Creating airplane mode proof (Xamarin) applications

Gerald Versluis

Gerald Versluis

@jfversluis

blog.verslu.is

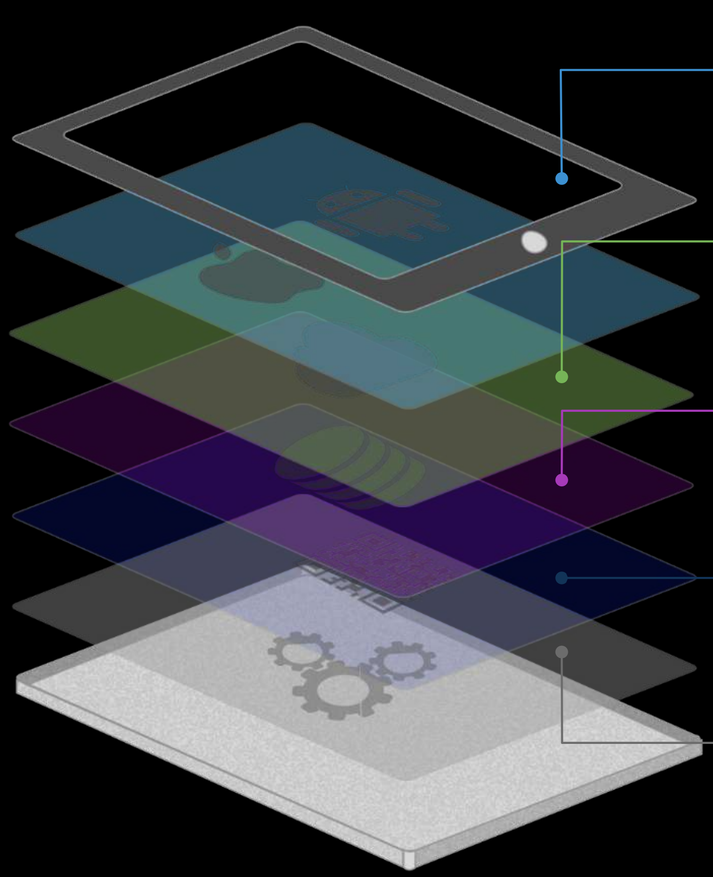
gerald@verslu.is

DOT
CONT
ROL.

“Our app doesn’t even have to *be* fast, it’s enough
that the user **thinks** it is fast”

@jfversluis

Layers of Mobile Development



Thousands of iOS and Android APIs

Public web APIs

Backend integrations

Third party libraries

Memory, CPU and network constraints

@jfversluis

Network constraints

- Mobile connection not always stable
- Bandwidth costs money
- Network calls take time

The eight fallacies

https://en.wikipedia.org/wiki/Fallacies_of_distributed_computing

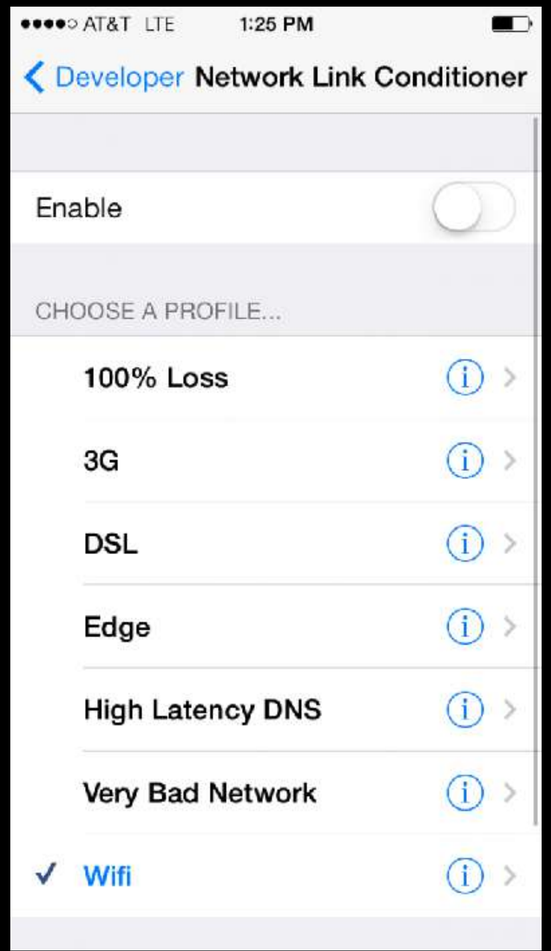
1. The network is reliable.
2. Latency is zero.
3. Bandwidth is infinite.
4. The network is secure.
5. Topology doesn't change.
6. There is one administrator.
7. Transport cost is zero.
8. The network is homogeneous.

