SCIENCES SORBONNE UNIVERSITÉ

Cameras & the image library

Fabrice.Kordon@lip6.fr

LIP6

As an introduction…

Many cameras now…

- Front
- Back (system of cameras seen as one 😎)
- And the image library

How to access images?

- UIIMagePickerController
  - **And its protocol, UIImagePickerDelegate**
- You may choose the source of the picker
  - **photoLibrary**
  - **camera**
  - **savedPhotosAlbum**
- Check for availability
  - **At least for the simulator, otherwise :** 

UIImagePickerController

- **Creation**
  - As a normal object
- **Check for camera's existence**
  - isSourceTypeAvailable(_:) / isSourceTypeAvailable:
- **Select a source**
  - sourceType property
  - Force taking a picture
    - **takePicture()**
    - **No need to present the Picker**
  - You may enforce the use of the flash
    - **cameraFlashMode**
    - **Possible values: off, auto, on**
  - Also for video (not detailed)
    - **startVideoCapture() / stopVideoCapture()**

UIImagePickerControllerDelegate

Two methods

```
func imagePickerController(_ picker: UIImagePickerController,
        didFinishPickingMediaWithInfo info: [UIImagePickerController.InfoKey : Any])

func imagePickerControllerDidCancel(_ picker: UIImagePickerController)
```

Requires UINavigationControllerDelegate too

- Even if no explicit usage

Launching the picker

- Presentation thanks to a dedicated controller
- As for a UIPickerView

Retrieving the image

In the info parameter…

```
func imagePickerController(_ picker: UIImagePickerController,
        didFinishPickingMediaWithInfo info: [UIImagePickerController.InfoKey : Any])
```

- Dictionary containing various data, info

Key	Information
cropRect	*The cropping rectangle that was applied to the original image*
editedImage	*An image edited by the user*
imageURL	*URL of the image file*
livePhoto	*Live Photo representation of the selected or captured photo*
mediaMetadata	*Metadata for a newly-captured photograph*
mediaType	*The media type selected by the user*
mediaURL	*The filesystem URL for the movie*
originalImage	*The original, uncropped image selected by the user*
phAsset	*A Photo asset for the image*

Retrieving the image

In the info parameter…

```
func imagePickerController(_ picker: UIImagePickerController,
        didFinishPickingMediaWithInfo info: [UIImagePickerController.InfoKey : Any])
```

- Dictionary containing various data, info

The one you want…

For taking pictures of course

Key	
cropRect	*…inal image*
editedImag	
imageURL	
livePhot	*Live Photo representation of the selected or captured photo*
mediaMeta…data	*Metadata for a newly-captured photograph*
mea	*The media type selected by the user*
media…	*The filesystem URL for the movie*
originalImage	*The original, uncropped image selected by the user*
phAsset	*A Photo asset for the image*

