

# Mobile Data: How to avoid the latency trap when using web services

# Tom Hughes-Croucher

@sh1mmer

**YAHOO!** DEVELOPER NETWORK

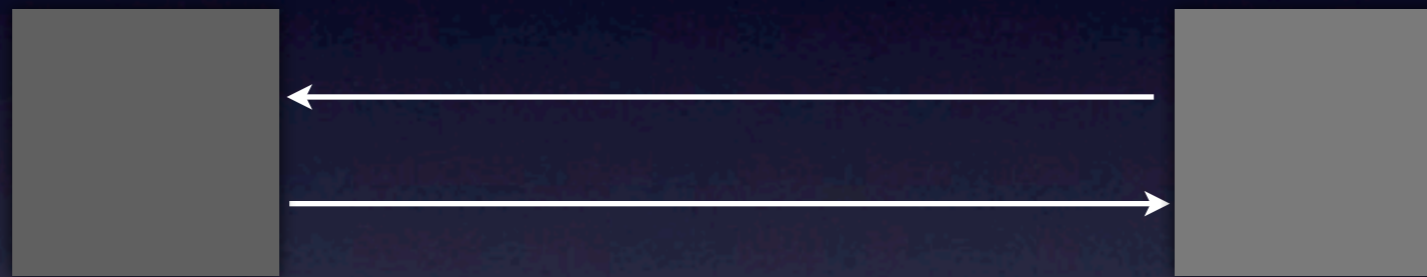
What is the latency  
trap?