@jfversluis

Image source: https://imgur.com/gallery/JQEWkBy

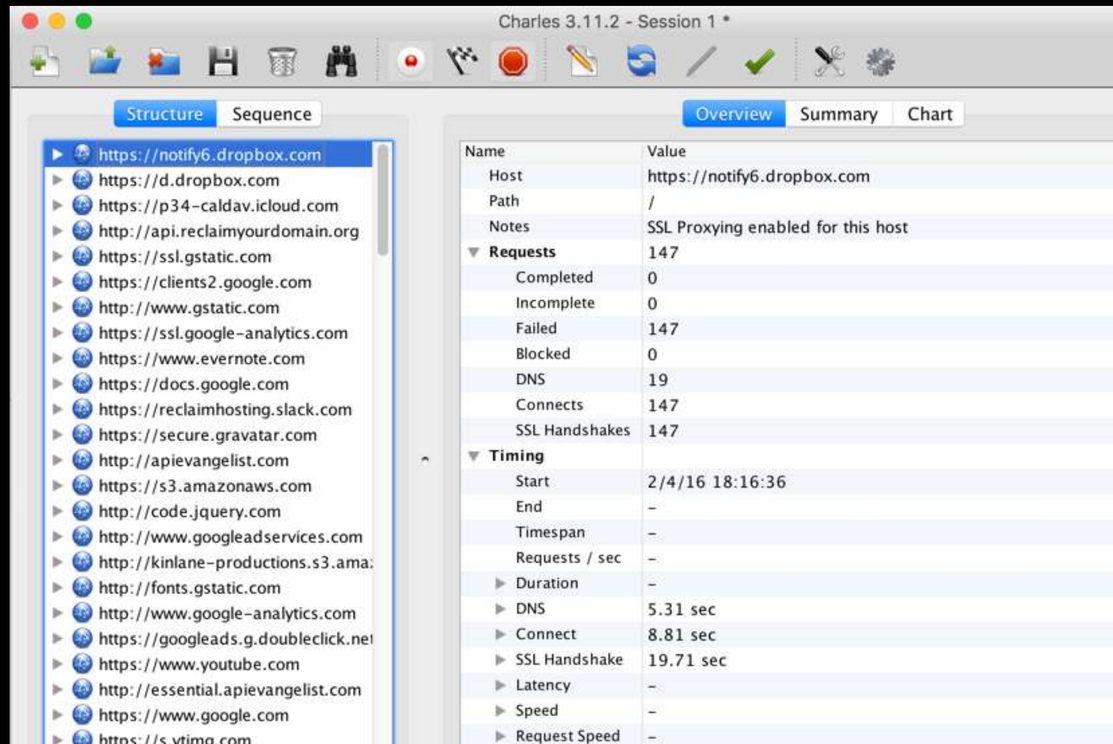
What Options Do We Have?