Privacy concerns

Differentiated access

- Camera
- microphone
- photo library

For movies

- Camera + microphone

Since iOS10

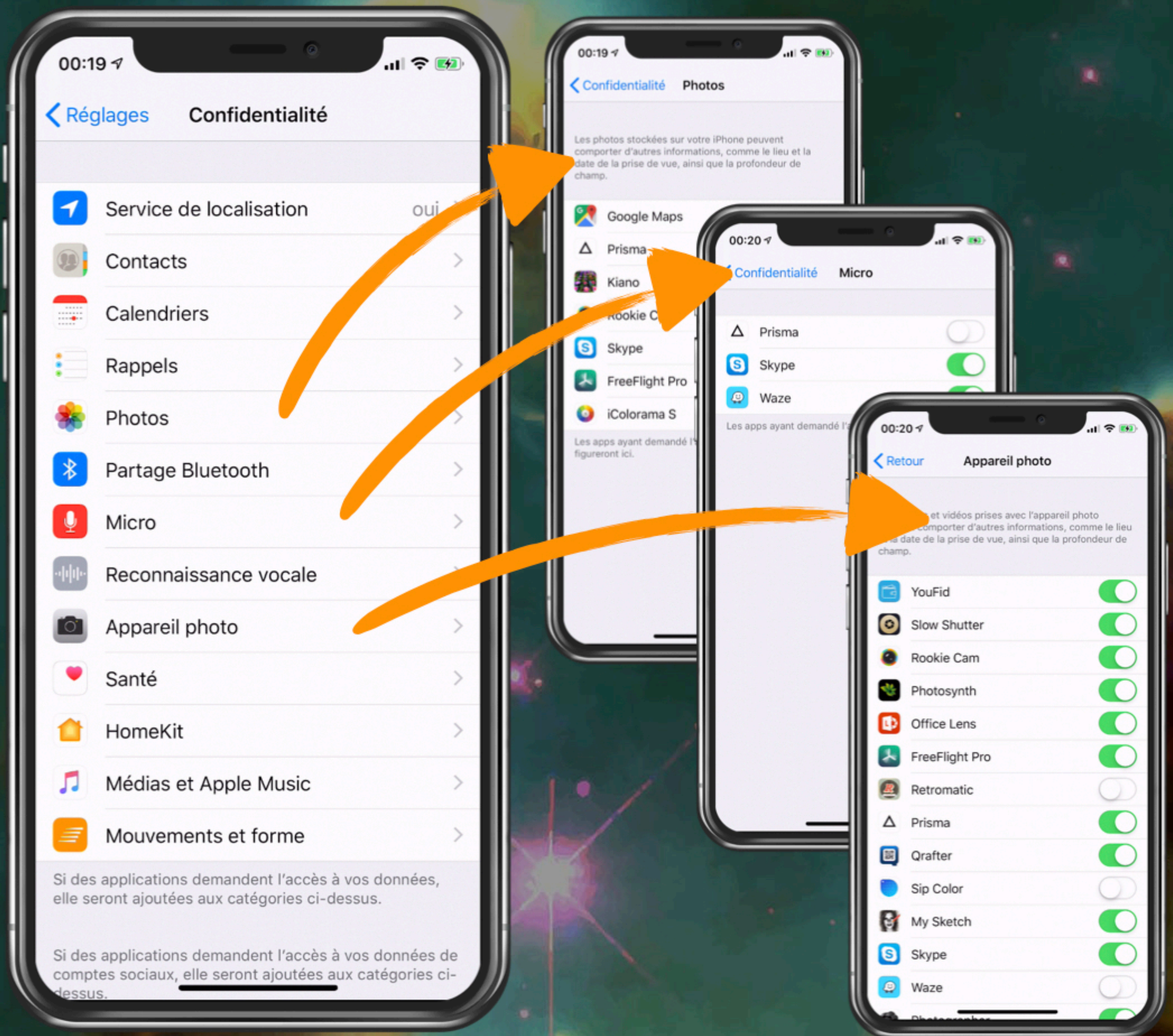
Yo must declare access in info.plist

- NSCameraUsageDescription
- NSMicrophoneUsageDescription
- NSPhotoLibraryUsageDescription

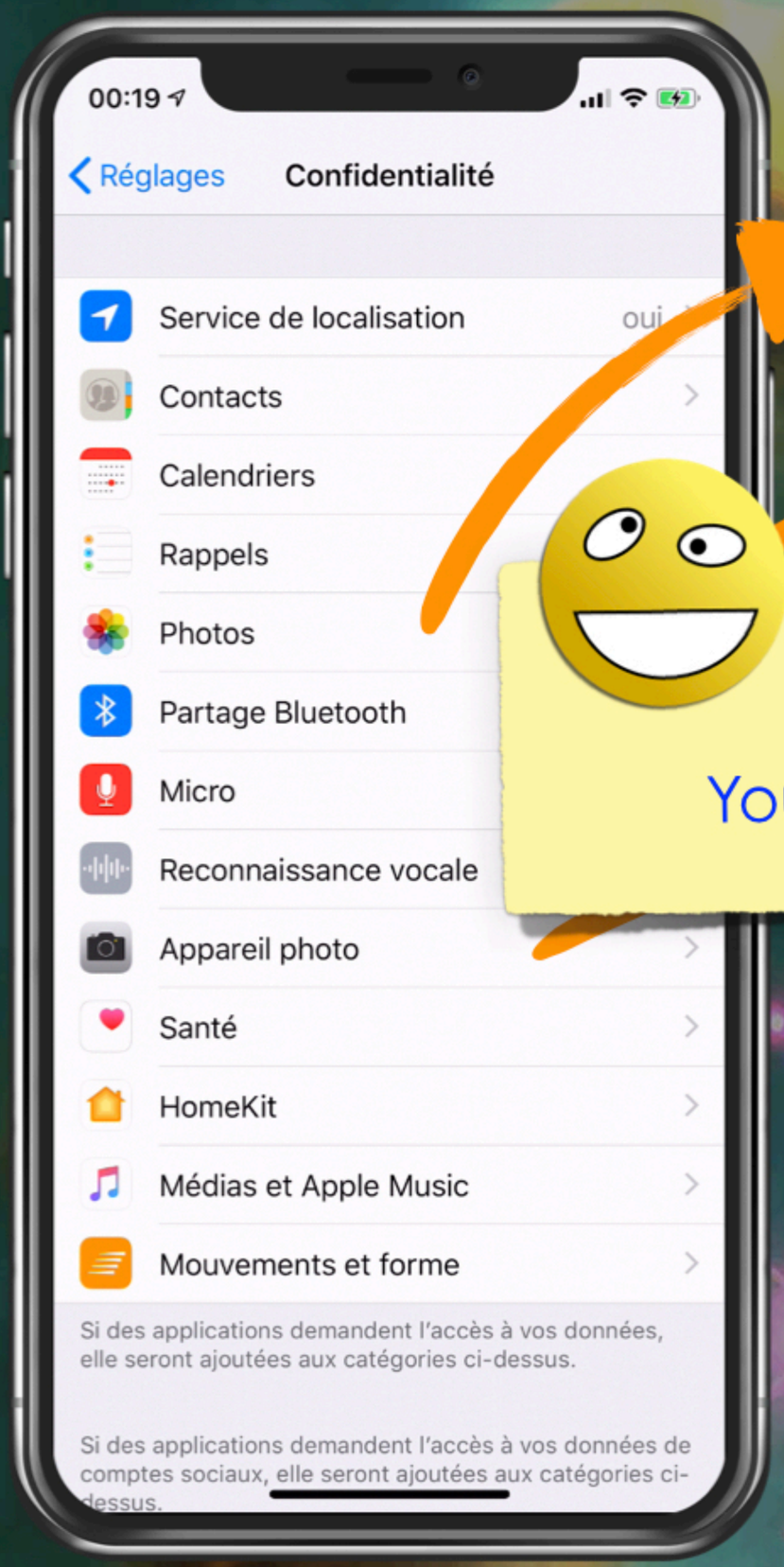
```
<key>NSCameraUsageDescription</key>
<string>This Application needs to acces your Camera</string>
<key>NSMicrophoneUsageDescription</key>
<string>This application needs to access the microphone</string>
<key>NSPhotoLibraryUsageDescription</key>
<string>This application needs to access the photo library</string>
```

