

NSURLSession: New Features and Best Practices

Session 711

Jeff Jenkins Software Engineer, Internet Technologies

Networking API

Foundation

NSURLSession

NSNetServices

NSStream

New
2013

Foundation

NSURLConnection

NSNetServices

NSStream

Deprecated
2015

NSURLSession

Review

NSURLSession Example

Benefits

HTTP/1.1, SPDY, HTTP/2

App Transport Security (ATS)

HTTP Strict Transport Security (HSTS)

Cache, cookies, proxy, authentication

Configuration

NSURLSession Example

Benefits

NSURLSessionConfiguration

Transport Layer Security (TLS) version

Prohibit cellular usage

Network service type

Cookie policy

Cache policy

Storage objects

Request and resource timeouts

NSURLSession Example

Review

```
let url = NSURL(string: "https://www.example.com/")!
```

```
let task = session.dataTask(with: url) { (data: NSData?, response: NSURLResponse?, error:
NSError?) in
    ...
}
```

```
task.resume()
```

NSURLSession API

HTTP/2

NEW

HTTP/2 Protocol

- Multiplexing and concurrency
- Header compression
- Stream priorities
- Server Push



- Apple
- Facebook
- Instagram
- Wikipedia
- Twitter
- YouTube
- Yahoo

NSURLSession API

HTTP/2 Server Push

NEW

Prevent network round trips

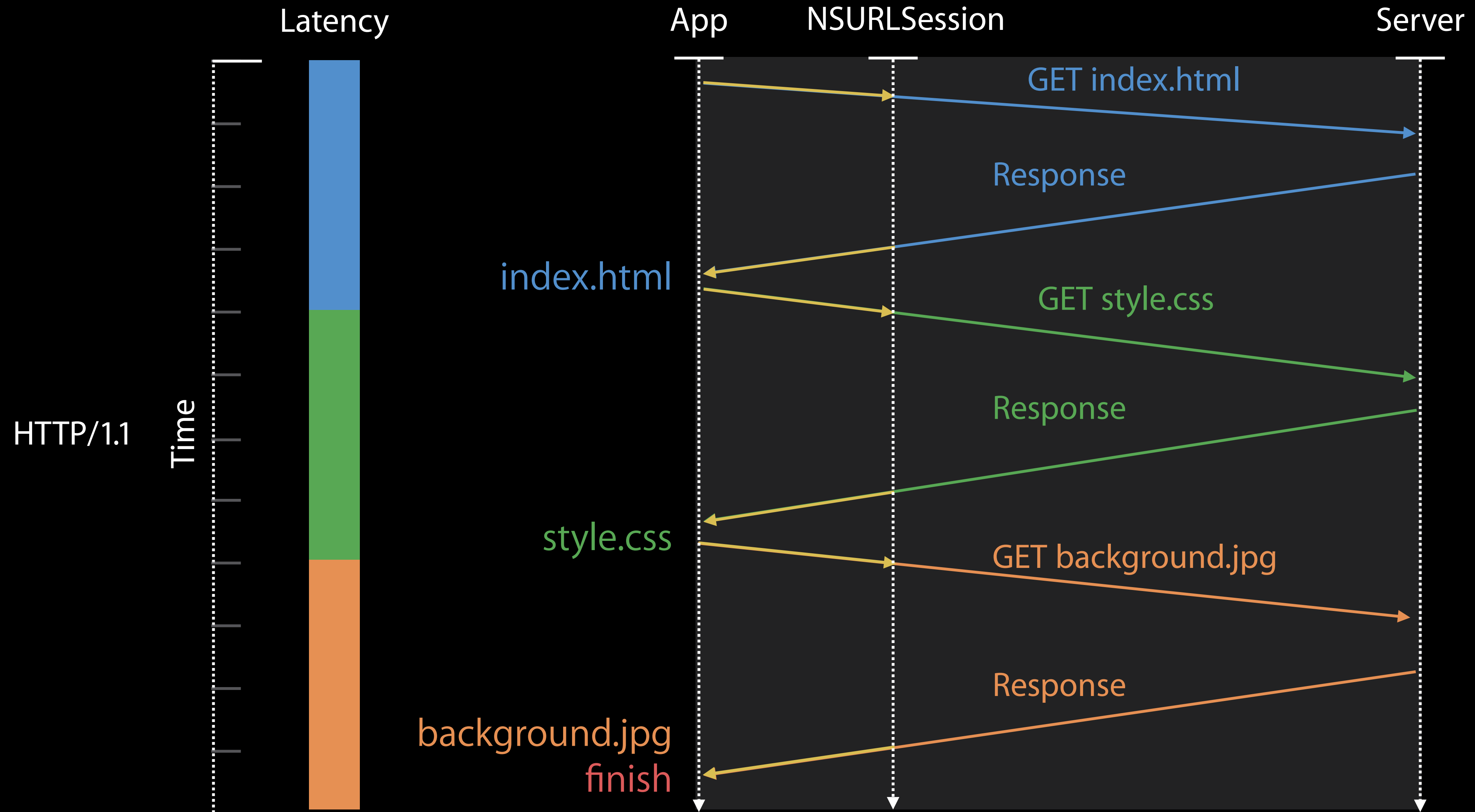
Server support required

Now available in NSURLSession



HTTP/1.1

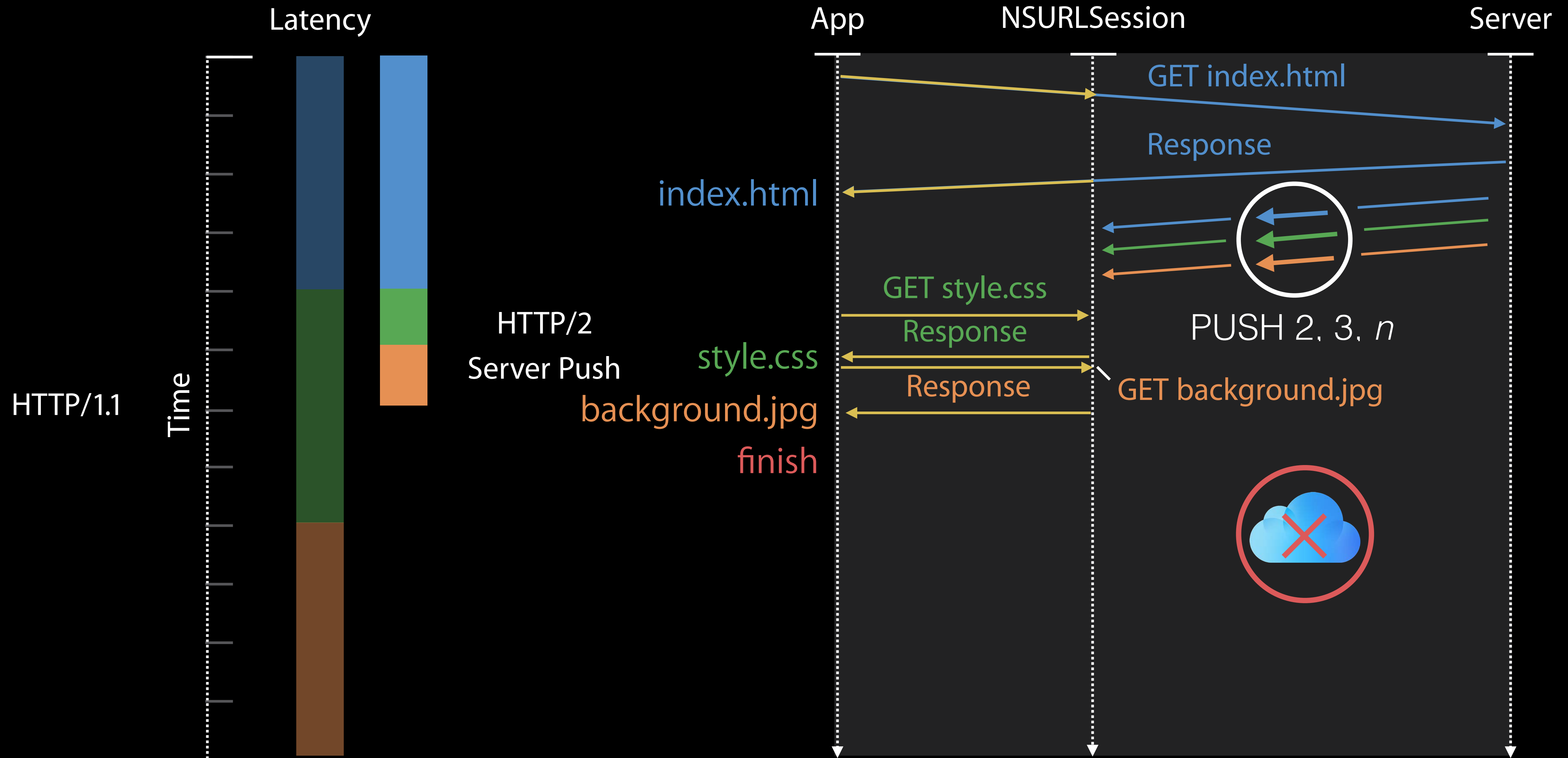
Loading



HTTP/2 Server Push

NEW

Loading



Demo

HTTP/2 Server Push

Andreas Garkuscha Software Engineer, Internet Technologies



此处应有视频



HTTP/2 Server Push

Summary

Server Push supported in NSURLSession

No opt in, just works

Even better performance

NSURLSession API

NEW

Network statistics

Did you know?

98% of all statistics are made up!

Network statistics: Performance, bugs, understanding

Currently supported on iOS, macOS, and tvOS

NSURLSession API

Network statistics delegate

NEW

```
func urlSession(_ session: NSURLSession, task: NSURLSessionTask,  
didFinishCollecting metrics: NSURLSessionTaskMetrics)
```

NSURLSession API

NSURLSessionTaskMetrics

NEW

NSURLSessionTaskMetrics

NSURLSessionTaskTransactionMetrics

```
taskInterval: NSDateInterval
```

```