- Emulator/Simulator have options to mimic network conditions
  - Network Link Conditioner (iOS) https://developer.apple.com/download/more/?q=Additional%20Tools
  - Or on your physical device 👉
  - Built into the Android Emulator (Stock/Genymotion)



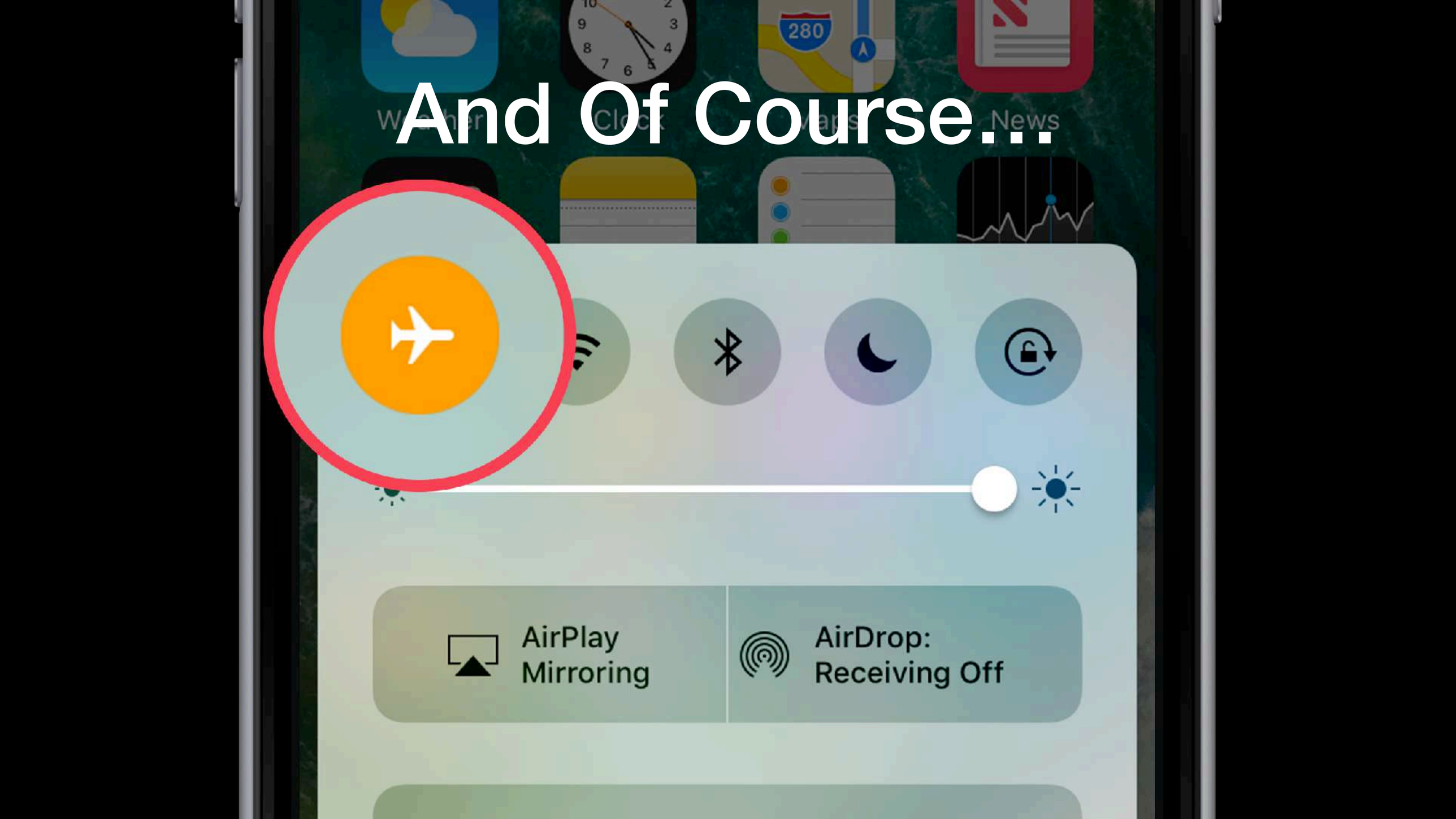
What Options Do We Have?

- Monitoring tool to see what is going over the line (or, you know over the air..)
  - Charles (Windows/Mac/Linux)
  - Fiddler (Windows/Mac (Mono))
  - Wireshark (Windows/Mac)
    (if you like pain)
  - HTTPScoop (Mac)



@jfversluis

And Of Course...



