

1.2: Current Trends

As important as it is to know how we reached where we are today, it is also important to stay current in web development. New products and innovations can greatly affect the landscape in a short amount of time. We can look to the rapid rise in Facebook, Twitter, and the myriad of Google services now relied upon around the world as examples of how fast new technology is embraced.

Cloud Computing

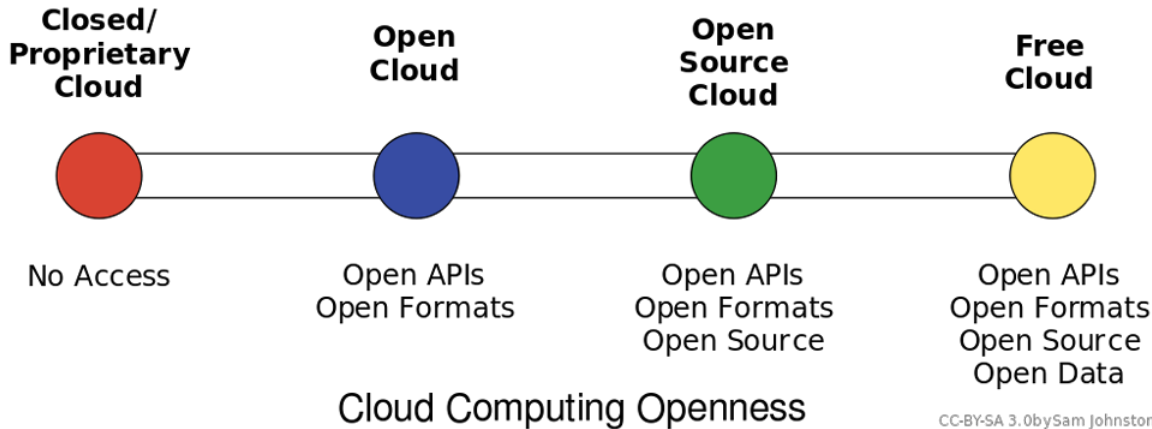


Figure 1.2.1

Cloud Computing Styles

Cloud computing can be loosely defined as the allocation of hardware and/or software under a service model (resources are assigned and consumed as needed). Typically, what we hear today referred to as cloud computing is the concept of business-to-business commerce revolving around “Company A” selling or renting their services to “Company B” over the Internet. A cloud can be public (hosted on a public internet, shared among consumers) or private (cloud concepts of provisioning and storage are applied to servers within a fire wall or internal network that is privately managed), and can also fall into some smaller subsets in between, as depicted in the graphic above.

Under Infrastructure as a Service (IaaS) computing model, which is what is most commonly associated with the term cloud computing, one or more servers with significant amounts of processing power, capacity, and memory, are configured through hardware and/or software methods to act as though they are multiple smaller systems that add up to their capacity. This is referred to as virtualizing, or virtual servers. These systems can be “right sized” where they only consume the resources they need on average, meaning many systems needing little resources can reside on one piece of hardware. When processing demands of one system expand or contract, resources from that server can be added or removed to account for the change. This is an alternative to multiple physical servers, where each would need the ability to serve not only the average but expected peak needs of system resources.

Software as a Service, Platform as a Service, and the ever-expanding list of “as-a-service” models follow the same basic pattern of balancing time and effort. Platforms as a service allow central control of end user profiles, and software as a service allows simplified (and/or automated) updating of programs and configurations. Storage as a service can replace the need to manually process backups and file server maintenance. Effectively, each “as-a-service” strives to provide the end user with an “as-good-if-not-better” alternative to managing a system themselves, all while trying to keep the cost of their services less than a self-managed solution.

Additional Notes

One of the best methods to keep current is by following trade magazines, industry leader blogs, and simply browsing the internet looking for new items or site features you have not noticed before. Content aggregators like Zite, Feedly, and Slashdot are some of my favorites.

