Major Changes with Android Marshmallow

Major changes with Marshmallow

- A number of changes have been made to the android system that affects not only users but also developers
  - Among these changes are systems geared towards giving user's much more control over permissions
  - And also for reducing the amount of battery drain on a device.
  - We will cover those changes here and what they mean for you as a developer

App Permissions

- Perhaps the most significant change between previous versions of android and marshmallow.
  - There is now a clear divide how permissions are handled for devices pre-marshmallow and post-marshmallow.
  - We will detail both systems and will compare them

App Permissions before marshmallow

- In devices that are pre marshmallow the permission situation was simple
  - In order to install an application a user would have to give the application all of the permissions that it requested.
    - Or else not install the application at all.
  - Not a good security model.
    - Apps that wanted a lot of permissions could be considered invasive but the app may still be needed for the user to do their daily tasks.

App Permissions before Marshmallow

- Also it enables applications to perform “permission creep”
  - Starts by requesting few permissions to begin with
  - But over the course of multiple updates it requests an additional permission at a time.
    - Until it gets close to or has the full set of permissions
    - A security issue in the long run.

App Permissions after Marshmallow

- To rectify this situation the security system is now changed in marshmallow.
  - By default an application will be granted all of the permissions it requests when it is installed for the first time.
  - However, users in the application settings system menu are now permitted to selectively grant or reject individual permissions for each application

App Permissions after Marshmallow

- This gives a user a lot more control over what an individual application can do with regards to permissions
  - As users become more educated about permissions and the kind of information that can be gleaned from simple pieces of information
    - They will close down applications better
  - See the difference in privacy awareness and mass survelliance ever since the Snowden revelations

App Permissions after Marshmallow

- This now means that for the lifetime of your application you may find that certain permissions have been removed.
  - Meaning you have to ask for them again from your user.
  - Slightly increases the complexity of your code
  - Also you have to be upfront about why you need the permissions

App Permissions after Marshmallow

- Some permissions are obvious and don't need explaining
  - Use of camera for a photo app
  - Use of location for a navigation app
- Others you need to inform your user why you need the permission
  - Users will be suspicious of apps that require location information when it is not clear why it is needed.
  - Or needs access to your contacts when it cannot make communications of any kind.

App Permissions after Marshmallow

- Thus you are now required to check if you have access to a permission before you try and access it
  - Failure to do so results in an exception.
  - Further details can be found here
  - http://developer.android.com/training/permissions/requesting.html

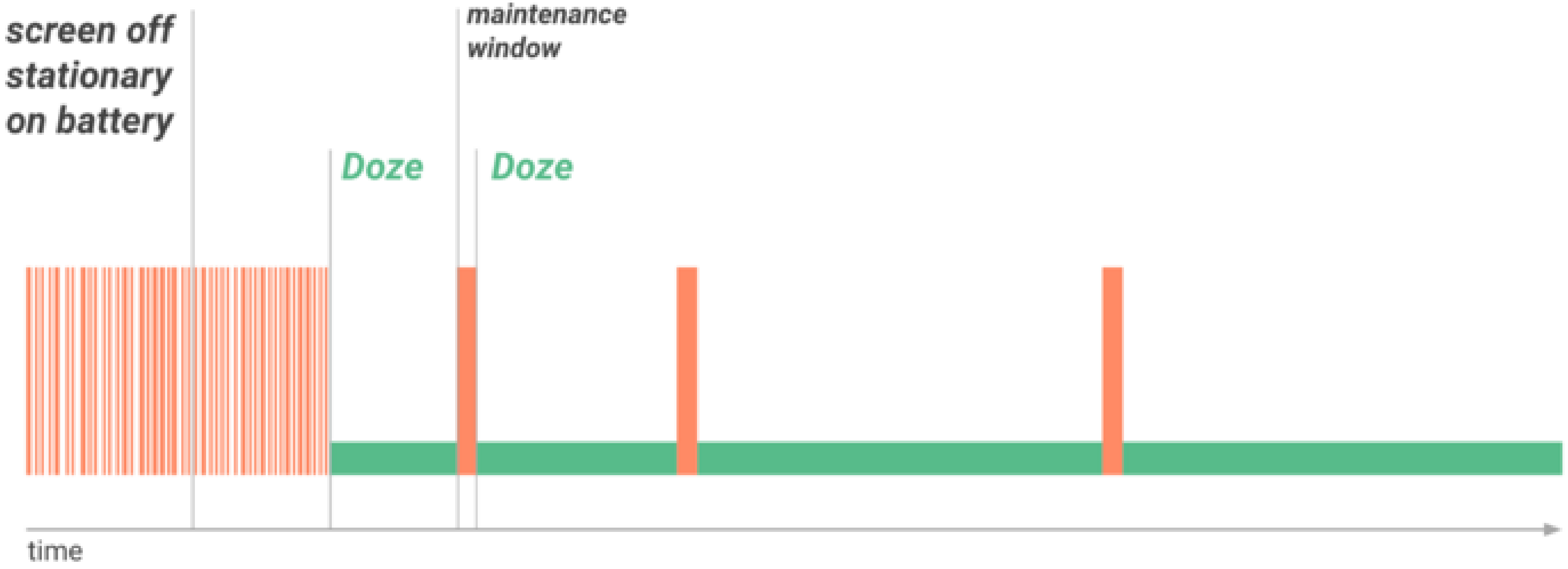
Changes to Battery usage

- Android has also introduced a couple of extra features to further improve the battery life of devices.
  - Doze mode
  - App Standby
- These are always on and cannot be disabled by your application.

Doze mode

- Doze mode keeps track of whether the device is active and motionless.
  - If the screen has been switched off and the device has not been used for a while it will modify the behaviour of apps that require network and CPU intensive services.
    - The idea being the longer you keep a CPU in the deep sleep state the longer the battery will last.
    - See the activity diagram on the next slide

Doze Mode


screen off
stationary
on battery
maintenance
window
Doze
Doze
time

Doze mode

- The orange lines represent periods of activity and the green represents periods of inactivity
  - What you see before the first vertical line is a period where the user is using a device with the screen on and there is lots of cpu and network activity.
    - The device rarely hits deep sleep
  - For a set time period after the device has been left stationary the device determines that it can enter doze mode
    - All network activity will be deferred and grouped until the next maintenance window is reached.

Doze mode

- Each maintenance window is denoted by a contigious block of orange where all deferred tasks are executed.
  - Upon completion the device reenters doze mode until the next window.
- The longer the device is left inactive the longer the gap between each maintenance window
  - Device exits doze mode when it is moved or the screen is turned on.

Doze Restrictions

- This is what is restricted when the device enters doze mode
  - Suspended network access
  - Wake locks are ignored
  - AlarmManager alarms are deffered to the next window
    - Normal alarms are not affected
  - The system does not perform WiFi scans
  - Sync adapters and job schedulers are not run.

App Standby

- App Standby determines if an app is idle when the user is not actively using it.
- An app is considered to be in standby if the following three conditions do not hold
  - The user explicity launches an app
  - The app has a process currently in the foreground
  - The app generates a notification that users see on the lock screen or in the notification tray.

App Standby

- Apps are only released from standby when the user
  - Interacts with them
  - Plugs in a device to a power supply