All code can be found on:

https://github.com/jfversluis/AirplaneModeProof

@jfversluis

Image source: http://getwallpapers.com/collection/matrix-code-wallpaper-hd

Version 1.0

The basics

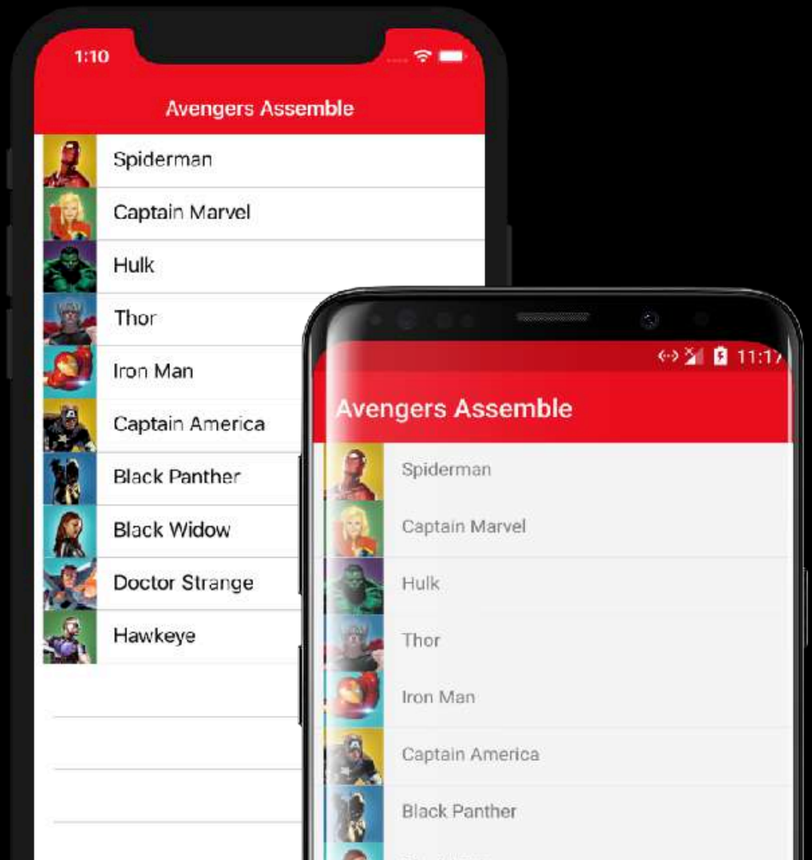
@jfversluis

Image source: https://www.themarysue.com/avengers-age-of-ultron-easter-eggs/

Avengers Assemble!

https://github.com/jfversluis/AirplaneModeProof

- Sample App
- Access information about each member
- See in which comics they appear
- Xamarin.Forms & ASP.NET Core backend (on Azure)



@jfversluis

Version 2.0

Quick Wins!

@jfversluis

Image source: http://knowyourmeme.com/memes/success-kid-i-hate-sandcastles

FFImageLoading

https://github.com/luberda-molinet/FFImageLoading

- Fast & Furious Image Loading
- Can cache images in-memory and on disk
- Deduplication of similar download/ load requests
- Retry image downloads

FFImageLoading

https://github.com/luberda-molinet/FFImageLoading

- Has more awesome features:
  - Transformations
  - SVG / WebP / GIF support
  - Loading and error placeholders
- Don’t forget **Init();**!

Connectivity Plugin

https://github.com/jamesmontemagno/ConnectivityPlugin

- Check connection status
- Information about connection type, bandwidth, etc.
- Has methods to see if a certain host is reachable
- Has events to catch changes in state

Refit

https://github.com/paulcbetts/refit

- No more duplicate code to do REST calls 🎉
- Implement the contract as an interface
- Calls will be generated at compile time
- Not just for Xamarin!

