iOS Foundations II Session 6

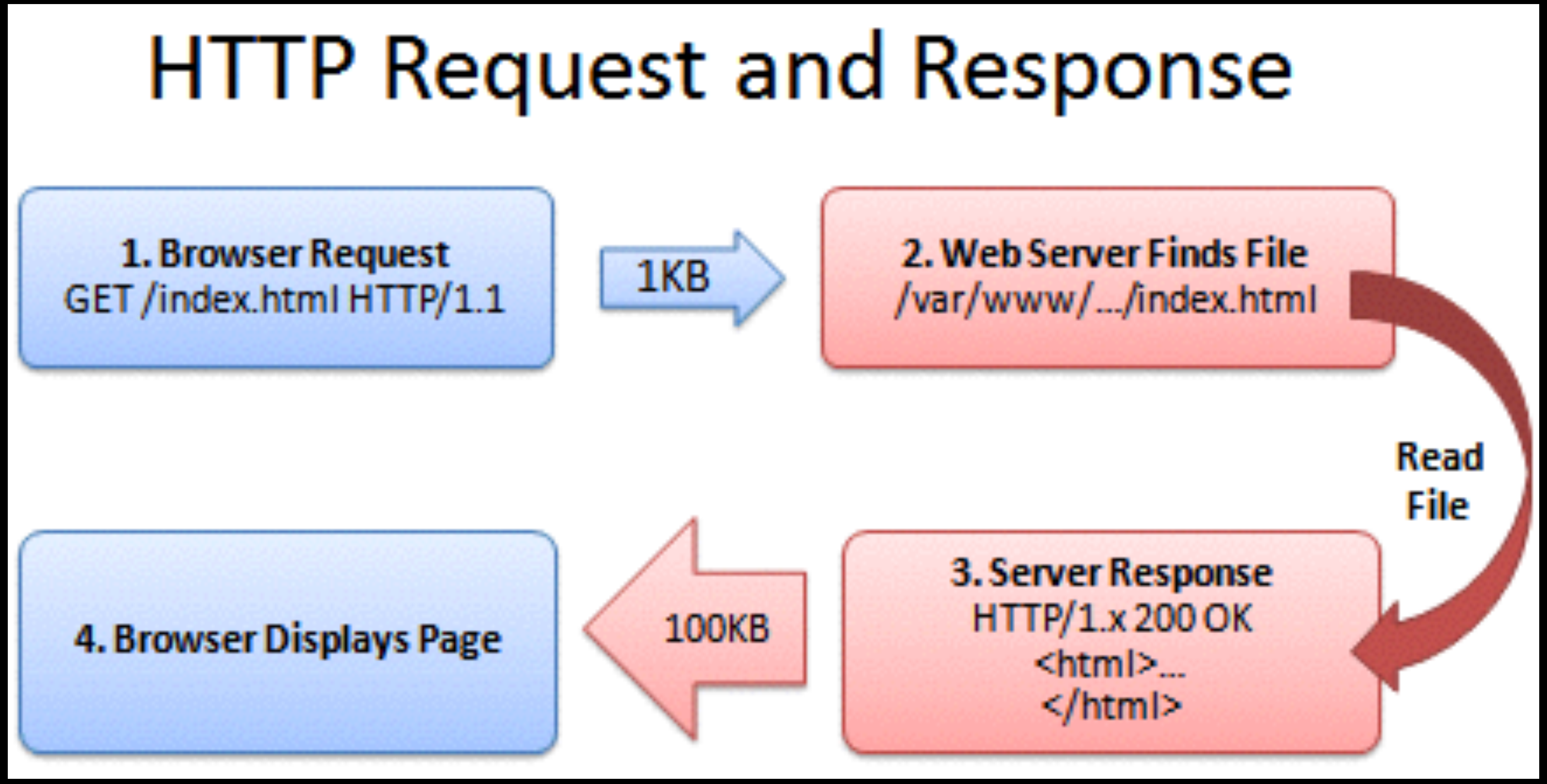
- Intro to networking (how the internet works)
- Intro to multithreading

HTTP (Hyper Text Transfer Protocol)

- HTTP is a protocol that the internet uses to format and transmit web pages.
- It also defines how web servers and browsers should respond to various commands.
- When entering a URL into your browser, you are just sending an HTTP command to a web server to retrieve a resource/webpage.