# Mobile

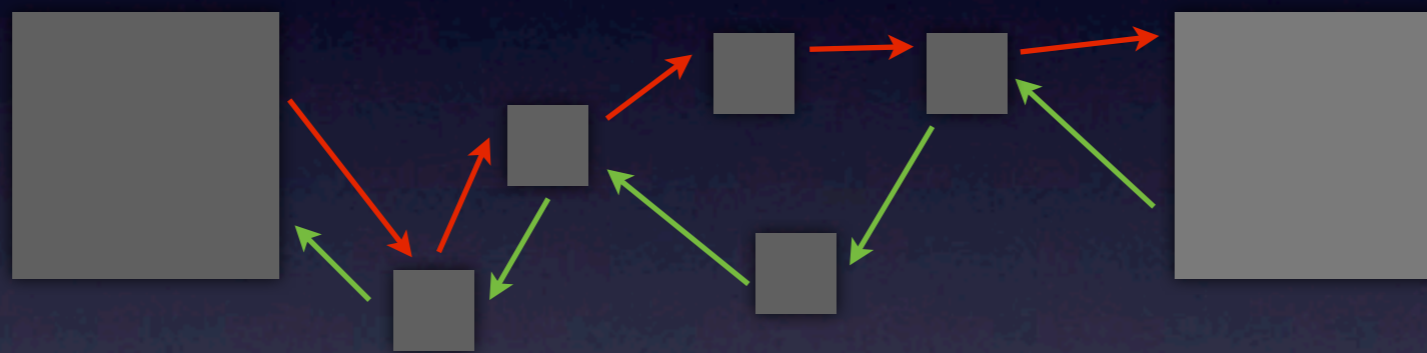
Web Service

My Server

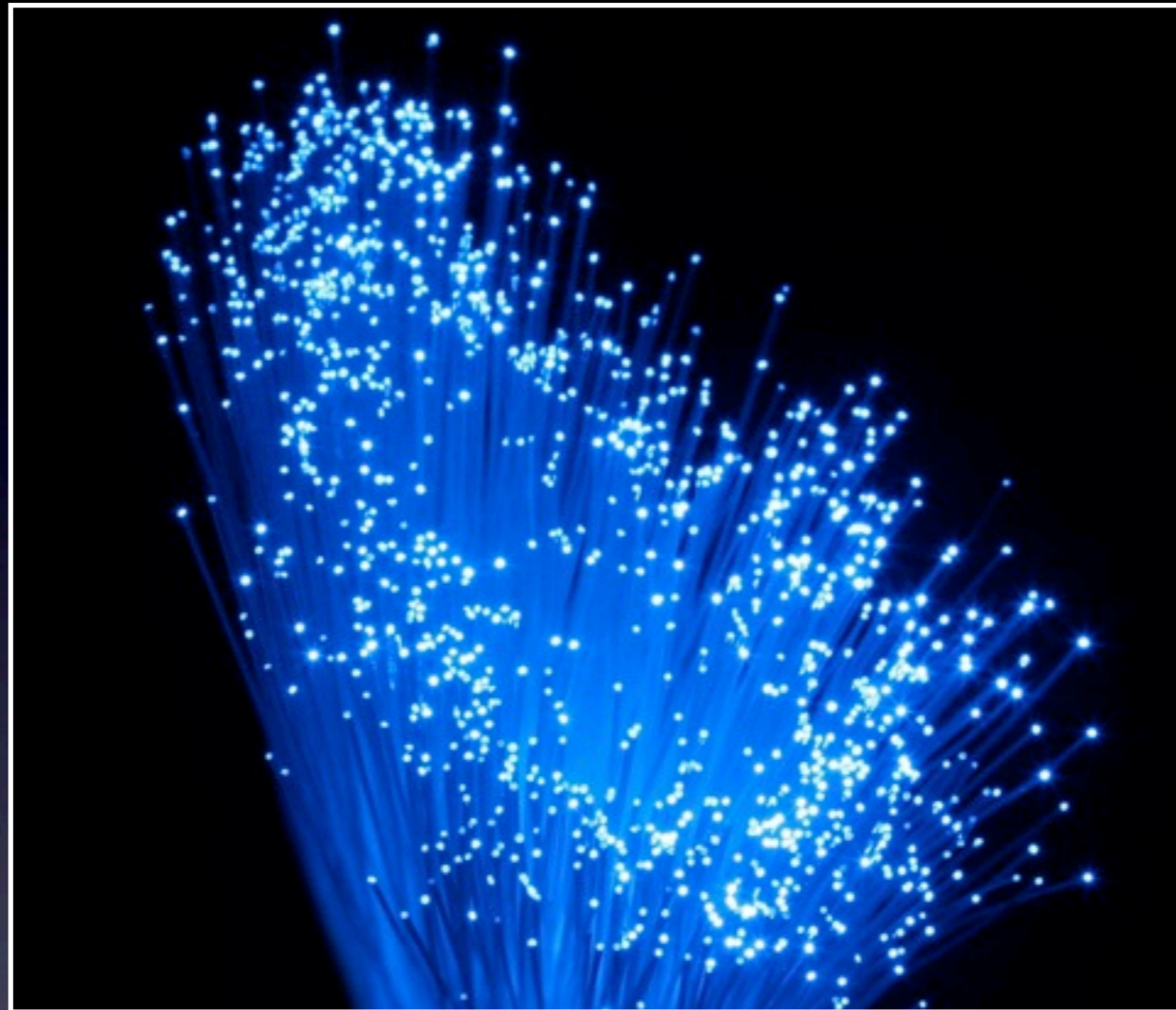


# My Server

# Web Service



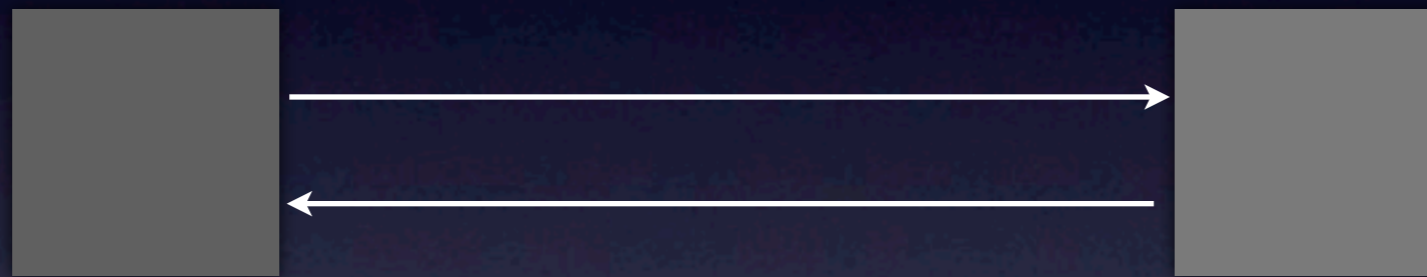




# Optical Fibre

Web Service

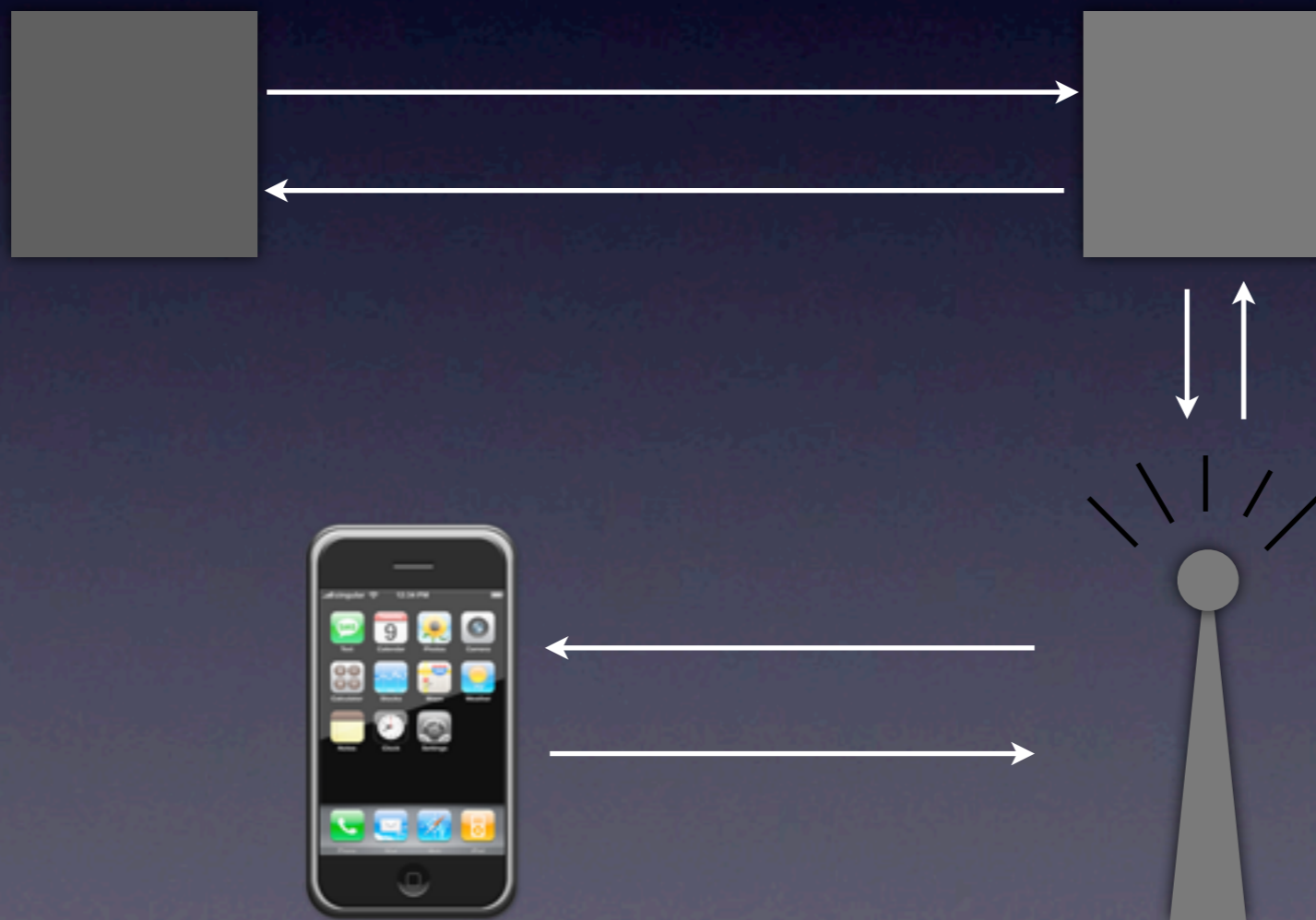
AT&T





Web Service

AT&T

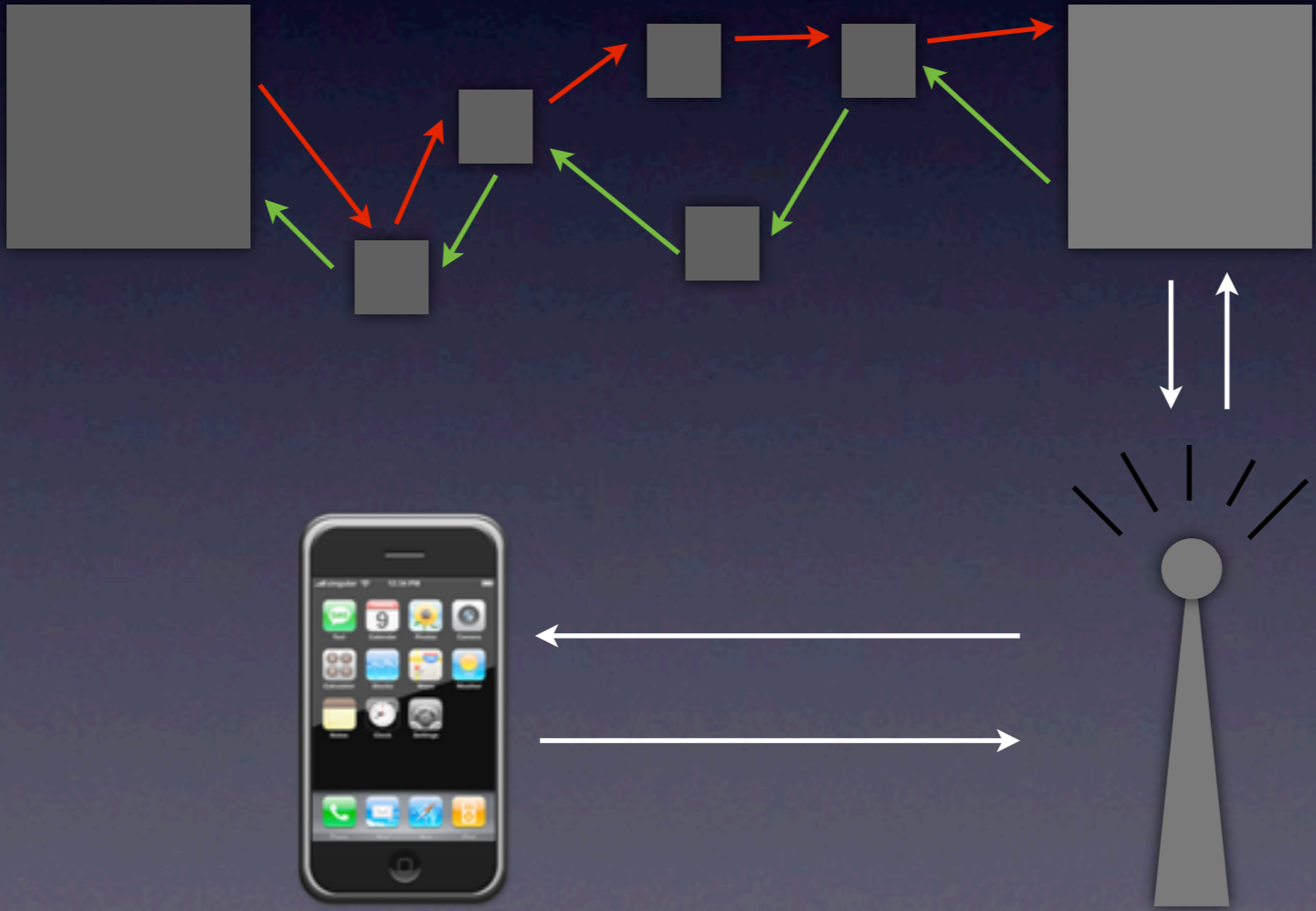




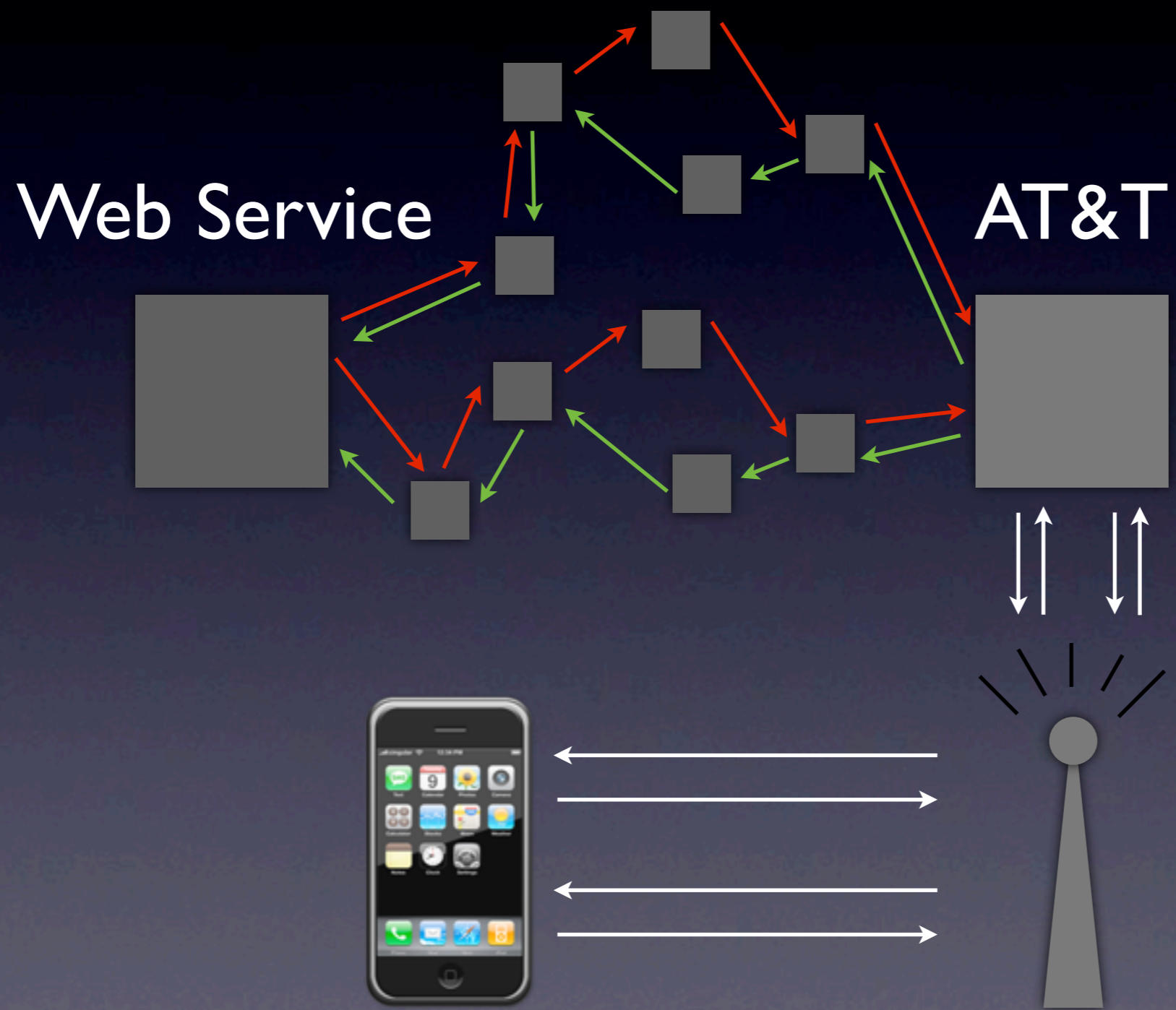
# Radio Waves

Web Service

