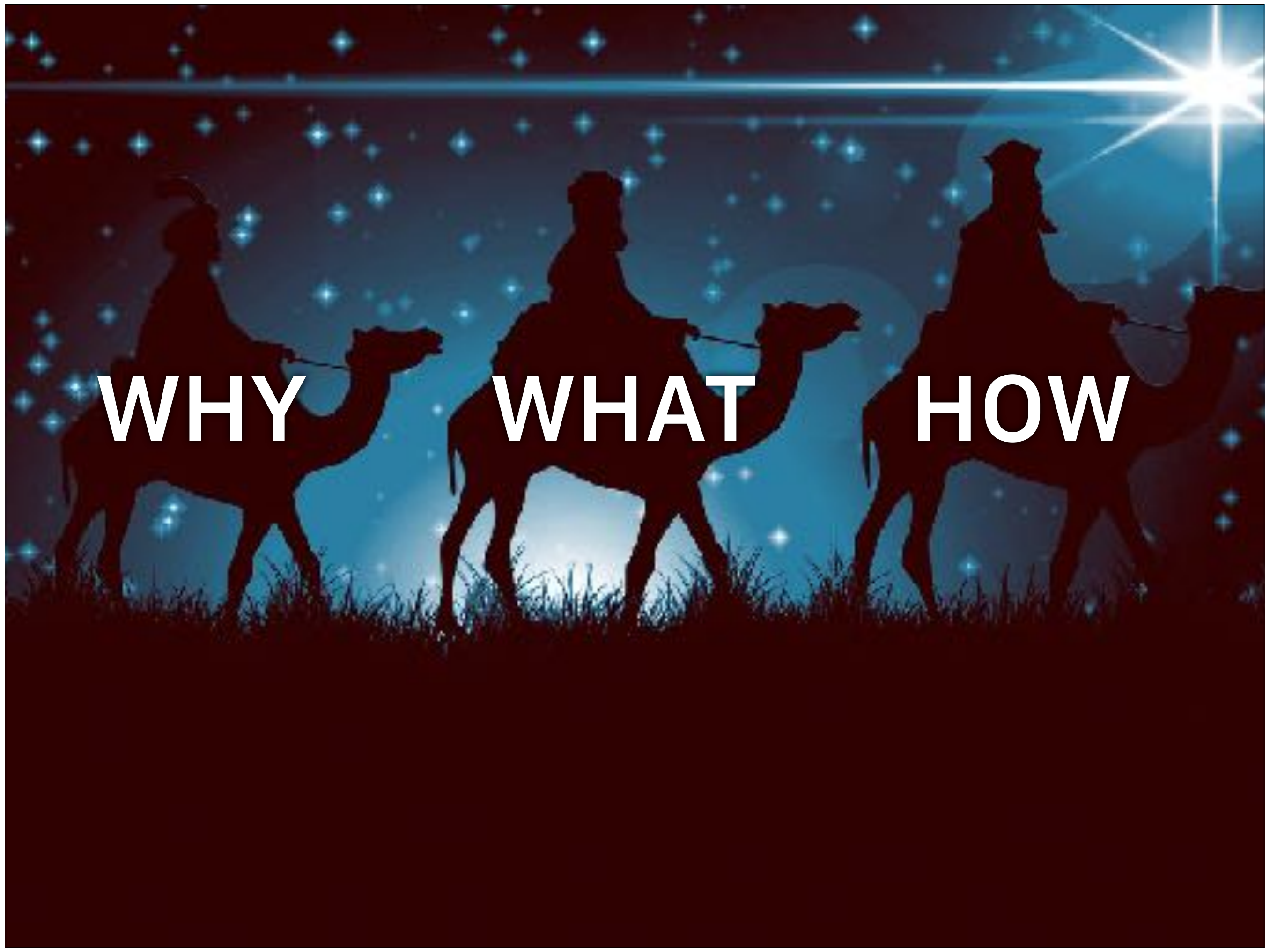
VICTOR AUGUSTEO

EASY TESTING WITH MOCK SERVER

How’s your long weekend? welcome to the best talk of the day with your favourite speaker, yours truly.