As a micro-scale example, imagine you and four friends are all starting small businesses. Faced with the costs of buying servers and software for data storage, web hosting, and office programs, each of you would invest funds into equipment and the staff to maintain it, even though much of it may get little use in the early stages of your company. This high initial investment reduces available funding that may have been used elsewhere, and your return on investment becomes longer. Instead, each of you would

create an account with Amazon's cloud services for file storage and website hosting, which are private to you, but physically stored on servers shared by other users. Since these services are managed offsite by Amazon staff, none of you need to hire IT staff to manage these servers, nor do you have to invest in the equipment itself. Just by not needing to hire a system administrator (estimated at \$40,000 salary) you can pay for just over 3 years of Amazon service (calculated using [Amazon's pricing calculator](#)¹ for basic web services and file storage). When you combine the savings of that employee's fringe costs like health care, along with those of not purchasing your own hardware, this approach can make your initial investment last longer.

These lowered costs are attractive to small businesses and startups for obvious reasons, but are also attractive to large companies with highly fluctuating levels of need. For example, a football team's website sees far more traffic on game days than the off-season. They do not need the ability to serve the same amount of users all the time. Some tangible examples of "as-a-service" tools you may already be using are file hosting services like [Dropbox](#)² or [Google Drive](#).³ Your files are kept on servers along with those from other users that you do not see (unless you share with them intentionally) and you can add or remove extra space to your account whenever you like. Similarly, services like [Amazon Web Services](#)⁴ offer the ability to host your files, applications, and more to both home consumers and commercial clients.

Virtualization

Server virtualization is the act of running multiple operating systems and other software on the same physical hardware at the same time, as we discussed in [Cloud Computing](#). A hardware and/or software element is responsible for managing the physical system resources at a layer in between each of the operating systems and the hardware itself. Doing so allows the consolidation of physical equipment into fewer devices, and is most beneficial when the servers sharing the hardware are unlikely to demand resources at the same time, or when the hardware is powerful enough to serve all of the installations simultaneously.

The act of virtualizing is not just for use in cloud environments, but can be used to decrease the "server sprawl," or overabundance of physical servers, that can occur when physical hardware is installed on a one-to-one (or few-to-one) scale to applications and sites being served. Special hardware and/or software is used to create a new layer in between the physical resources of your computer and the operating system(s) running on it. This layer manages what each system sees as being the hardware available to it, and manages allocation of resources and the settings for all virtualized systems. Hardware virtualization, or the stand alone approach, sets limits for each operating system and allows them to operate independent of one another. Since hardware virtualization does not require a separate operating system to manage the virtualized system(s), it has the potential to operate faster and consume fewer resources than software virtualization. Software virtualization, or the host-guest approach, requires the virtualizing software to run on an operating system already in use, allowing simpler management to occur from the initial operating system and virtualizing program, but can be more demanding on system resources even when the primary operating system is not being used.

Ultimately, you can think of virtualization like juggling. In this analogy, your hands are the servers, and the balls you juggle are your operating systems. The traditional approach of hosting one application on one server is like holding one ball in each hand. If your hands are both "busy" holding a ball, you cannot interact with anything else without putting a ball down. If you juggle them, however, you can "hold" three or more balls at the same time. Each time your hand touches a ball is akin to a virtualized system needing resources, and having those resources allocated by the virtualization layer (the juggler) assigning resources (a hand), and then reallocating for the next system that needs them.

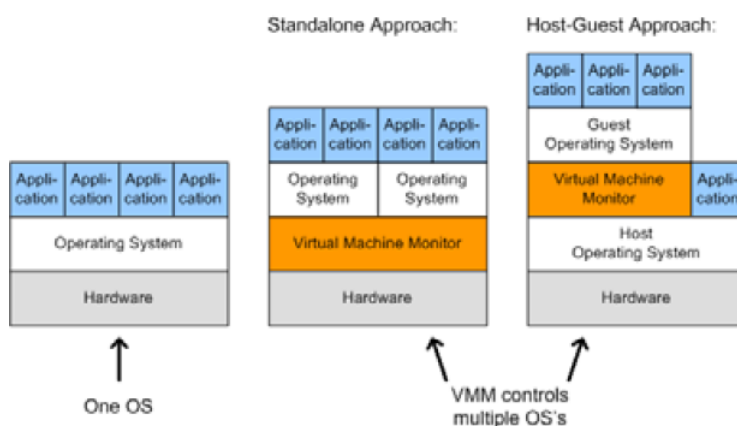


Figure 1.2.2 Virtualization Styles. (By [Daniel Hirschbach](#)