HTTP (Hyper Text Transfer Protocol)

- HTTP uses a client-server model.
- A client sends a request to a server, and then the server returns a response message and usually the resource that was requested.



HTTP Methods (Verbs)

Commonly Used

- GET - Most common HTTP method. Retrieves whatever resource is at the URL location. Your browser history is just a history of all the GET requested you have made.
- POST - Method used to request that the server accept data enclosed in the request. When you tweet, you are using POST.
- PATCH - Used to update a resource. Designated for sending only the changes, rather than a fully constructed object

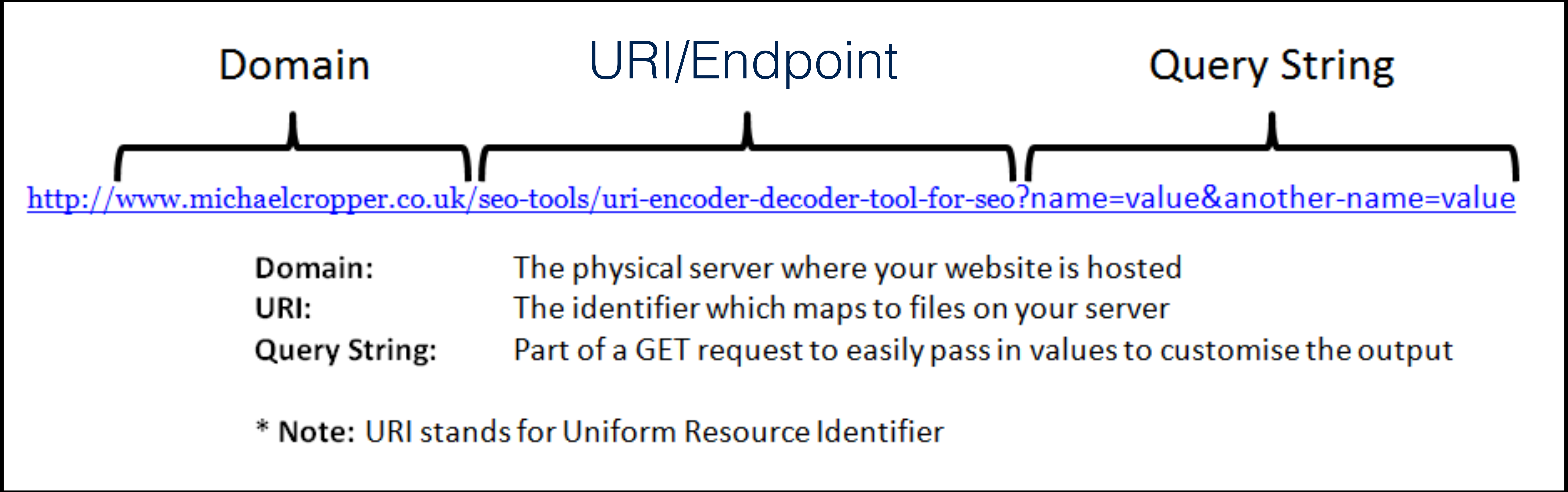
Less Commonly Used

- DELETE, HEAD, PUT, TRACE, OPTIONS, CONNECT

HTTP Status Codes

- For the most part only a few status codes that mobile apps need to check for
- 200 OK - standard response for a successful HTTP request
- 400 Bad - bad request, most likely syntax error
- 401 Unauthorized - authentication was required but not provided or incorrect in request
- 403 Forbidden - Request was valid, but the server refuses to respond.
- 404 Not found - The requested resource was not found
- 429 Too Many Requests - rate limited
- 5xx Server Error - not your app’s fault!

URL


Domain
URI/Endpoint
Query String
http://www.michaelcropper.co.uk/seo-tools/uri-encoder-decoder-tool-for-seo?name=value&another-name=value

Domain:	The physical server where your website is hosted
URI:	The identifier which maps to files on your server
Query String:	Part of a GET request to easily pass in values to customise the output

* **Note:** URI stands for Uniform Resource Identifier

HTTP GET

- Lets say I wanted to get the box score for a specific mariners game. Heres the URL I would use for my GET request:
  - ★ http://scores.espn.go.com/mlb/boxscore?gameId=340718103
- the domain is http://scores.espn.go.com
- the endpoint is mlb/boxscore
- the query string is gameId=340718103. The beginning of the query string is marked by a question mark. You can add multiple parameters by appending them with a &
- Demo!