WHO AM I?

- Victor Augusteo
- Senior iOS Dev @ Tabcorp
- Mostly Apple and web apps
- augusteo@gmail.com
- www.augusteo.com



WHY WHAT HOW

we are going to discuss about why, what and how of mock server. the why and what will be quite brief as i’d like to spend more time demonstrating how easy it is to get started. of course if you have questions, please raise your hand anytime.

now why do we even need this? what problem do we face now that requires a mock server?

practically everyone deploys on multiple platform nowadays. java on android, js on web, swift on iOS. and unless you do full hybrid like xamarin or react native, you will need to mock the web services for testing. and you have to do it in each and every separate language.

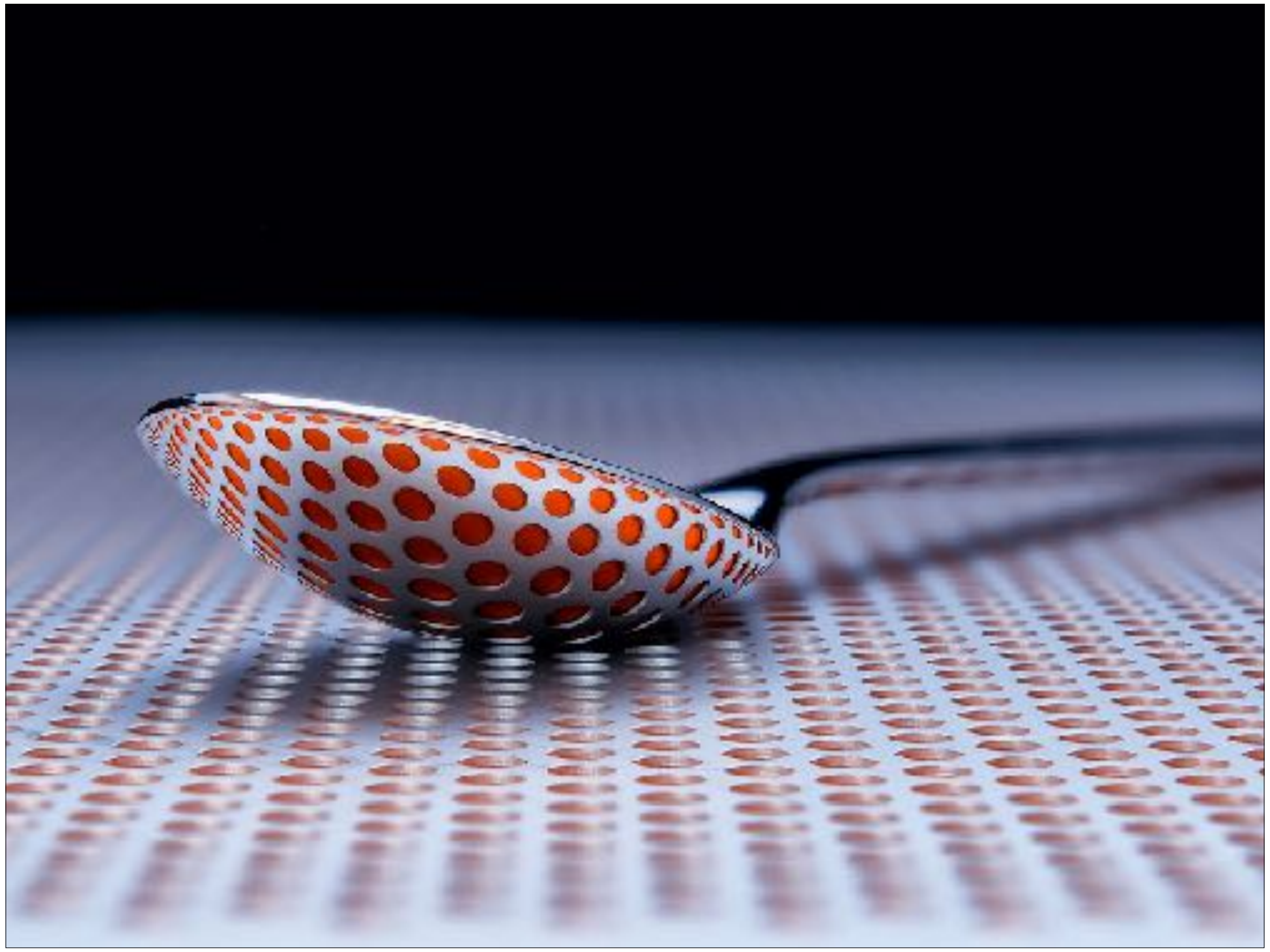
mock server promotes sharing and code reuse, write once and use to test ios/android/web app. no need to write mocking codes several times for different platform. you can just connect to the mock server and done.