[\[CC-BY-SA-2.0 Germany\]](#) via [Wikimedia](#))

The addition of a virtual machine as shown above allows the hardware or software to see the virtual machine as part of the regular system. The monitor itself divides the resources allocated to it into subsets that act as their own computers.

Net Neutrality

This topic is commonly misconstrued as a desire for all Internet content to be free of cost and without restrictions based on its nature. In fact, net neutrality is better defined as efforts to ensure that all content (regardless of form or topic) and the means to access it, are protected as equal. This means Internet Service Providers (ISPs) like your cable or telephone company cannot determine priority of one site over another, resulting in a “premium” Internet experience for those able to pay extra. Additional concerns are that without a universal agreement, a government may elect to restrict access to materials by its citizens (see [North Korea censorship](#)5), and similarly that corporations controlling the physical connections would be able to extort higher prices for privileged access or pay providers to deny equal access to their competitors.

Useful Features

Legislation continues to change regarding what is and is not legal or acceptable content on the internet. Laws change over time as well as across jurisdictions and can greatly differ. Just because material is legal in your area does not mean it is in others and you may still be in violation of laws applicable in the location of your server.

Existing laws vary around the world, some protecting the providers, some protecting the user. The United States received considerable attention in 2012 for anti-piracy bills that were highly protested both with physical rallies and online petitions. Each bill drew debate over what affects the stipulations would have not only within the United States, but over the Internet as a whole. Even though [SOPA](#)6 (introduced by the House) and [PIPA](#)7 (introduced by the Senate after the failure of [COICA](#)8 in 2010) were not ultimately ratified, The United States and other countries had at that point already signed [ACTA](#)9 in 2011, which contained provisions that placed the burden on ISPs to police their users regardless of sovereign laws in the user’s location.

Learn more

Keywords, search terms: Cloud computing, virtualization, virtual machines (VMs), software virtualization, hardware virtualization

Xen and the Art of Virtualization: <http://li8-68.members.linode.com/~caker/xen/2003-xensosp.pdf>

Virtualization News and Community: <http://www.virtualization.net>

Cloud Computing Risk Assessment: <http://www.enisa.europa.eu/activities/risk-management/files/deliverables/cloud-computing-risk-assessment>

Without formal legislation, judges and juries are placed in positions where they establish precedence by ruling on these issues, while having little guidance from existing law. As recently as March 2012 a file sharing case from 2007 reached the Supreme Court, where the defendant was challenging the constitutionality of a \$222,000 USD fine for illegally sharing 24 songs on file sharing service Kazaa. This was the first case for such a lawsuit heard by a jury in the United States. Similar trials have varied in penalties up to \$1.92 million US dollars, highlighting a lack of understanding of how to monetize damages. The Supreme Court denied hearing the Kazaa case, which means the existing verdict will stand for now. Many judges are now dismissing similar cases that are being brought by groups like the Recording Industry Association of America ([RIAA](#)10), as these actions are more often being seen as the prosecution using the courts as a means to generate revenue and not recover significant, demonstrable damages.

As these cases continue to move through courts and legislation continues to develop at the federal level, those decisions will have an impact on what actions are considered within the constructs of the law, and may have an effect on the contents or location of your site.