redirectCount: Int
```

```
transactionMetrics: [NSURLSessionTaskTransactionMetrics]
```



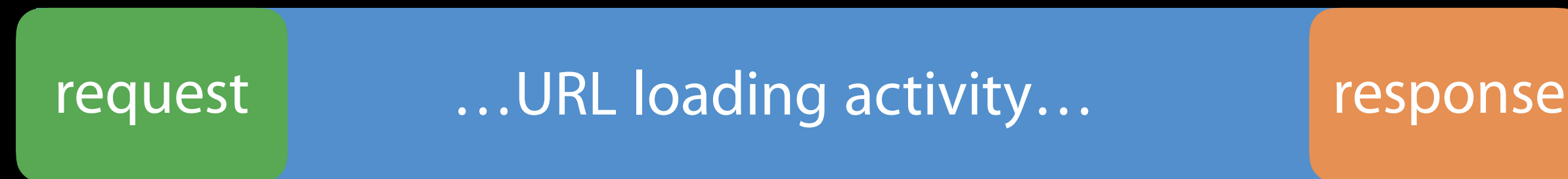
Four Categories of Metrics

NSURLSession API

NEW

NSURLSessionTaskTransactionMetrics

1 - Request and Response



```
request: NSURLRequest
```

```
response: NSURLResponse?
```

NSURLSession API

NEW

NSURLSessionTaskTransactionMetrics

2 - Protocol and Connection

```
networkProtocolName: String?
```

http/1.1

h2

spdy/3, spdy/3.1

```
isProxyConnection: Bool
```

```
isReusedConnection: Bool
```



NSURLSession API

NEW

NSURLSessionTaskTransactionMetrics

3 - Load Info

```
resourceFetchType: NSURLSessionTaskMetricsResourceFetchType
```

```
.networkLoad
```

```
.localCache
```

```
.serverPush
```

NSURLSession API

NEW

NSURLSessionTaskTransactionMetrics

4 - Connection Establishment and Transmission

```
fetchStartDate: NSDate?  
domainLookupStartDate: NSDate?  
domainLookupEndDate: NSDate?  
connectStartDate: NSDate?  
secureConnectionStartDate: NSDate?  
secureConnectionEndDate: NSDate?  
connectEndDate: NSDate?
```