I’m sure most of us have been dependent on another team. if the api team is delayed for whatever reason on that shiny new feature, we would be developing in the dark. Mock server allow us to develop against a service that is incomplete or is unstable, essentially reducing dependency and coupling of teams.

you just need to set a contract of request and response format, then implement it quickly on mock server and fire away. you can even ask the api team to write the mock for you!

It is also very helpful to isolate a single req/res for debugging purposes. One good example is from our Keno app.

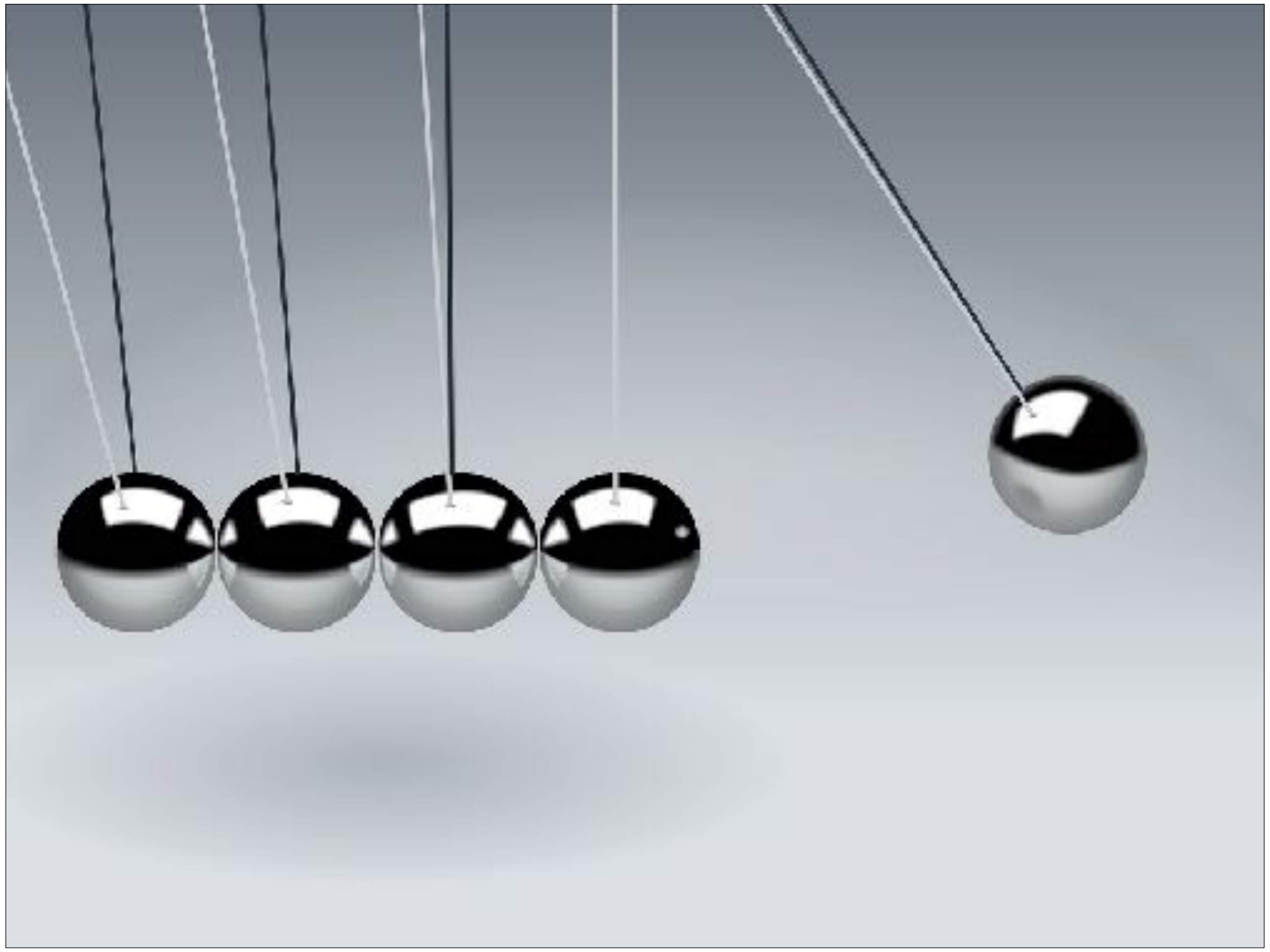
we have a 3 minute long live draw which contains a 5 second countdown at the end of a game. when we need to debug the countdown, instead of waiting for 3 minutes for a 5 second countdown to happen, we could use mock server and make countdown happen every 15 second. it saves so much time.