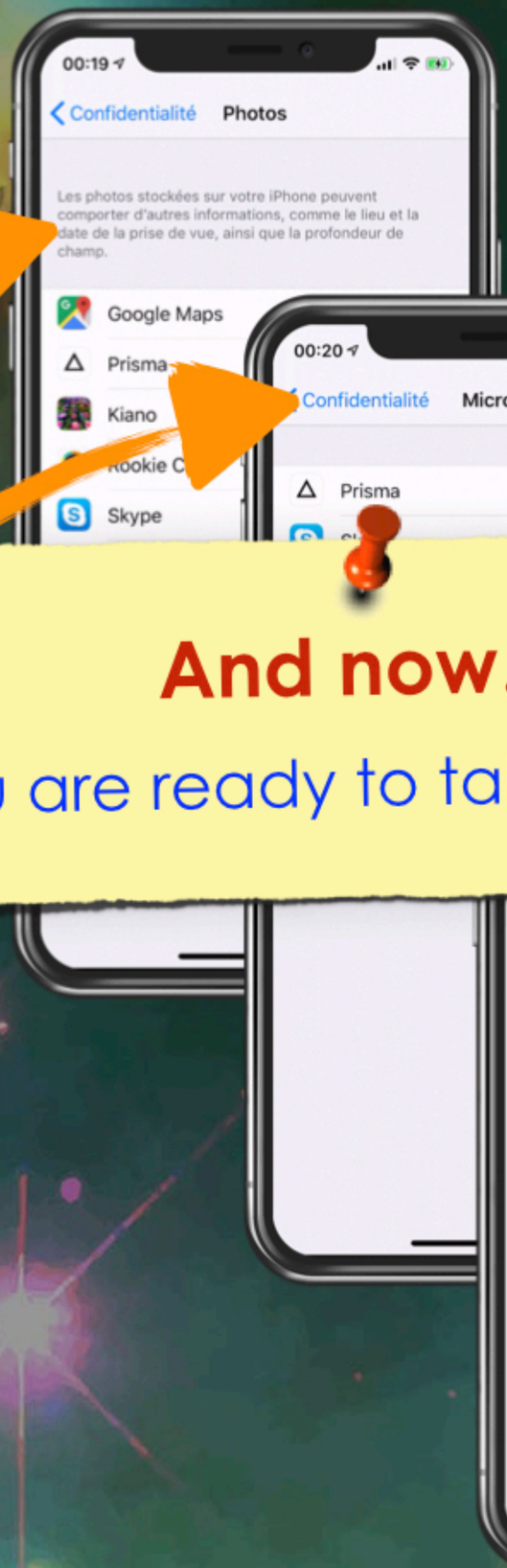
- Otherwise it does not work

As a conclusion…



As a conclusion…





And now…

You are ready to take pictures