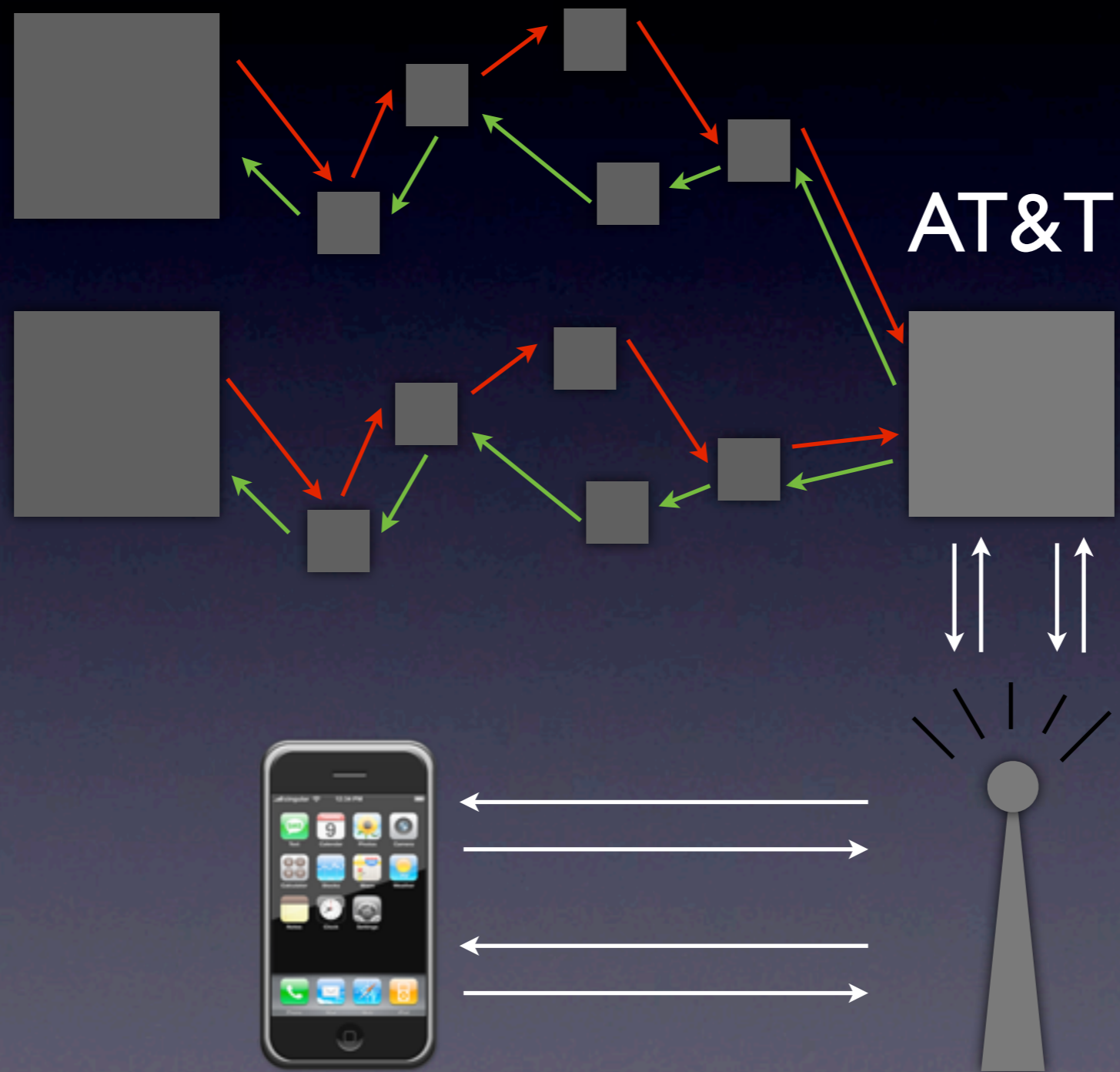
AT&T







# Web Services



# Bigger than mobile?

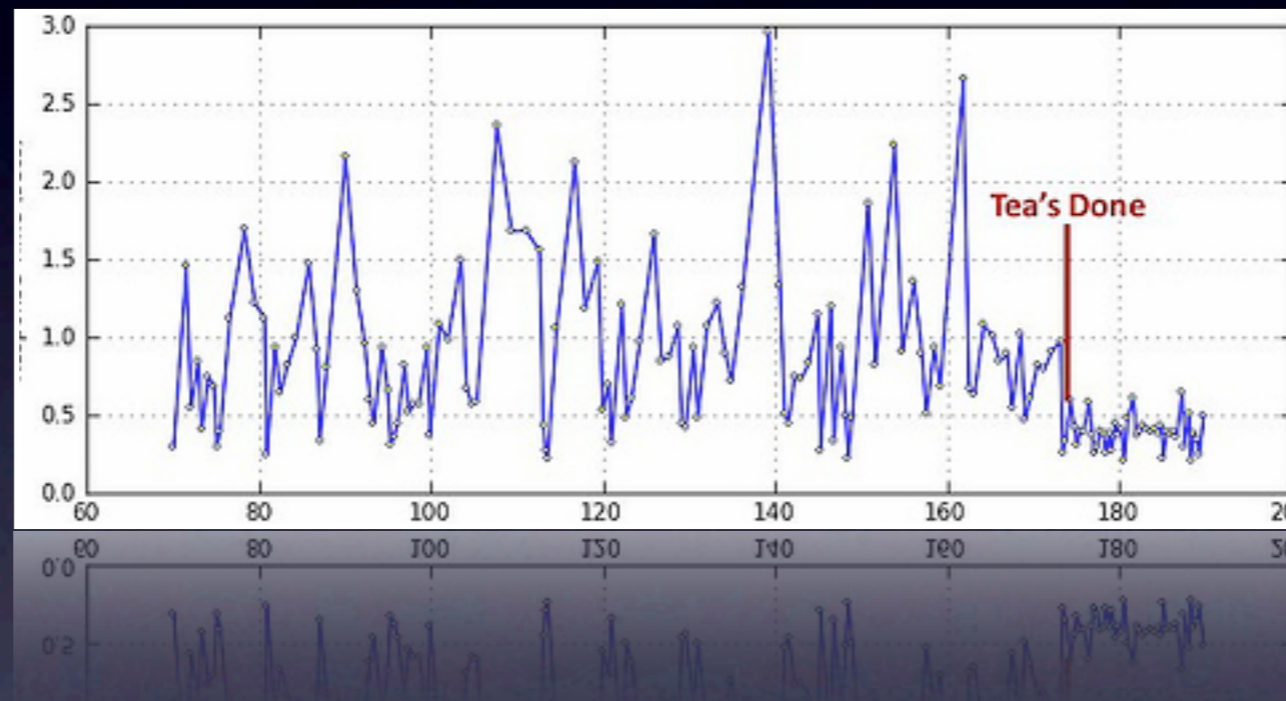


Packet Loss Measurement Device



Packet Loss Generator





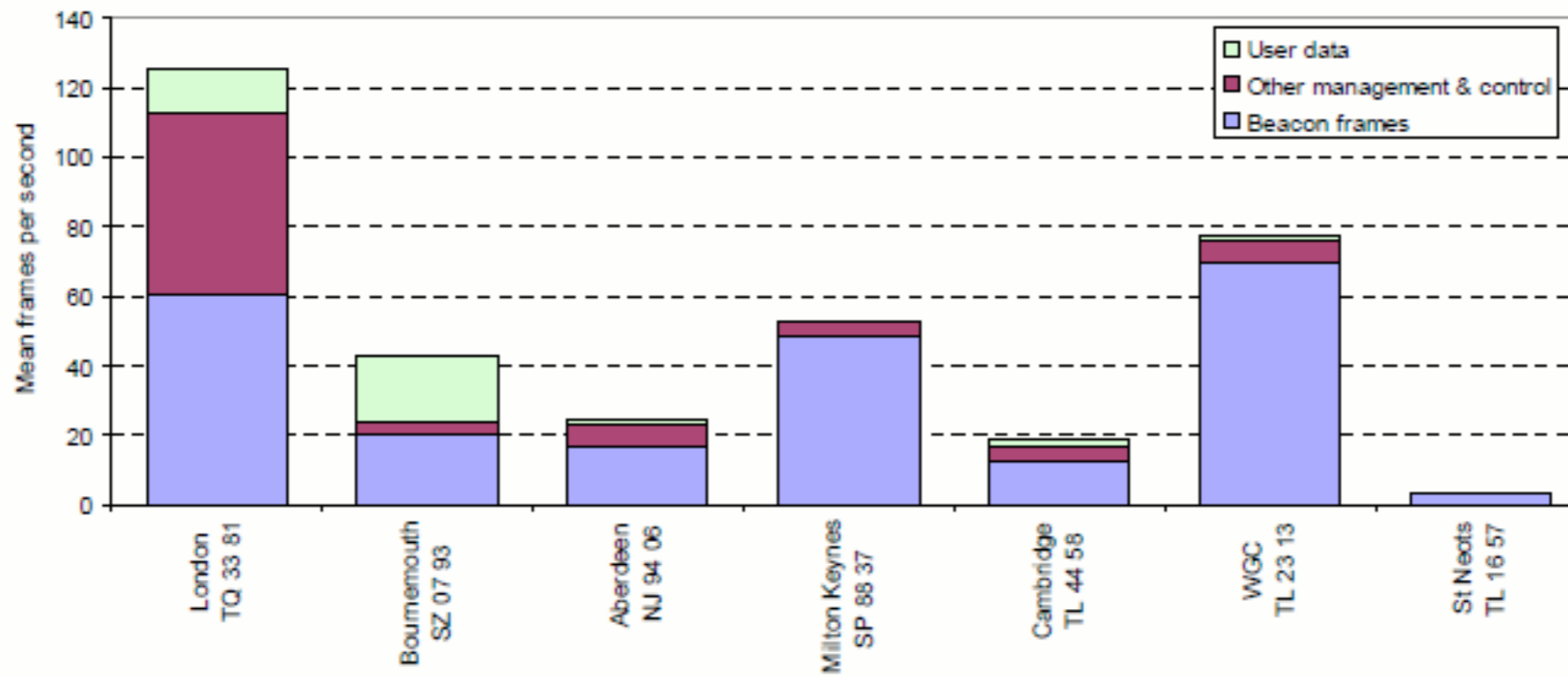
Wifi sucks.



# Ofcom study

- UK study into Wifi interference (PDF)
- Upto 90% of bandwidth used on protocol
- Many other sources of interference
  - Baby monitors
  - Microwaves
