Introduction to the Cloud

As the popular saying goes:

> The cloud is just somebody else's computer.

There is no magic, just using the cloud does not magically solve problems we are having with a traditional infrastructure. This lecture will teach you how a cloud is built and what the typical services are that they offer as well as the pitfalls which may come with such setups.

What is a Server?

In a hurry?

Servers:

- Nowadays either x86 or ARM architecture.
- Redundant, hot-swap hardware (power supply, fans, etc).
- Flat build profile for rack mounts.
- Out-Of-Bounds management interface for monitoring and remote management.
- Power consumption is a concern.
- Buying servers can take a long time and present an up-front investment.

In older times, servers were completely different from the machines we used for regular work. Not just in weight and form factor, but in architecture. The landscape would stretch from SPARC and Power architectures to the x86 architecture we use in our PCs.

Over time, however, the x86 architecture took over to the point where a server today, from an architectural standpoint, is exactly the same as the machine you are using right now. This means that you can copy the machine you are using onto a server and chances are it will run without any modification.

The only notable exception is the rise of the ARM architecture which is popular in phones and tables and is known for its low power consumption. In recent years there has been a small but noticeable trend to run ARM in a datacenter.

At home it may not matter if your computer uses 200 or 300 watts of power. In a datacenter, at the scale of hundreds, thousands or tens of thousands of servers, saving even a few percent of power will translate to huge cost savings.

The difference in build is, however, quite apparent. While some servers, mainly built for office use, have the standard “tower” build, most servers have a flat profile designed to be mounted in racks as displayed on the picture. Since most servers are not high enough to take full size expansion cards (graphics cards, network cards, etc), servers may contain a separate removable component for these usually called a riser.

A server in a rack, pulled out and the top taken off. The internal components, such as dual, hot-plug power supply units, hot-plug fans, CPU, the riser and hot-swap disks are visible. *An HP Proliant G5 server pulled out of its rack. Source: Wikipedia*

Server racks are standardized closets that have mounting screws for rails that allow pulling servers out even mid operation and replace components while the server is running.

Servers also come with a high level of redundancy. While you may have a single power supply in your home computer, servers typically have two that are able to take power from different power inputs. This makes sure that the server keeps running even if one of the two power supplies fail, or if one of the two power inputs goes down.

Also in contrast to your home setup these servers contain an Out-Of-Bounds Management interface that allows remote management of servers even when they are turned off. The hardware components built into the server report their health status to this OOB management interface which allows for the simultaneous monitoring of thousands of machines.

Note

This is a fairly generic description of servers. Different vendors may chose to leave out certain features of their more “budget” line of servers, or call certain components differently. HP, for example, calls their OOBM “Integrated Lights Out”, Dell “DRAC - Dell Remote Access Control”, etc.

When it comes to purchasing servers larger companies tend to go with the same components for a longer period of time and they also buy support from the vendor. This sometimes includes hardware replacement done entirely by the vendor in the datacenter without the need for the customer to have staff on site. However, purchasing a specific set of components or ordering larger quantities of servers presents a logistics challenge and can sometimes take up to 3-4 months. Buying hardware is also an up-front investment which is hard to justify when demands change rapidly.

What components are redundant in a server?

Power supply

CPU

RAM

Fan

OOBM

What components are redundant in a server?

Remotely manage a server

Receive hardware malfunction alerts

The Anatomy of a Datacenter

In a hurry?

Datacenter components:

- Racks to house servers.
- Larger customers have cages for their racks.
- Cabling under the floor.
- Redundant cooling, fire suppression systems and power supply.
- Some datacenters provide internet connectivity.
- Eco friendliness is becoming a factor.

Since the cloud is just somebody else's computer, that computer needs to be hosted somewhere. Servers are almost exclusively hosted in datacenters. Let's take a look at what is involved in running a datacenter.

First of all, as mentioned above, most servers are going to be rack-mounted so you need a bunch of racks. These racks are installed in rows, often with a fake floor to allow for cabling to go under the floor.