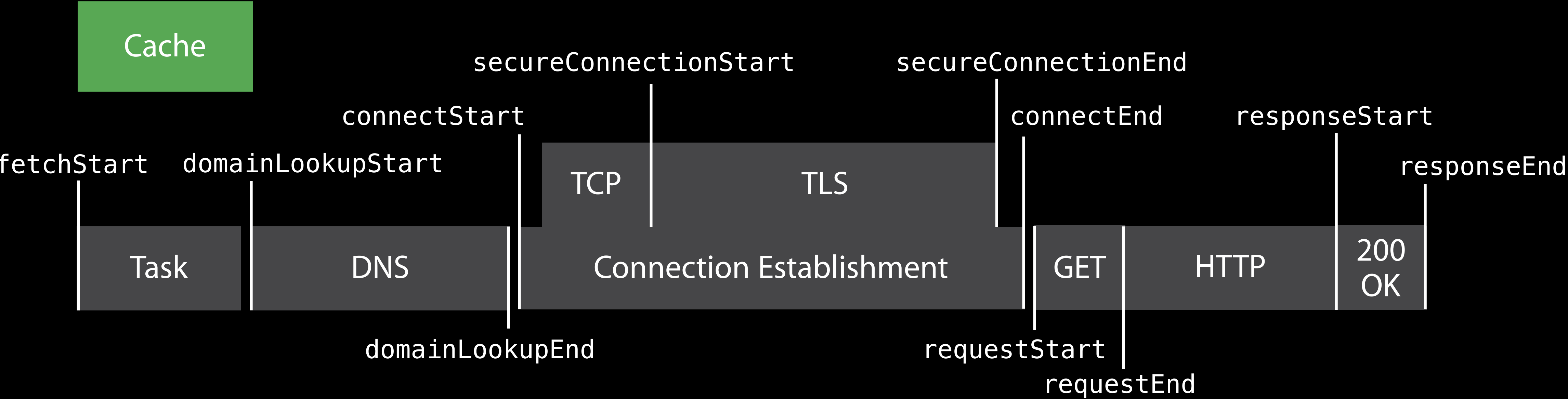
```
requestStartDate: NSDate?  
requestEndDate: NSDate?  
responseStartDate: NSDate?  
responseEndDate: NSDate?
```



NSURLSession API

NEW

NSURLSessionTaskTransactionMetrics



NSURLSession API

NSURLSessionTaskTransactionMetrics

```
class MySessionDelegate: NSObject, NSURLSessionTaskDelegate {
    @objc(URLSession:task:didFinishCollectingMetrics:)
    func urlSession(_ session: NSURLSession, task: NSURLSessionTask, didFinishCollecting metrics:
NSURLSessionTaskMetrics) {
        //metrics.redirectCount
        //metrics.taskInterval
        //metrics.transactionMetrics[0].connectStartDate ...
    }

    let myDelegate = MySessionDelegate()

    let config = NSURLSessionConfiguration.defaultSessionConfiguration()
    let myDelegateQueue = NSOperationQueue()
    let session = NSURLSession(configuration: config, delegate: myDelegate, delegateQueue: myDelegateQueue)

    let url = NSURL(string: "https://www.example.com/")!
    let task = session.dataTask(with: url) { (data: NSData?, response: NSURLResponse?, error: NSError?) in
        ...
    }
}
```

```
task.resume()
```

NSURLSession API

Network statistics

Summary

- ✓ Deep view into network loading
- ✓ Faster debugging, easier profiling, better performance
- ✓ Available in NSURLSession

Security

Never an afterthought

Security

Transport Layer Security

Protects data

Cipher—algorithm for encrypting/decrypting data

The RC4 cipher is no longer supported



- Tuesday—Security session and labs

How iOS Security Really Works

Nob Hill

Tuesday 4:00PM

What's New in Security

Nob Hill

Tuesday 5:00PM

Security

nsurl

```
$ nsurl https://insecure.example.com/
```

```
nsurl[1234:123456] NSURLSession/NSURLConnection HTTP load failed  
(kCFStreamErrorDomainSSL, -9824)
```

Security

nscurl

```
$ nscurl --enable-rc4 https://insecure.example.com/
```

```
Enabling RC4 cipher suites
```

```
<html><body><h1>It works!</h1></body></html>
```

Summary

NSURLSession

- HTTP/2 Server Push Performance!
- Network statistics Faster and better analysis of applications
- Security Protect user data

NSURLSession in your applications!