HTTP Header Fields

- A POST request is a bit more complicated. It requires certain header fields so the server knows what the data we are sending is. Good API's will tell you exactly which header fields you need. A lot of them won't :(
- The format of a header field is "Header-Name : value"
- Content-Type header field is used to specify the nature of data in the body of the request.
- Authorization header field can be used to pass credentials for protected resources. Usually an OAuth token.

Foundation Network Objects

- NSData - an object oriented wrapper for byte buffers (binary data).
- NSData has a number of methods to create data objects from raw bytes.
- var path = “/u/smith/myFile.txt”
- var myData = NSData(dataWithContentsOfFile:path)
- Can also go the other way:
- var image = person.Image
- var imgData = UIImagePNGRepresentation(image)

Foundation Network Objects

- NSURL : object that represents a URL that can contain the location for a resource on a remote server or a path of a file on disk.
- Can examine the URL scheme (http), host (www.example.com), path (/scripts), and query string (name=value) via properties on the NSURL object.
- Instantiated with a string.
- NSURLRequest : object for a URL Request.
- Instantiate with an NSURL object.
- Has many properties for specifying the attributes of the HTTP request (http method, body, header fields)

APIs

- Now that you Learned The Internet, lets look at how our iOS apps can communicate with web apps/sites
- API, or Application programming Interface, is way for parts of your software to interface with other software.
- Web APIs, what you will be working with, are defined as a set of HTTP requests and response messages usually in JSON or occasionally in XML.
- Most apps on your iPhone are just clients for a web api (facebook, twitter, instagram, spotify, etc)

Web API Client workflow

1. Clients makes a request to the server at a specific endpoint
2. Server receives request, does some server magic, and then sends back response
3. Client checks the response http status code
4. If its 200 (everything is good) client parses the JSON response into model objects
5. client updates UI and state with newly acquired model objects

Multi-Threading

- “Threads are a relatively lightweight way to implement multiple paths of execution inside of an application”
- “In a non-concurrent application, there is only one thread of execution”
- “A concurrent application starts with one thread and adds more as needed to create additional execution paths”
- The problem with a non-concurrent application is it’s one thread can only do one thing at a time.
- When an app is sluggish or frozen, its main thread is probably bogged down by an expensive computation. Running the expensive computation on another thread would allow the main thread to remain responsive to the user.

Multi-Threading adds complexity

- “Having multiple paths of execution can add a considerable amount of complexity to your code”
- “Each thread has to coordinate its actions with other threads to prevent it from corrupting the app’s state information”
- If two threads try to access the same object in memory you may corrupt the object.
- ALL code that interacts with User Interface (UIKit) objects must be performed on the main thread otherwise you get very unreliable behavior.

Operations

- Thankfully Apple has a few API’s designed to make multithreading code easier to manage.
- “Cocoa operations are an object-oriented way to encapsulate work that you want to perform asynchronously.
- “By far, the easiest way to execute operations is to use an operation queue, which is an instance of NSOperationQueue class”