  - Cordless Phones





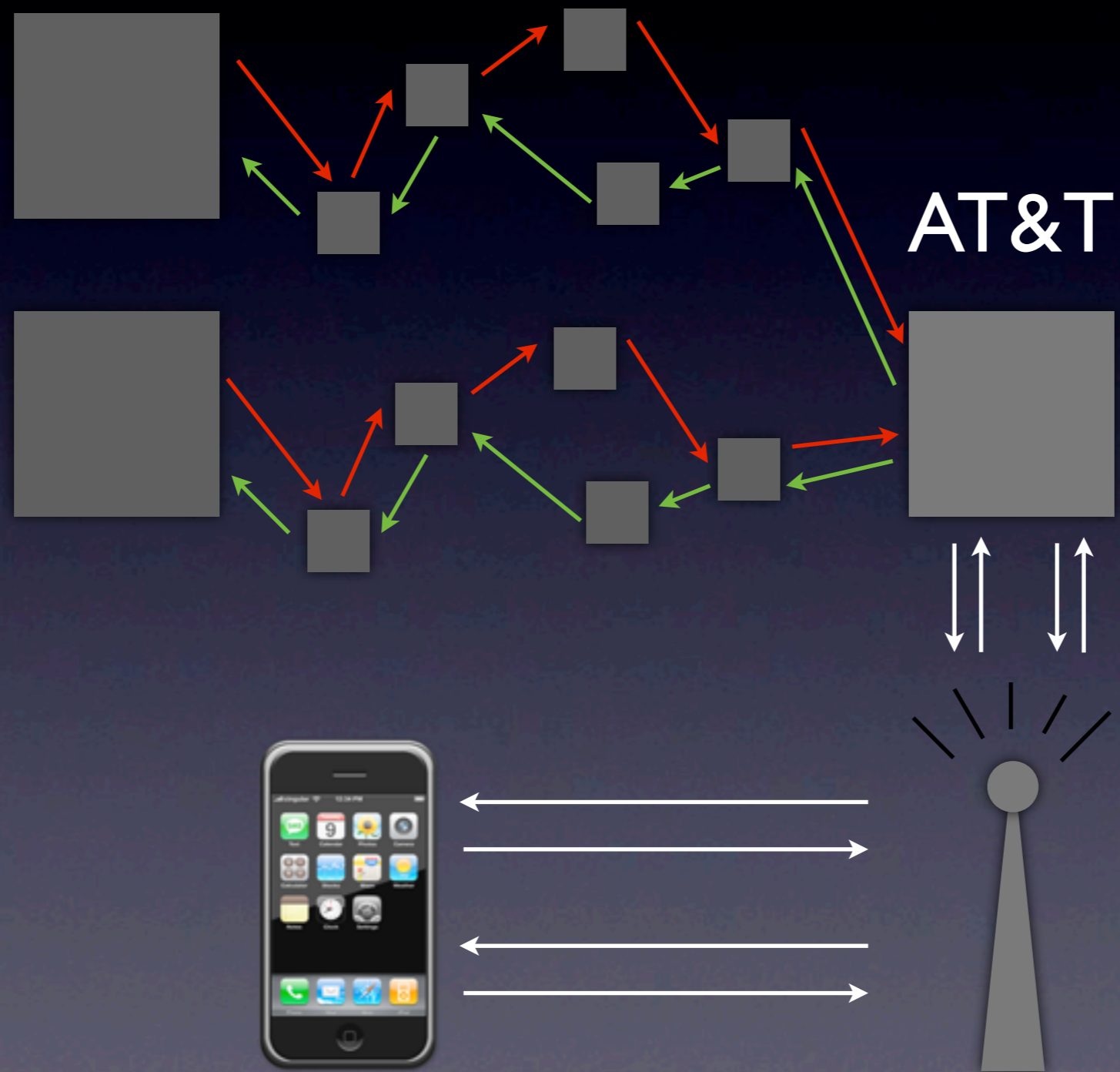
Interference sucks.

So how do we fix it?

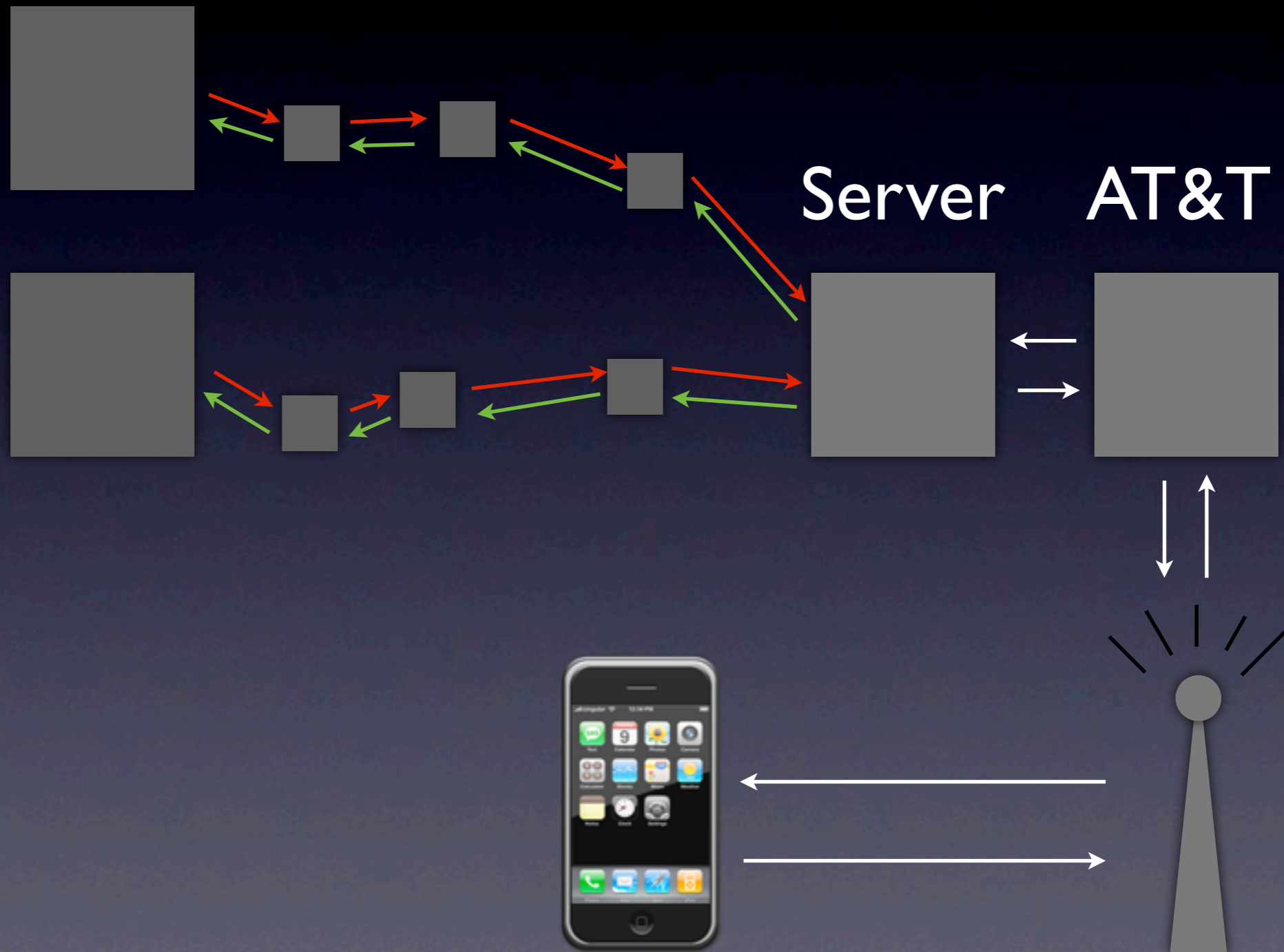
# I. Reduce Requests



# Web Services



# Web Services



# Reducing requests

- Yahoo! Performance Guidelines
  - <http://developer.yahoo.com/performance>
- Reduce web service requests
  - Bundle requests from device logically
  - Gzip all web service responses



