

## GCSE Bitesize Computer Science

### Facebook's Jackson Gabbard on compilers and interpreters

I'm Jackson Gabbard. I'm an engineer at the Facebook London office. I've been at Facebook about four years and across those four years I've worked on a lot of different teams, which includes, like, Facebook Timeline for mobile devices. I also worked on the App Center, and then I also spent a little bit of time working on Facebook Groups.

Compilers inside of Facebook have a very interesting place because we're a PHP website in our heart, in our history. At the same time, at our scale, we're finding PHP is not enough - it's just not fast enough. So one very brave engineer Haiping Zhao decided that he was going to solve this problem by translating PHP into C++ and then compiling it, because a C++ binary, an executable, will be much, much more efficient than PHP running as an interpreted script. And so over the course of a couple of years, they did this. They went through the process of writing an interpreter for PHP that sort of translates the PHP code into C++ code, which is then compiled and what we found is that we're running about the same number of servers for about four times the amount of traffic, because compiled code, if it's written well, is much, much more efficient than interpreted code.

Any page on PHP that you are going to see in your browser will be PHP, and the PHP layer does a very small amount of work relative to the bigger picture. PHP will assemble the mark-up, will fetch the data, will build the page that gets rendered to you. But those things that you get, the mark-up itself, could be composed of news feed stories that come from a news feed service that is written in a more systems-oriented language. It could be that you're getting a list of search results from Graph Search, which I believe is entirely written in C++. And the notion here is that the thing that we need the ability to change the most, we should write in a very high-level language. And the things that we know are going to be very stable but also very important and that need to be very, very fast – we should isolate those from the parts that change quickly and write them in a highly efficient systems-level language.