

End-to-end

1

1. Memory management is perhaps the most crucial part of the computer system. All programs that are running on the computer system are loaded into memory, where they are then run in the CPU. Memory management also takes care of, that there is not any memory overflows or other risks of errors in memory allocation.
2. Networking function, where system can access outside internet/other network and communicate between other computers. Networking is a way to access other computers and their resources.

2

Streaming protocols used in different applications, ie. Discord and Zoom. Streaming protocols include protocols such as webRTC, FTL and SRT. Users in this case are different softwares that use the protocols interfaces to manage and send streams between applications users. Usually protocol interface, ie. webRTC, is exposed to the application via own package implementation for different stacks.

[Firebase + WebRTC Codelab](#)

[Rust package](#)

[npm package](#)

3

1. Low-level function implementation is about moving some solutions in the stack to the lower level, because it can be implemented there and it could provide performance boost.
2. If functionality can be implemented in the higher layer, then it should be implemented there. This because if you implement some perfect functionality of, ie. file system management, there are still so many risks of things going wrong in some other parts of the program.
3. End-to-end argument means that the computers at the end of communication should perform the integrity check on the even though lower levels have already done so. This means that communication system should be performant when sending data but not really care about reliability of data integrity.

4

1. My phohnes cpu uses ARM architecture, and ARM stands for Advanced RISC Machine.
2. Modern ARM processors can basically do same computations as x86 processors, so the main difference is speed. Bigger difference comes between ARM and TPU cores where TPU (Tensor Processing Unit) specializes to tensor computations.
3. RISC argument is similar to end-to-end argument. Idea is to simplify CPU instructions in favor of simplicity, performance and energy consumption so that smaller and smaller devices can be manufactured.

5

By end-to-end argument my APIs and virtualization platforms as cloud provider should guarantee performance and not necessarily reliability. If I have thousands or millions of user for my APIs or for my virtualization pools, trying to guarantee reliability would be a massive overhead for the system and would probably cause more problems than benefits, ie. system downtimes, massive latency.