# 2. Shape to packets

# Indiana Packet and the Byte of Doom





1 extra byte means  
1 extra packet

Reduce number of  
packets

# Packet sizes

- Make sure static assets (JS, images, etc) fit in discrete packet boundaries
  - Don't forget Gzip
- Try and figure out 'high latency' connections to send inlined assets to
  - E.g. detect 'iPhone'



# Packet Research

[http://developer.yahoo.net/blog/archives/2009/10/  
a\\_engineers\\_gui.html](http://developer.yahoo.net/blog/archives/2009/10/a_engineers_gui.html)

- What is YQL?
- Why is YQL *bad-ass* for mobile development?
- How you can use YQL in mobile development
- Examples

What is YQL?

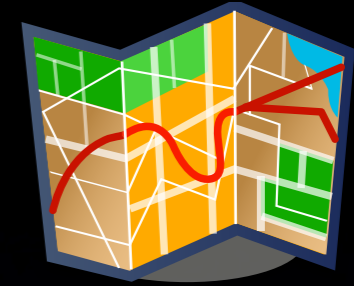
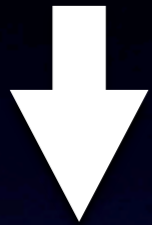


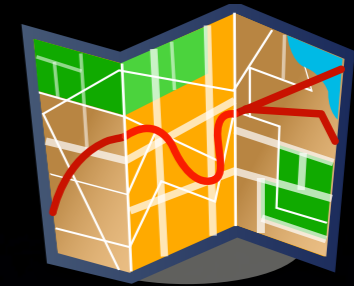
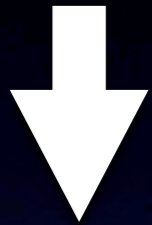


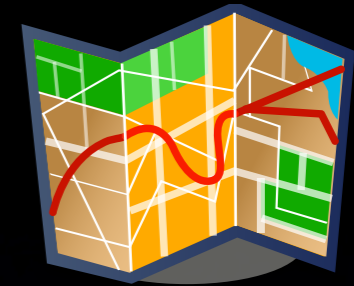




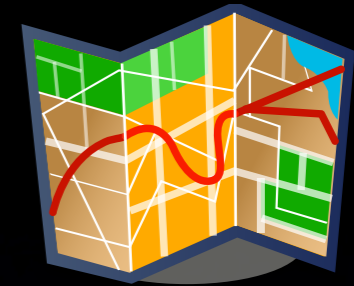
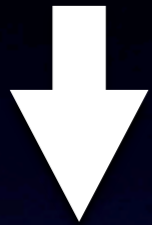


