of activities, where it can be thought of as the glue between activities.
- It is basically a passive data structure holding an abstract description of an action to be performed.



Intent Structure

- The primary pieces of information in an intent are:
  - action -- The general action to be performed, such as ACTION_VIEW, ACTION_EDIT, ACTION_MAIN, etc.
  - data -- The data to operate on, such as a person record in the contacts database, expressed as a Uri.
- In addition to these primary attributes, there are a number of secondary attributes that you can also include with an intent:

Cont ...

- category -- Gives additional information about the action to execute. For example, CATEGORY_LAUNCHER means it should appear in the Launcher as a top-level application, while CATEGORY_ALTERNATIVE means it should be included in a list of alternative actions the user can perform on a piece of data.
- type -- Specifies an explicit type (a MIME type) of the intent data. Normally the type is inferred from the data itself. By setting this attribute, you disable that evaluation and force an explicit type.

Cont ...

- component -- Specifies an explicit name of a component class to use for the intent. Normally this is determined by looking at the other information in the intent (the action, data/type, and categories) and matching that with a component that can handle it. If this attribute is set then none of the evaluation is performed, and this component is used exactly as is. By specifying this attribute, all of the other Intent attributes become optional.
- extras -- This is a Bundle of any additional information. This can be used to provide extended information to the component. For example, if we have a action to send an e-mail message, we could also include extra pieces of data here to supply a subject, body, etc.

Cont ...

- Put together, the set of actions, data types, categories, and extra data defines a language for the system allowing for the expression of phrases such as "call john smith's cell".
- As applications are added to the system, they can extend this language by adding new actions, types, and categories, or they can modify the behavior of existing phrases by supplying their own activities that handle them.

Intent Resolution

- There are two primary forms of intents you will use.
  - Explicit Intents have specified a component (via setComponent(ComponentName) or setClass(Context, Class)), which provides the exact class to be run. Often these will not include any other information, simply being a way for an application to launch various internal activities it has as the user interacts with the application.
  - Implicit Intents have not specified a component; instead, they must include enough information for the system to determine which of the available components is best to run for that intent.

Cont ...

- When using implicit intents, given such an arbitrary intent we need to know what to do with it.
- This is handled by the process of Intent resolution, which maps an Intent to an Activity, BroadcastReceiver, or Service (or sometimes two or more activities/receivers) that can handle it.

Cont ...

- The intent resolution mechanism basically revolves around matching an Intent against all of the <intent-filter> descriptions in the installed application packages.
- (Plus, in the case of broadcasts, any BroadcastReceiver objects explicitly registered with Context.registerReceiver(BroadcastReceiver, IntentFilter).) More details on this can be found in the documentation on the IntentFilter class.

Cont ...

- There are three pieces of information in the Intent that are used for resolution:
  - the action,
  - type, and
  - category.
- Using this information, a query is done on the PackageManager for a component that can handle the intent.

Cont ...

- The appropriate component is determined based on the intent information supplied in the AndroidManifest.xml file as follows:
  - The **action**, if given, must be listed by the component as one it handles.
  - The **type** is retrieved from the Intent's data, if not already supplied in the Intent. Like the action, if a type is included in the intent (either explicitly or implicitly in its data), then this must be listed by the component as one it handles.

Cont ...

- For data that is not a content: URI and where no explicit type is included in the Intent, instead the **scheme** of the intent data (such as http: or mailto:) is considered. Again like the action, if we are matching a scheme it must be listed by the component as one it can handle.
- The **categories**, if supplied, must all be listed by the activity as categories it handles. That is, if you include the categories CATEGORY_LAUNCHER and CATEGORY_ALTERNATIVE, then you will only resolve to components with an intent that lists both of those categories. Activities will very often need to support the CATEGORY_DEFAULT so that they can be found by Context.startActivity().