Cyber Warfare

Intentional, unauthorized intrusion of systems has existed about as long as computers have. While organized, coordinated attacks are not new, carrying them out in response to geopolitical issues is now emerging, as was found in the brief 2008 war between Russia and Georgia. Whether the attacks on each country’s infrastructures were government sanctioned or not is contested, but largely irrelevant. What is relevant is that these attacks will only continue, and likely worsen, in future disputes.

In the United States and other countries, equipment that controls aging infrastructure for utilities is increasingly connected, with control computers at facilities for electric, water, gas, and more being placed online to better facilitate monitoring and maintenance.

However, many of these systems were not developed with this level of connectivity in mind, therefore security weaknesses inherent in the older equipment can result in exploits that allow **Hackers** to cause real, permanent damage to physical equipment, potentially disrupting the utilities we rely on every day.

Tehran's uranium enrichment development facilities were targeted in late 2010 by a custom-created virus that focused on equipment used in the refining of nuclear material. The virus would randomly raise or lower the speed of the equipment in a manner that would not create alarms, but enough to strain the equipment. This would lead to equipment failures, after which the replacement hardware would be similarly infected. Eventually discovered, the virus had been running for many months, delaying the project and increasing its costs. This virus was intentionally designed to run in that particular environment and was based on the specific **SCADA** hardware involved, and in this case was such a sophisticated attack that it is widely believed to have been facilitated by the United States and Israel.

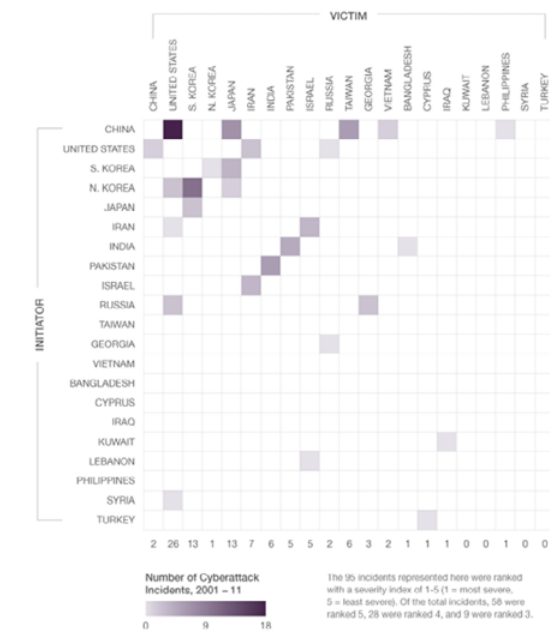


Figure 1.2.3 10 Years of Known Cyber Attacks (foreignaffairs.com)

The graph above, from foreignaffairs.com, provides an idea of how prevalent government to government attacks are becoming. We should keep in mind that the ninety-five incidents depicted are only the known, reported incidents, and the true number is likely higher.

Botnets

Botnets are not exactly a new threat to the Internet, but they remain one of the most persistent threats to the average user and their computer. The word botnet, an amalgamation of the words robot and network, is an accurate description of what it entails. Botnets are programs that use a network connection to communicate with each other to coordinate and perform tasks. Originally, botnets were created as part of programs for Internet Relay Chat (**IRC**) to help establish and control channels people would log into to talk to each other. They are still in frequent use today for a number of legitimate, non-malicious tasks.

We have also seen a rise in malicious botnets, designed to work undetected in the background of your computer. The controller (typically referred to as the command and control server) uses the infected machines to complete tasks that require large amounts of processing power and/or bandwidth to complete, like finding or exploiting weaknesses in networks or websites, or to “mine” infected systems for personal data such as credentials, credit card numbers, and other information that can then be used or sold to others.

Additional Notes

It did not take long for the first virus to enter the internet. Just two years after the first systems were connected, The Creeper (a self-replicating script) was created in 1971, which did no harm but displayed “I’m the creeper, catch me if you can” on infected

machines. It was immediately followed by The Reaper, the first anti-virus program, which too self-replicated, removing the Creeper wherever it was found.