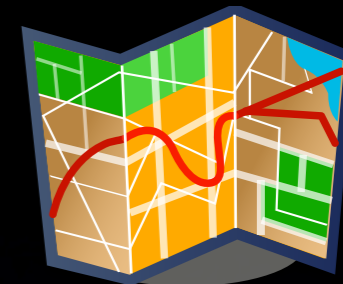


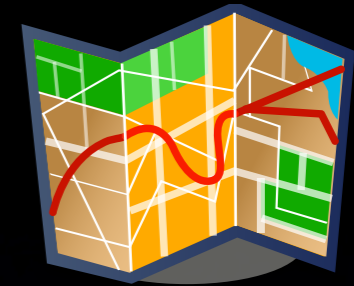




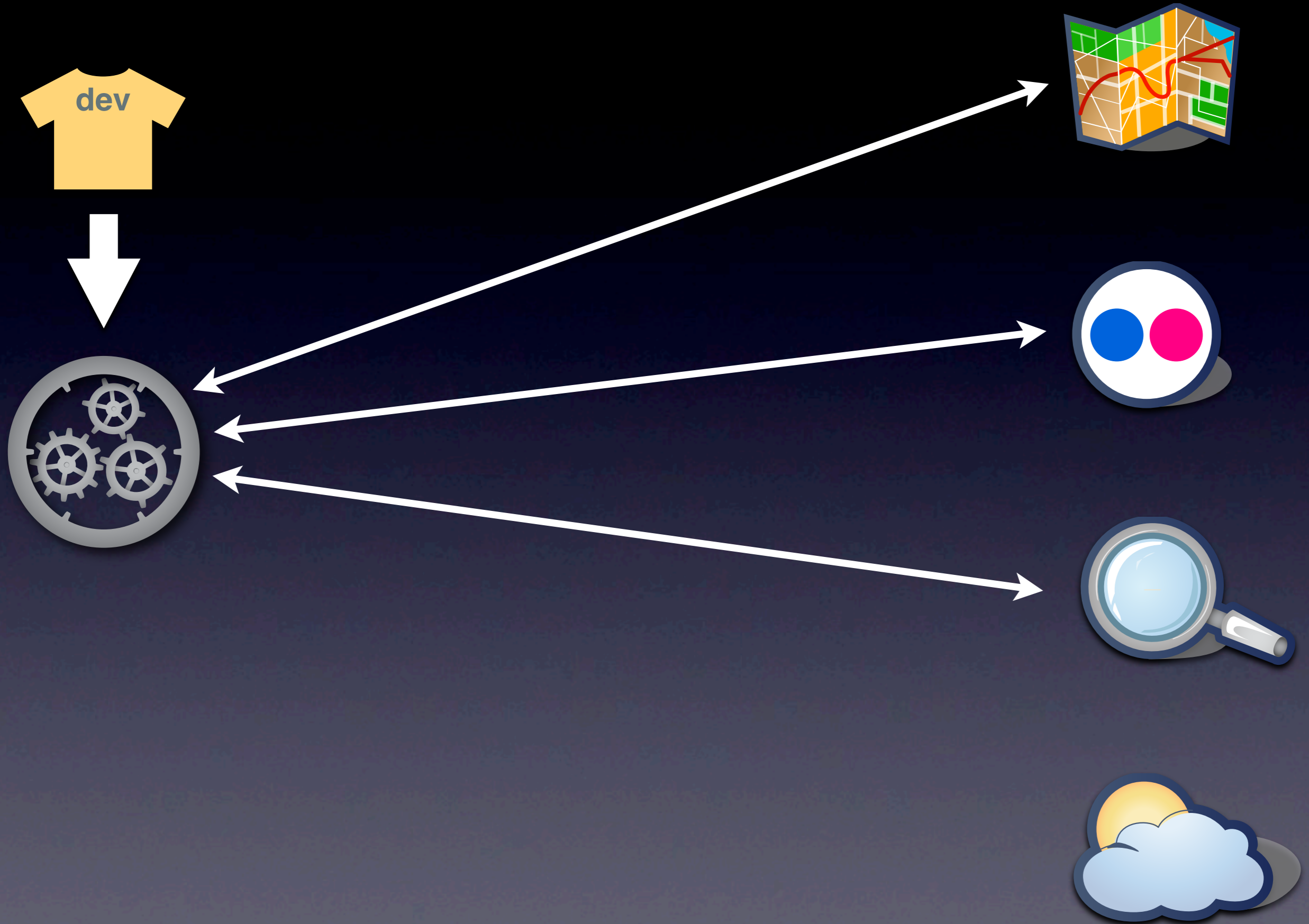


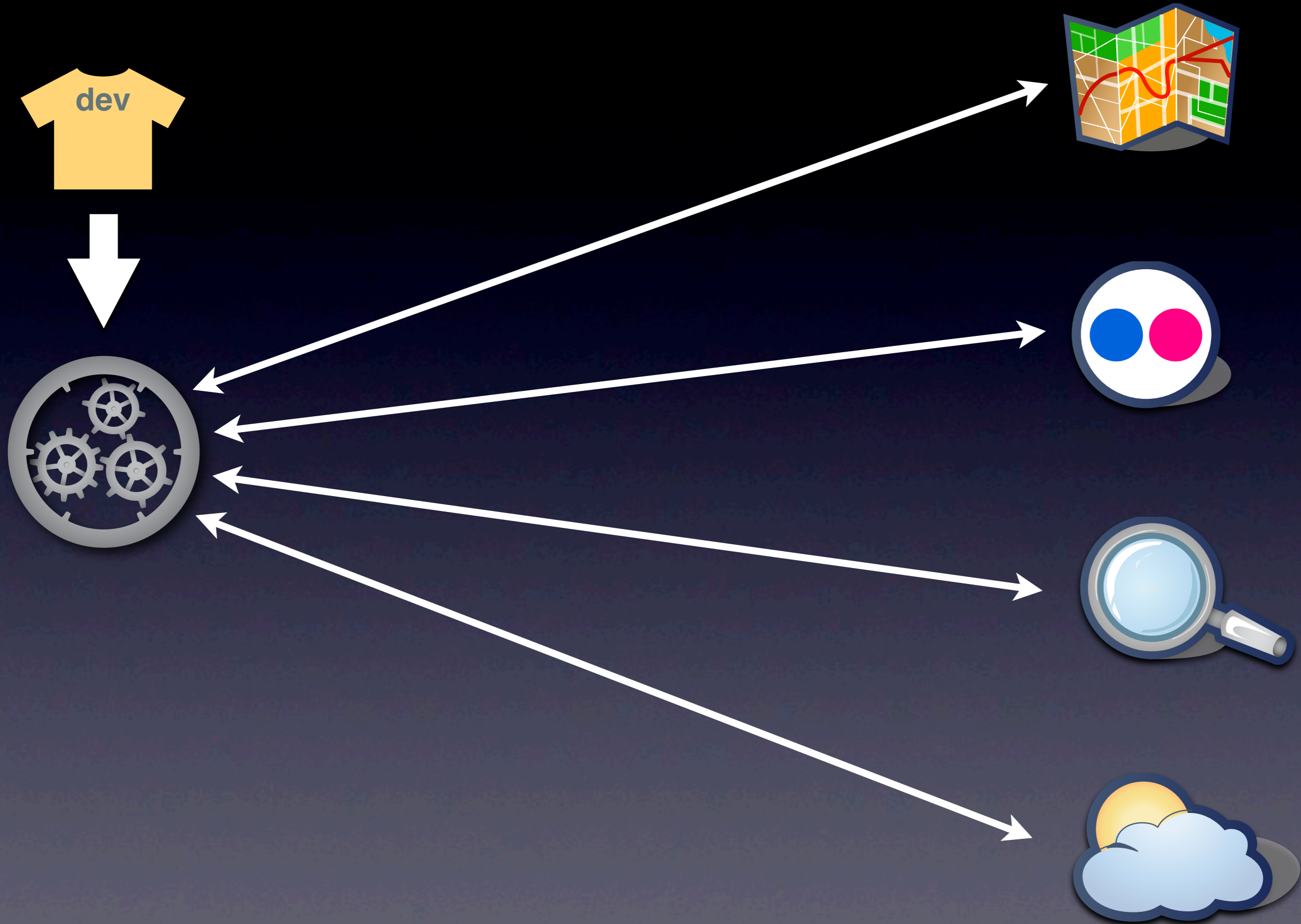


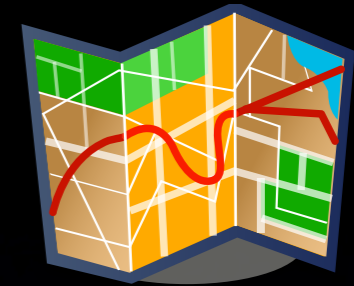




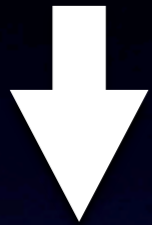




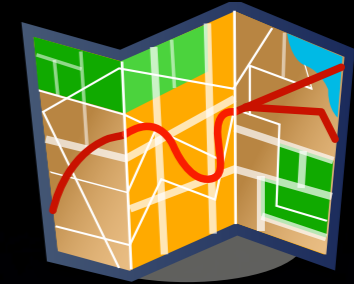


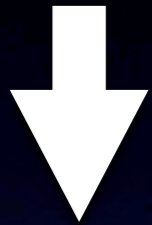






```
var map = new  
YMap(document.getElementById( 'map' ));  
...  
var currentGeoPoint = new  
YGeoPoint( _c.Lat, _c.Lon );  
map.addMarker(currentGeoPoint);  
...
```

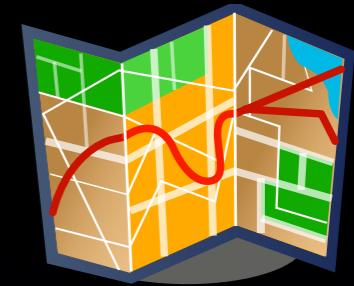




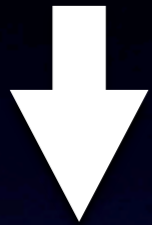
```
var map = new
YMap(document.getElementById( 'map' ));
...
var currentGeoPoint = new
YGeoPoint( _c.Lat, _c.Lon );
map.addMarker(currentGeoPoint);
...

...
if (flickrcurl_prepare(fc,
"flickr.photos.geo.correctLocation",
parameters, count)) { ... }
...

```



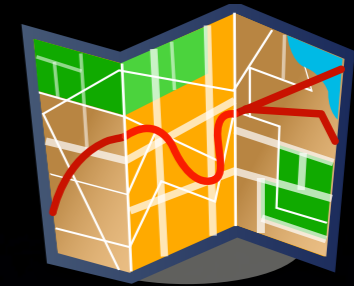




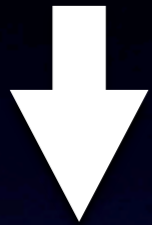
```
var map = new  
YMap(document.getElementById( 'map' ));  
...  
var currentGeoPoint = new  
YGeoPoint( _c.Lat, _c.Lon );  
map.addMarker(currentGeoPoint);  
...
```

```
...  
if (flickrcurl_prepare(fc,  
"flickr.photos.geo.correctLocation",  
parameters, count)) { ... }  
...
```

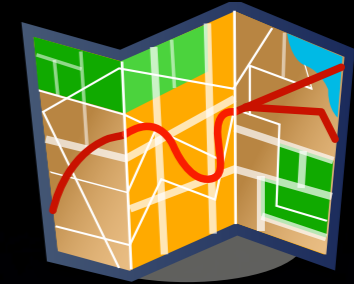
```
http://search.yahooapis.com/  
ImageSearchService/V1/imageSearch?  
appid=YahooDemo&query=Corvette&results=2
```







```
var map = new
YMap(document.getElementById( 'map' ));
...
var currentGeoPoint = new
YGeoPoint( _c.Lat, _c.Lon );
map.addMarker(currentGeoPoint);
...
```



```
...
if (flickrcurl_prepare(fc,
"flickr.photos.geo.correctLocation",
parameters, count)) { ... }
...
```

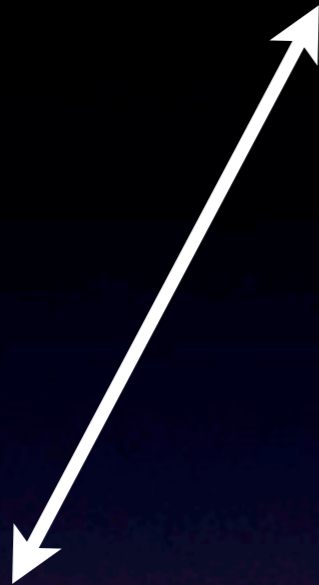


```
http://search.yahooapis.com/
ImageSearchService/V1/imageSearch?
appid=YahooDemo&query=Corvette&results=2
```



```
http://weather.yahooapis.com/
forecastrss?p=FRXX0076&u=c
```



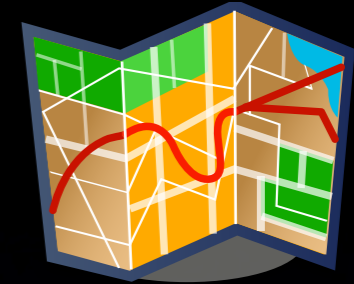


```
var map = new
YMap(document.getElementById( 'map' ));
...
var currentGeoPoint = new
YGeoPoint( _c.Lat, _c.Lon );
map.addMarker(currentGeoPoint);
...
```

```
...
if (flickrcurl_prepare(fc,
"flickr.photos.geo.correctLocation",
parameters, count)) { ... }
...
```

```
http://search.yahooapis.com/
ImageSearchService/V1/imageSearch?
appid=YahooDemo&query=Corvette&results=2
```

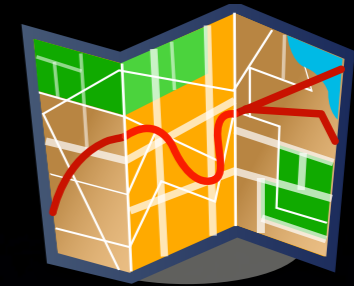
```
http://weather.yahooapis.com/
forecastrss?p=FRXX0076&u=c
```







```
var map = new  
YMap(document.getElementById( 'map' ));  
...  
var currentGeoPoint = new  
YGeoPoint( _c.Lat, _c.Lon );  
map.addMarker(currentGeoPoint);  
...
```



```
...  
if (flickrcurl_prepare(fc,  
"flickr.photos.geo.correctLocation",  
parameters, count)) { ... }  
...
```



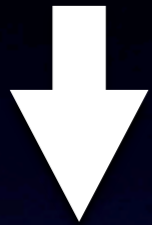
```
http://search.yahooapis.com/  
ImageSearchService/V1/imageSearch?  
appid=YahooDemo&query=Corvette&results=2
```