Version 3.0

Adding Akavaching

@jfversluis

Image source: http://www.douglasequipment.com/hand-trucks/heavy-duty-warehouse-trucks-miami/

Implement Caching the Hard Way

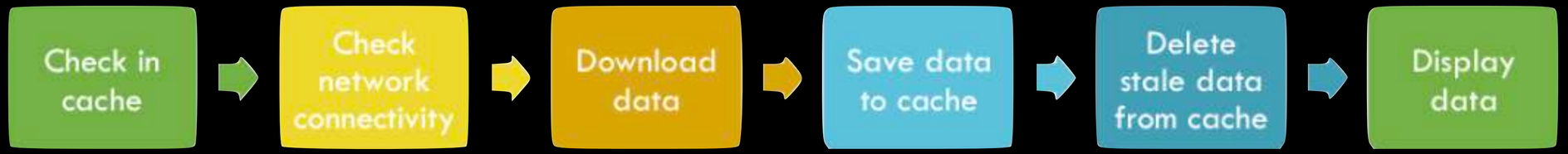


Image source: https://codemilltech.com/akavache-is-aka-awesome/

Akavache

https://github.com/akavache/Akavache

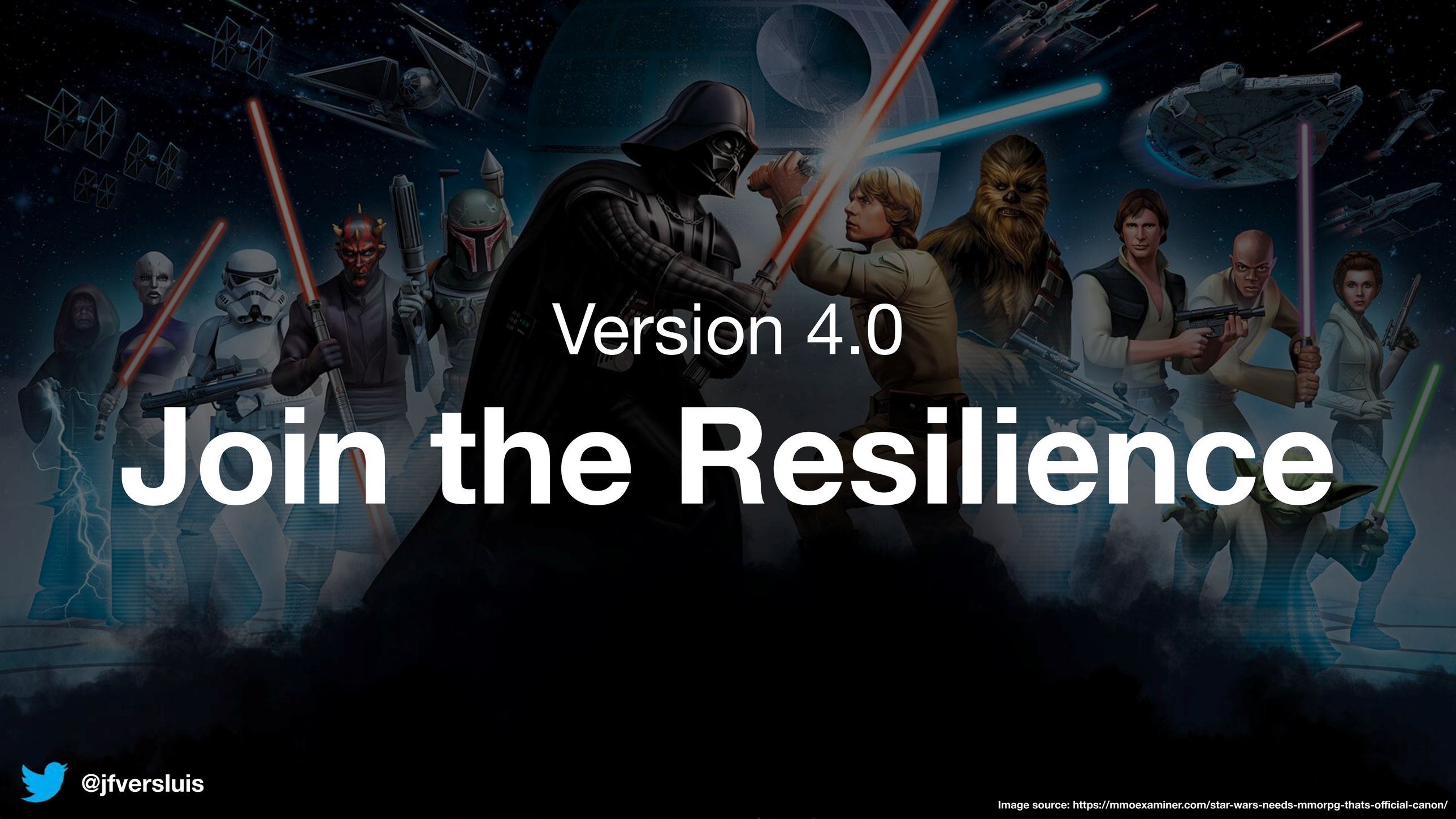
- “Asynchronous, persistent key-value store based on SQLite3”
- Caching is important
  - Network resiliency
  - Perceived app performance
- Not just for Xamarin!

