



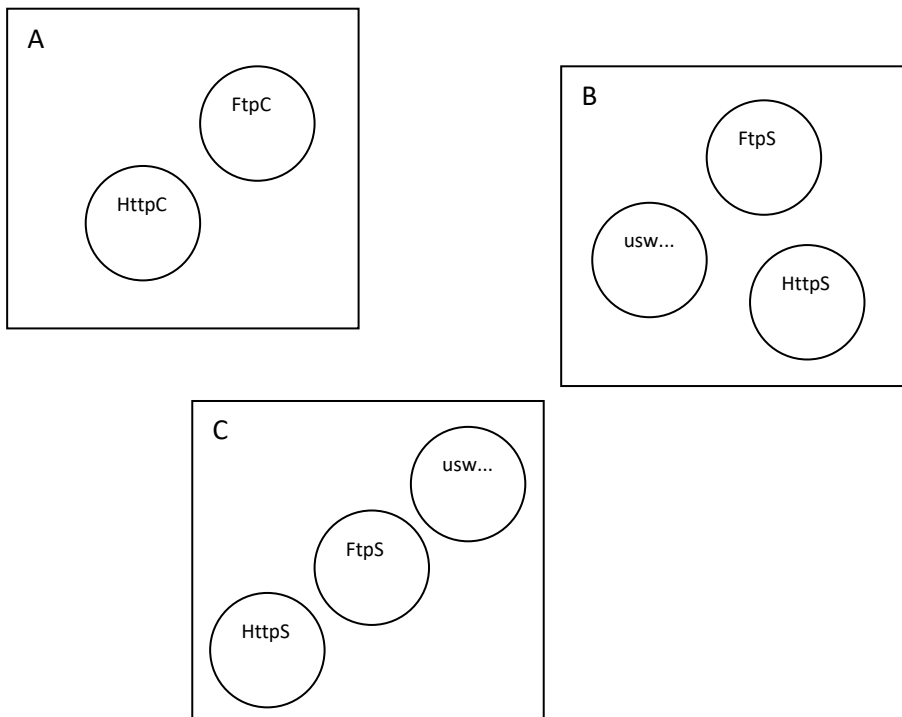
Ports und cgi

cgi sorgt dafür, daß auf einen http-Request hin die Antwort, der http-Reponse, auch wirklich da ankommt, wo die Frage gestellt wurde. Sie wollen ja auch von Amazon nicht das Ergebnis einer Suche sehen, die einer aus Belgien gestellt hat, oder ?

Nehmen wir als Beispiel drei „Häuser“ (= Rechner im Internet). Jedes hat eine Adresse (= IP-Adresse).

In Ihrem Haus (IP-Adresse A) sind zwei Akteure. Einer will immer was von Webservern wissen (er ist ein http-Client, z.b. Firefox, sein Name ist HttpC), der andere will dauernd Programme laden (er ist ein ftp-Client, sein Name ist FtpC).

Die Server dazu leben in anderen Häusern, B und C (usw..). Die Webserver heißen HttpS, die ftp-Server heißen FtpS. Es gibt viele davon.

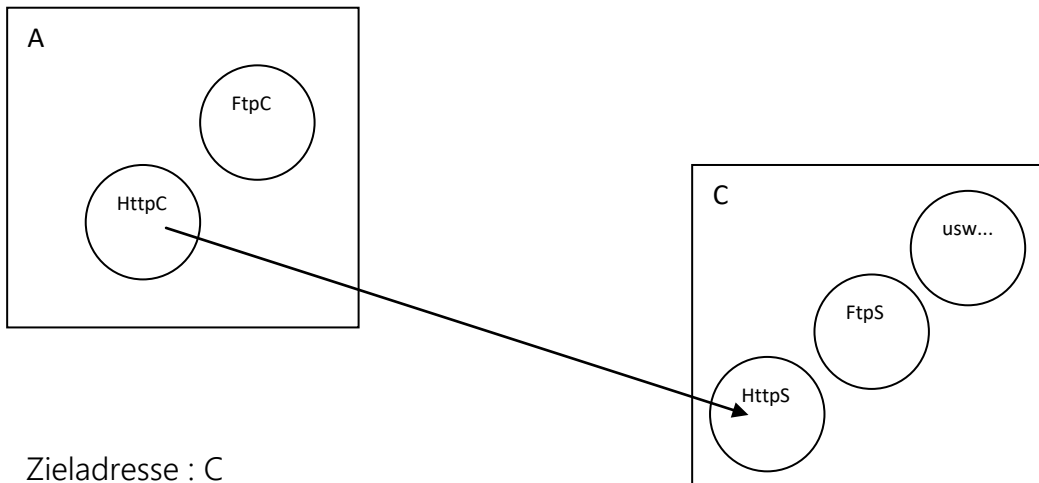


Jetzt möchte ihr HttpC eine Frage an den Webserver HttpS in Haus C stellen. Die Adresse ist einfach : IP-Adresse ist C !
Aber in C ?

Ihr Client weiß aber (steht in seiner Konfiguration), daß der Webserver eine „Zimmernummer“ hat, nämlich 80. Die wurde mal vereinbart („well known ports“, siehe Web). Er schreibt also als Adresse drauf : „C:80“ (Im Original z.b. 62.245.200.166:80)

Und wenn jetzt ne Antwort zurückkommt ?

Damit die beim Zurückkommen an den richtigen Bewohner gehen kann (also den Fragesteller, nämlich den HttpC), wird die Frage beim Abschicken in A mit einer „fiktiven Zimmernummer“ versehen. Die werden erzeugt, indem man die Nummern nach den well known ports (geht bis 1023) einfach hochzählt. Wenn es die erste Frage ist : 1024.



Zieladresse : C

Sendeadresse : A

Zielzimmernummer : 80 (well known port)

Sendezimmernummer : 1024 (von Portier in A vergeben)

Antwort :

Hausnummer = A, Zimmernummer = 1024

Das klappt immer, auch wenn in einem Haus mehrere „gleiche“ Bewohner leben. Man muß die well known ports nicht benutzen, man kann Server auch auf beliebige andere Ports legen, dann muß man halt eben die Clients dafür umkonfigurieren.

Cgi führt nun einfach eine Liste, und merkt sich für jeden ankommenden Request die IP und den Port, z.b. 32.34.33.45:1024. Diese Kombination aus IP und Port nennt man „Socket“ (= Steckdose).

Die von dem Server darauf erzeugt Antwort (z.b. auch die Ausgaben eines aufgerufenen cgi-Skripts in PHP) werden einfach an den zutreffenden Socket in der Liste zurückgeschickt.