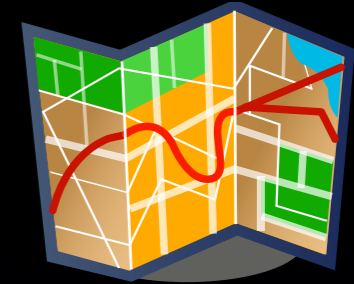
```
http://weather.yahooapis.com/  
forecastrss?p=FRXX0076&u=c
```







```
var map = new
YMap(document.getElementById( 'map' ));
...
var currentGeoPoint = new
YGeoPoint( _c.Lat, _c.Lon );
map.addMarker(currentGeoPoint);
...
```



```
...
if (flickrcurl_prepare(fc,
"flickr.photos.geo.correctLocation",
parameters, count)) { ... }
...
```



```
http://search.yahooapis.com/
ImageSearchService/V1/imageSearch?
appid=YahooDemo&query=Corvette&results=2
```

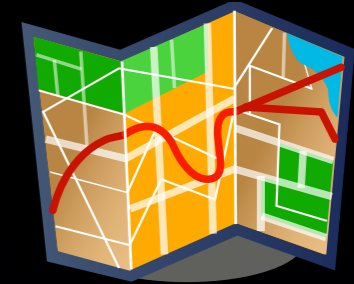


```
http://weather.yahooapis.com/
forecastrss?p=FRXX0076&u=c
```





```
var map = new
YMap(document.getElementById( 'map' ));
...
var currentGeoPoint = new
YGeoPoint( _c.Lat, _c.Lon );
map.addMarker(currentGeoPoint);
...
```



```
...
if (flickrcurl_prepare(fc,
"flickr.photos.geo.correctLocation",
parameters, count)) { ... }
...
```



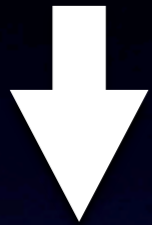
```
http://search.yahooapis.com/
ImageSearchService/V1/imageSearch?
appid=YahooDemo&query=Corvette&results=2
```



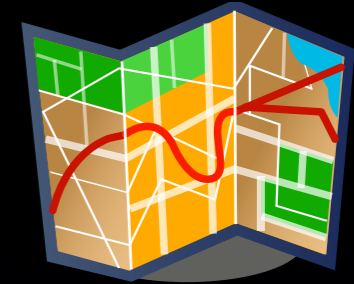
```
http://weather.yahooapis.com/
forecastrss?p=FRXX0076&u=c
```







```
var map = new
YMap(document.getElementById( 'map' ));
...
var currentGeoPoint = new
YGeoPoint( _c.Lat, _c.Lon );
map.addMarker(currentGeoPoint);
...
```



```
...
if (flickrcurl_prepare(fc,
"flickr.photos.geo.correctLocation",
parameters, count)) { ... }
...
```



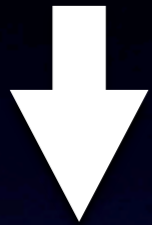
```
http://search.yahooapis.com/
ImageSearchService/V1/imageSearch?
appid=YahooDemo&query=Corvette&results=2
```



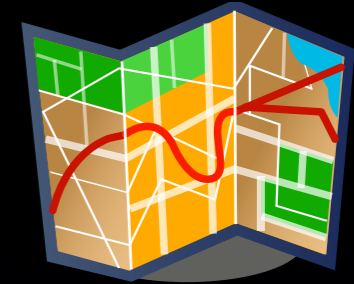
```
http://weather.yahooapis.com/
forecastrss?p=FRXX0076&u=c
```







```
var map = new
YMap(document.getElementById( 'map' ));
...
var currentGeoPoint = new
YGeoPoint( _c.Lat, _c.Lon );
map.addMarker(currentGeoPoint);
...
```



```
...
if (flickrcurl_prepare(fc,
"flickr.photos.geo.correctLocation",
parameters, count)) { ... }
...
```



```
http://search.yahooapis.com/
ImageSearchService/V1/imageSearch?
appid=YahooDemo&query=Corvette&results=2
```



```
http://weather.yahooapis.com/
forecastrss?p=FRXX0076&u=c
```



## Yahoo! Maps Web Services Started Guide

Learn the fundamentals of the Yahoo! Maps Web Services API.

- Introduction and Prerequisites
- Examples
- Display a Yahoo! Map
  - Add controls to your Yahoo! Map
  - Add a logger to a Yahoo! Map
  - Place a marker on a Yahoo! Map
  - Draw Polygons on a Yahoo! Map
  - Place Smart Markers on a Yahoo! Map
  - Add multiple custom markers on a Yahoo! Map
  - Traffic information overlay
  - Yahoo! Pipes RSS output information
  - Yahoo! Pipes JSON output information
  - Local search results overlay
  - Overlay GeoRSS XML data
  - Overlay custom objects at YGeoRSS
  - Overlay custom objects at YGeoRSS
  - Overlay polyline on Yahoo! Map
  - Export Yahoo! Map in GeoRSS
  - GeoRSS overlay with polyline
  - Get zoom level given radius from a point
  - Custom position of controls
  - Change Marker Images Dynamically
  - Use YMapTypeControl to position markers
- Other Features of the Yahoo! Maps API
- Rate Limit
  - Terms of Service
  - Support & Community

Download all of the cheatsheets and Maps API Reference Bundle.

For detailed descriptions of the classes and methods, check out [v3.8 Reference Manual](#).

## Introduction and Prerequisites

The Yahoo! AJAX Maps API lets developers embed maps that are fully embeddable and scriptable. The built-in geocoder means that you can search for a location, as you like.

In order to create and test applications, you need a browser: [Firefox 2](#), [Internet Explorer 7](#), or [Safari 3](#) are supported. You can code your JavaScript in any IDE.

An application ID is required to use the Yahoo! Maps API.

For best results, you should launch your browser directly from your hard drive by double-clicking the Internet Explorer browser or by using the `start` command.

## Displaying a Yahoo! Map

All the API classes and methods need to be included in your page. You include the Yahoo! Maps AJAX API by including the following script in your page:

```

<html>
<head>
<script type="text/javascript" src="http://api.yahooapis.com/MapsAPI/v3/maps-core.js">
  
```

## Flickr Services

API Documentation | Feeds | Your API Key

## flickr.places.getPlaceTypes

Fetches a list of available place types.

### Authentication

This method does not require authentication.

### Arguments

**api\_key** (Required)  
Your API application key. [See here](#) for more information.

### Example Response

```

<place_types>
  <place_type place_type_id="1" name="City" />
  <place_type place_type_id="2" name="Town" />
  <place_type place_type_id="3" name="Village" />
  <place_type place_type_id="4" name="Hamlet" />
  <place_type place_type_id="5" name="City" />
  <place_type place_type_id="6" name="City" />
  <place_type place_type_id="7" name="City" />
</place_types>
  
```

### Error Codes

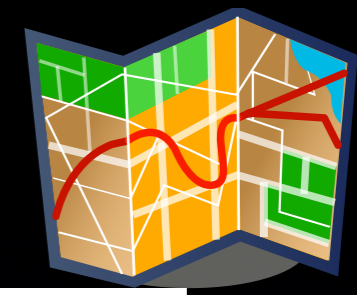
- 100: Invalid API Key**  
The API key passed was not valid or does not exist.
- 105: Service currently unavailable**  
The requested service is temporarily unavailable.
- 111: Format "xxx" not found**  
The requested response format was not found.
- 112: Method "xxx" not found**  
The requested method was not found.
- 114: Invalid SOAP envelope**  
The SOAP envelope send in the request is invalid.
- 115: Invalid XML-RPC Method Call**  
The XML-RPC request document contains an invalid method call.

## API Explorer

[API Explorer : flickr.places.getPlaceTypes](#)

```

tById( 'map' ) );
  
```



## Yahoo! Weather RSS Feed

The Weather RSS feed enables you to get up-to-date weather information for a location. The Weather RSS feed is a dynamic feed that updates every hour.

This page describes the format of the request URL and the response.

- RSS Request
  - Summary
  - Request Parameters
- RSS Response
  - Top-Level Elements
  - Channel Elements
  - Image Elements
  - Item Elements
  - Condition Codes
- Examples
- Terms of Use

### RSS Request

The Weather RSS feed request follows simple HTTP GET syntax. The request consists of the base URL, followed by the parameters and values after a question mark (?).

The base URL for the Weather RSS feed is `http://weather.yahooapis.com/foreca`

In this section, [Summary](#) shows how to make an RSS request. [Request Parameters](#) shows a table summarizing those parameters.

### SUMMARY

The Weather RSS feed request follows simple HTTP GET syntax. The request consists of the base URL, followed by the parameters and values after a question mark (?). The Weather RSS feed there are two parameters:

- `p` for location.
  - `u` for degrees units (Fahrenheit or Celsius).
- The location parameter `p` is required. Use this parameter to specify a location by zip code or Location ID.

`http://weather.yahooapis.com/foreca`

For example, to get weather for Yahoo! Headquarters (94089):

`http://weather.yahooapis.com/foreca`

Or use the Location ID for Sunnyvale (USCA111):

## Image Search Documentation for Yahoo! Search

The Image Search Web Service allows you to search the Internet for images.

### Submitting Image Queries

The Image Search Web Service allows you to search the Internet for images.