Akavache: Lessons Learned

https://github.com/akavache/Akavache

- Add the **Akavache.SQLite3** package and make sure you have the **AkavacheSqliteLinkerOverride.cs** file included
- Shut down properly with: **BlobCache.Shutdown().Wait();**
  - On mobile a bit hard because of the lifecycle
  - Wrap in a repository pattern
- Alternative: Monkey Cache by James Montemagno

@jfversluis



Version 4.0

Join the Resilience

@jfversluis

Image source: https://mmoexaminer.com/star-wars-needs-mmorpg-thats-official-canon/

Polly

https://github.com/App-vNext/Polly

- ".NET resilience and transient-fault-handling library"
- Defining policies for failed requests
- More info: http://www.thepollyproject.org/
- Not just for Xamarin!

Polly

Policy	Premise	Aka	Mitigation
Retry	Many faults are transient and may self-correct after a short delay.	"*Maybe it's just a blip*"	Allows configuring automatic retries.
Circuit-breaker	When a system is seriously struggling, failing fast is better than making users/callers wait. Protecting a faulting system from overload can help it recover.	"*Stop doing it if it hurts*" or *“Give that system a break”*	Breaks the circuit (blocks executions) for a period, when faults exceed some
Timeout	Beyond a certain wait, a success result is unlikely.	"*Don't wait forever*"	Guarantees the caller won't have to wait beyond the timeout.
Fallback	Things will still fail - plan what you will do when that happens.	"*Degrade gracefully*"	Defines alternative value to be returned (or action to be executed) on failure.

@jfversluis

Source: https://github.com/App-vNext/Polly/blob/master/README.md

Version 5.0

HTTP Conditional Requests

@jfversluis

Image source: https://mmoexaminer.com/star-wars-needs-mmorpg-thats-official-canon/

HTTP Conditional Requests

https://developer.mozilla.org/en-US/docs/Web/HTTP/Conditional_requests

- “HTTP conditional requests are requests that are executed differently, depending on the value of specific headers.”
- Only returns data if something changed
- **If-Match**, **If-None-Match**, **If-Modified-Since**, **If-Modified-Since** and **If-Range**
- Most common use-case is caching

Take-aways

- Write less redundant code with Refit
- Keep in mind the limitations of a mobile device
- Give the user the perception of speedy loading
- More optimizations possible with **If-None-Match** and/of **If-Modified-Since**

@jfversluis

Heads-up! Get your camera out to take pictures of the resources links on the next slide

Resources

- https://robgibbens.com/resilient-network-services-with-xamarin/
- https://codemilltech.com/akavache-is-aka-awesome/
- https://xamgirl.com/consuming-a-restful-web-service-in-xamarin-forms-using-refit-part-1/
- https://xamarinhelp.com/resilient-network-connectivity-xamarin-forms/

@jfversluis

Gerald Versluis

Thank you!

@jfversluis

gerald.verslu.is

blog.verslu.is

gerald@verslu.is

DOT
CONT
ROL.