Some botnet controllers have grown so large and organized that they act as businesses in competition, typically “renting” their botnet out as a service or tool to others for agreed upon rates. Efforts by security researchers to detect and analyze botnets often involve close coordination with government agencies and law enforcement as the size of an average botnet typically involves computers from multiple countries. Simply shutting down or attempting to remove the malicious files from infected systems could cause unintended damage to the machines, further complicating the process of eliminating a botnet.



Figure 1.2.1: Botnets. (By Tom-B [CC-BY-

SA-3.0], via Wikimedia)

[Learn more](#)

Keywords, search terms: Botnets, command and control system, malware, network security

Build Your Own Botnet: http://howto.wired.com/wiki/Build_your_own_botnet_with_open_source_software

Honeynet Project: <http://www.honeynet.org/papers/bots/>

Internet of Things

In much the same vein of the connection of older equipment to the networks of the modern world, the newest devices emerging into the market can also be a bit more non-traditional. This results in an internet that is soon to be awash with live connections from everything from cars to ovens and refrigerators, an explosion of devices no longer focused on delivering information to the masses as much as aggregating many data sources of interest to a small set of recipients. Some cars now include the ability for consumer service companies to perform tasks like remotely shutting down your car if stolen; coordinating use of these tools with law enforcement allows them to stop a vehicle before or during a pursuit. While these are innovative tools with positive uses, they also add new vectors for a malicious person to attack. Instead of the thief being thwarted, he might use a device to shut your car down at an intersection, eliminating your ability to simply drive away when he approaches. The very tool intended to stop him afforded him a means to gain access to your vehicle. This is not merely waxing philosophically, either. It has been demonstrated as a [proof of concept](#)¹¹

backed by researchers funded by DARPA.

As more devices are introduced to the Internet, the amount of interaction with things as simple as small appliances is increasing. Comments like “We have to stop by the store on the way home, the toaster report said we will need at least one loaf of bread for the week” seem silly to us now, but could eventually exist in the same breath as “The fridge called, it ordered our groceries for the week.” For about \$2,700 USD, Samsung already offers a fridge with interactive features similar to these ideas.

Items embedded with RFID tags contribute to the Internet of Things, as they can be tracked and provide information that can be aggregated and applied to processes. Shipping crates with RFID expedite taking inventory as their tags can be scanned

automatically in transit. Access cards not only allow privileged access to restricted areas but also let us know where people (or, at least their cards) are located and where they have been. Home automation systems allow lights, locks, cameras, and alarms to be managed by your smart phone to the extent that your lights can come on, doors unlocked, and garage door opened, when it detects that your phone has entered the driveway. All of these are items—not people—interacting with the Internet to fulfill a task, and are part of the emerging Internet of Things.

Proliferation of Devices

As reliance on the Internet and the drive for constant connection proliferate through our societies, and technology becomes more affordable and adaptable, we have not only left the age of one computer per home, but meandered even past the point where everyone in the house has their own device, and now the average consumer has multiple devices. The proliferation allows us to adjust technology to fit where we need it in our lives. I use my desktop for hardware intensive applications at home, or for doing research and web development where multiple monitors eases my need to view several sources at once. Out of the house, my tablet allows me to consume information and is easily slid into a keyboard attachment that allows it to operate as a laptop, turning it into a content creation device by reducing the difficulty of interacting by adding back a keyboard and mouse.

Improvements in software both in efficiency and ease of use allow older hardware to get second lives. My laptop, though ten years old, is still running happily (albeit without a useful battery life) and is still capable of browsing the internet and being used as a word processor due to a lightweight Linux operating system that leaves enough of its aging resources available for me to complete these tasks. When the average lifespan of a laptop is typically considered to be only three years, many older devices like mine have not left operation, and are still finding regular use in our growing set of tech tools.

1.2: Current Trends is shared under a [not declared](#) license and was authored, remixed, and/or curated by LibreTexts.

- 1.2: Current Trends by [Michael Mendez](#) has no license indicated.