### Request URL

`http://search.yahooapis.com/ImageSearchService/V1/imageSearch`

### Request parameters

See information on [constructing REST queries](#)

Parameter	Value	Description
appid	string (required)	The application ID. See <a href="#">Application IDs</a> for more information.
query	string (required)	The query to search for. Use + to include terms, - to exclude terms, and put quotes around "exact phrase".
type	all (default), any, or phrase	The kind of search to submit:  all returns results with all query terms. any returns results with one or more of the query terms. phrase returns results containing the query terms as a phrase.
results	integer: default 10, max 50	The number of results to return.
start	integer: default 1	The starting result position to return (1-based). The finishing position (start + results - 1) cannot exceed 1000.
format	any (default), bmp, gif, jpeg, png	Specifies the kind of image file to search for.
adult_ok	no value or 1	The service filters out adult content by default. Enter a 1 to allow adult content.
coloration	any (default), color, bw	The service returns only the images of the coloration specified (color or black-and-white).
site	string: default no value	A domain to restrict your searches to (e.g. www.yahoo.com). You may submit up to 30 values (site=www.yahoo.com&site=www.cnn.com).
output	string: xml (default), json, php	The format for the output. If json is requested, the results will be returned in JSON format. If php is requested, the results will be returned in Serialized PHP format.
callback	string	The name of the callback function to wrap around the JSON data. The following characters are allowed: A-Z a-z 0-9 . [] and _ . If output=json has not been requested, this parameter has no effect. More information on the

**READY TO GET STARTED?**

By applying for an Application ID for this service, you hereby agree to the [Terms of Use](#)

[YAHOO! GROUPS DISCUSSIONS](#) [view all](#)

[yahoo local search code in php](#)  
Tue, 27 Jan 2009

[My Web Search API shut down announcement](#)  
Wed, 21 Jan 2009

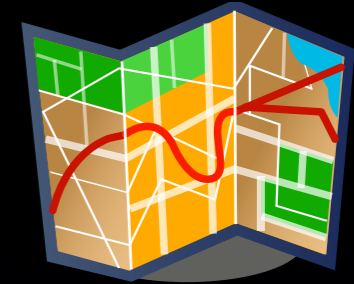
[Re: Site Explorer API returns 999 Rate Limit Exceeded](#)  
Wed, 21 Jan 2009

[Reminder: Old obsolete versions of Local Search \(V1, V2\) shutting do](#)  
Mon, 19 Jan 2009





```
var map = new
YMap(document.getElementById( 'map' ));
...
var currentGeoPoint = new
YGeoPoint( _c.Lat, _c.Lon );
map.addMarker(currentGeoPoint);
...
```



```
...
if (flickrcurl_prepare(fc,
"flickr.photos.geo.correctLocation",
parameters, count)) { ... }
...
```



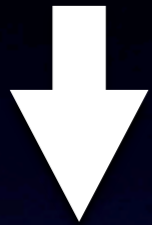
```
http://search.yahooapis.com/
ImageSearchService/V1/imageSearch?
appid=YahooDemo&query=Corvette&results=2
```



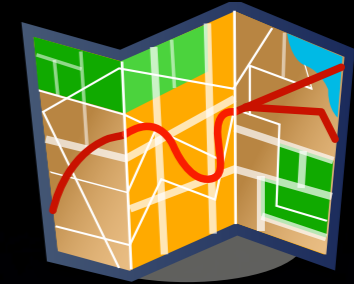
```
http://weather.yahooapis.com/
forecastrss?p=FRXX0076&u=c
```

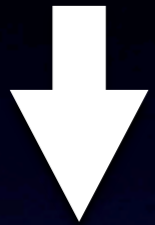




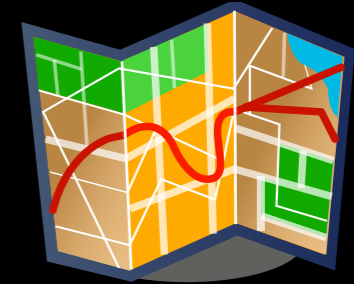


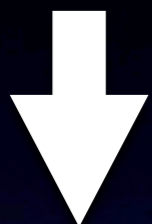
# YQL





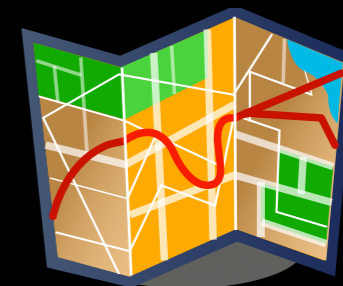
← YQL →





**YQL**

**Bindings**





# YQL

- Unified interface to Web services
- Self-describing to avoid documentation
- Uses a common way to return data

# YQL is like SQL

- Uses SQL verbs
- Uses SQL syntax
- Returns rows of data
- Self-describing
  - show tables;
  - desc table;

# Verbs

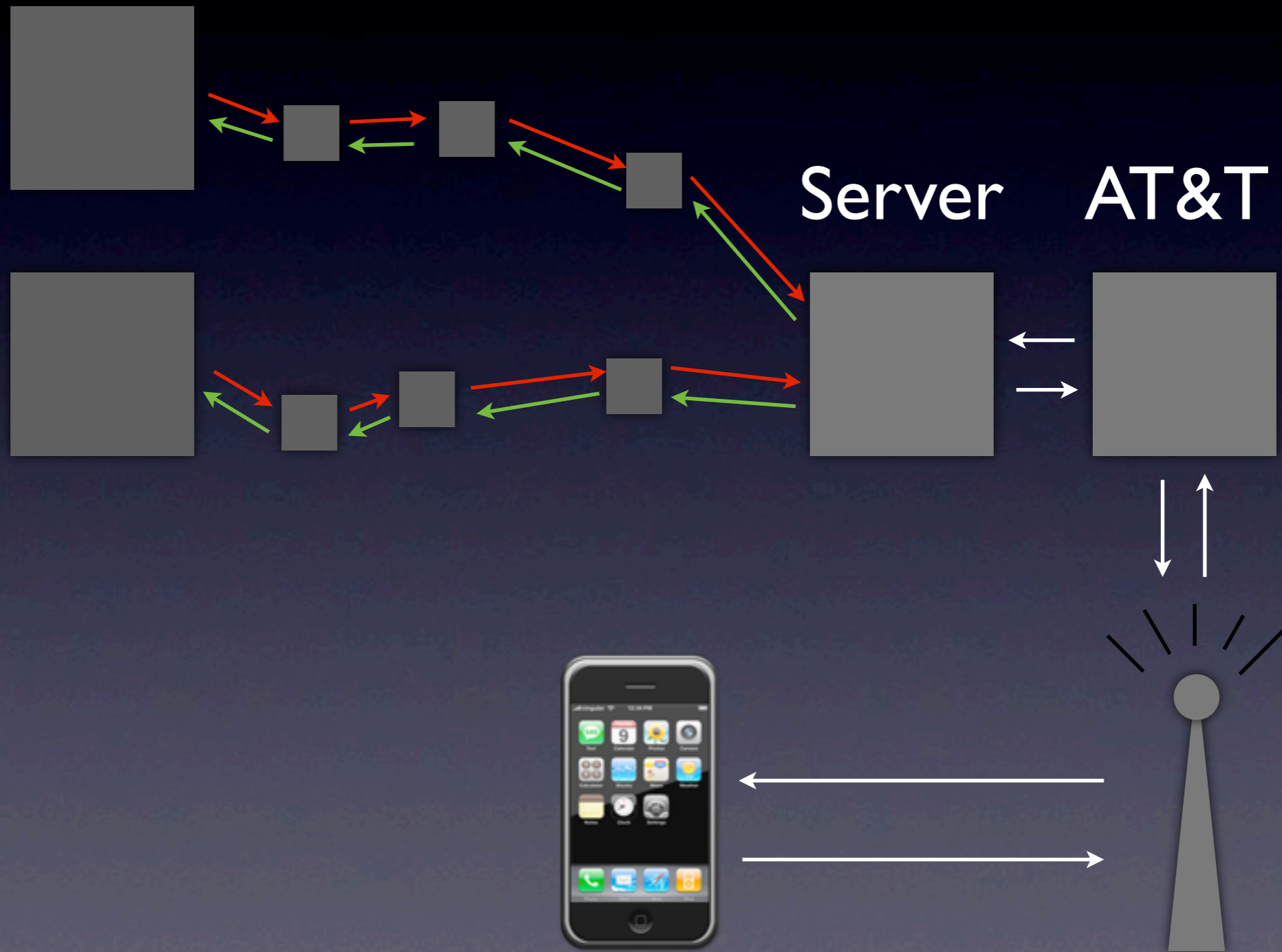
- `show` - *show tables;*
- `desc` - *desc search.web;*
- `select` - *select \* from flickr.photos.recent;*
- `use`
- `insert`
- `update`
- `delete`



# Demo

<http://developer.yahoo.com/yql/console>

# Web Services



# YQL Example



```
select * from flickr.photos.info
where photo_id in (select id from
flickr.photos.search where query =
“lolcats”)
```

# Conclusion

- Latency hurts anything wireless
- Fight latency by reducing connections
- Fight latency by reducing the number of unnecessary packets
- YQL can help you bundle your requests to mobile device without a backend

follow me on  
[twitter.com/sh1mmer](https://twitter.com/sh1mmer)

Slides? Feedback?  
[www.speakerrate.com/sh1mmer](http://www.speakerrate.com/sh1mmer)