lastly, mock server is predictable and consistent.

during debugging, if you hooked your app up to a staging server and getting 5s latency, how do you know the fault lies with the network connection or with the server? with mock server, since it runs on localhost, you know that the network couldn’t be an issue.

Now we have talked about the problems and the solutions, what exactly is a mock server?



it is just a web server that will give use consistent response for a particular request. that’s it.

the only requirement is that the request and response format need to be consistent with the production server so that our test wouldn’t be able to tell the difference.

since it is just a tool, it can be as simple or as complex as you need. if you are building mock server for tinder, you might just return a static 10 items to be swiped, or you could implement a randomised 10 items from a stack of 1000 items. it depends on what your test needs.

How do we build a mock server?

if you google `mock server` you would see many different choices to do this.
one of the most popular is `mock-server` and `wiremock` both built with java.
they are good frameworks, but it requires you to write java.

there is a mac only web server called `GCDWebServer` written with Obj-C. It has 2,7k stars on github, so obviously a lot of people are using it.

there are even SAAS companies specializing in this, one of them is mockable

there will be link on my blog post accompanying this talk. don't worry.

so out of all the above, which one would you use? I'd recommend to use whatever stack your api team uses so you can have extra help when you get stuck.

for tonight's talk however, we will skip all the mock frameworks and just go to the basic.

now we get to the fun part, which is of course the live demo. First I’ll give you a summary of what we will do.

we will create a mock server from scratch using node, express, morgan. the whole thing will be very simple but i’ll show how you can scale them up pretty easily. this is the part that i said will take 3 minutes.

after we set up the mock server with hello world, we will run a sample app that i've prepared and its accompanying UI test and rest api endpoints on the web. then i'll show you how to effectively switch endpoints from prod to mock in iOS.

then we will duplicate the api signature on the mock server and run the app test on the mock-server.

lastly, we will see how to easily run our tests using mock-server with CI test runners using fastlane.

- email me augusteo@gmail.com
- connect on linkedin.com/in/victoraugusteo
- blog at www.augusteo.com
- Article about this talk will be up very soon on the blog

ASK MORE QUESTIONS