NSOperationQueue

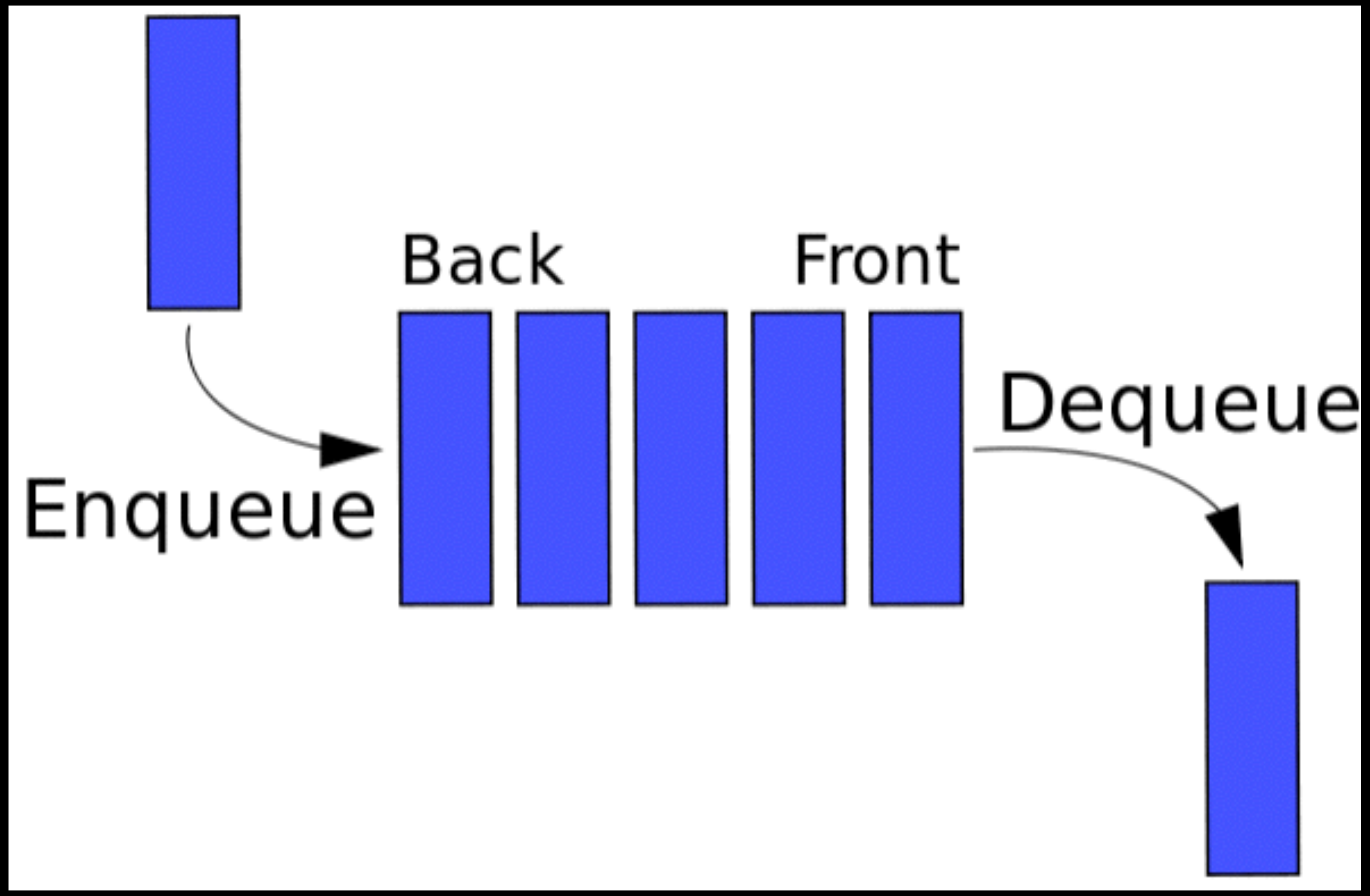
- 3 ways to add an operation to a queue:
    1. addOperation(operation:) - add a single NSOperation object
    2. addOperations(operations: waitUntilFinished:) - add an array of NSOperations
    3. addOperationWithBlock(closure:()->) - write a closure with the code you want to execute, this code is then packaged as an operation and added to the queue

NSOperationQueue details

- Concurrent by default, but can set maxConcurrentOperationCount to 1 to make it serial.
- Offers a mainQueue class property, which returns a queue that is associated with the main thread. Run your UI code on this queue.
- Has a cancelAllOperations method, although its not quite that simple to cancel all operations.

Queue data structure

- Another abstract data source, similar to a stack.
- The only actions are to position an entity at the rear (enqueue) and removal of an entity from the front (dequeue).
- FIFO - First in First out.
- Entities are kept in order



NSOperationQualityOfService

- Queues and Operations each have their own QoS level
- Operations get promoted to the QoS level of their queue
- Designed for better performance and power efficiency in your apps
- UserInitiated for timely, non-blocking tasks
- Background or Utility for low-priority tasks
- UserInteractive for nearly-mainQueue priority