A data center containing a bunch of racks for servers *A datacenter with racks. Source: Wikipedia*

Since servers produce a lot of heat, a datacenter also requires cooling. There are a variety of ways to solve cooling, some are more “green” than others. Some datacenters, for example, opt to install a “cold aisles” where the cold air is pumped between two rack rows and is pushed through the racks to cool the servers.

Apart from cooling, datacenters also require automated fire suppression systems simply because of the amount of electricity going through. Datacenters usually go with a non-destructive fire suppression system such as lowering the oxygen content of the air enough to stop the fire.

All critical systems in a datacenter (power, cooling, fire suppression systems) are usually built in a redundant fashion because the loss of either of those systems will potentially mean a complete shutdown for the datacenter. Datacenter operators usually have further contingency plans in place, too, such as a UPS (battery) system, diesel generator, fuel truck on standby, hotline to the fire department, etc. to make sure the datacenter can keep its required uptime.

On the networking side of things, matters get slightly more complicated. Some datacenter providers also offer you the ability to use their network uplink (also redundant), but larger customers will prefer to host their own networking equipment and negotiate their own internet uplink contracts. Since there is no generic rule for how datacenters handle this, we will dispense with a description.

It is also worth noting that larger customers (banks, cloud providers, etc) usually prefer to have their own racks in a separated gated area called a “cage” to which they control access.

The Anatomy of the Internet

> **In a hurry?**
>
> **Internet:**
>
> - IP ranges are advertised using BGP.
> - Providers connect direcly or using internet exchanges.
> - 16 global providers form the backbone of the internet (tier 1).

Once the physical infrastructure is set up there is also the question of how to connect to the Internet. As mentioned before, networks can be very complicated and there is no one size fits all solution. Smaller customers will typically use the network infrastructure provided by the datacenter while larger customers will host their own network equipment.

A diagram showing a typical datacenter networking setup.

Again, generally speaking racks will be equipped with a Top-of-Rack switch to provide layer 2 (Ethernet) connectivity between servers. Several ToR may have interconnects between each

other and are usually connected to one or more routers. Routers provide layer 3 (IP) routing to other customers in the same datacenter, internet exchange, or may be connected via dedicated fiber to another provider.

> **Note**
>
> If you are not familiar with computer networks we recommend giving the Geek University CCNA course a quick read. While you will not need everything, you **will** have to understand how IP addresses, netmasks, etc work in order to pass this course.

Providers on the internet exchange data about which network they are hosting using the Border Gateway Protocol. Each provider's router announces the IP address ranges they are hosting to their peer providers, who in turn forward these announcements in an aggregated form to other providers.

Providers have agreements with each other, or with an Internet Exchange, about exchanging a certain amount of traffic. These agreements may be paid if the traffic is very asymmetric or one provider is larger than the other. Alternatively providers can come to an arrangement to exchange traffic for free. Internet exchanges facilitate the exchange between many providers for a modest fee allowing cost-effective exchange of data. Depending on the exchange the rules are different. Local exchanges, for example, may only allow advertising local (in-country) addresses, while others are built for a specific purpose.

Generally speaking providers can be classified into 3 categories. Tier 1 providers are the global players that are present on every continent. They form the backbone of the Internet. At the time of writing there are 16 such networks. Tier 2 are the providers who are directly connected to the tier 1 providers, while tier 3 is everyone else.

An illustration of how internet providers are connected. In this example Global A and Global B are two providers which are connected in the same level, Local A and Local B are connected to Global A, while Local C is connected to Global B. Local A and Local B also exchange data directly and Local B and C exchange data over a local internet exchange. The example datacenter customer is connected to Local B.

Software Stack

> **In a hurry?**
>
> **Software stack:**
>
> - Virtualization.
> - Operating system.
> - Application runtime.

- Application.

The purpose of all this is, of course, to run an application. Each server hosts an operating system which is responsible for managing the hardware. Operating systems provide a simplified API for applications to do hardware-related operations such as dealing with files or talking to the network. This part of the operating system is called the kernel. Other parts form the userland. The userland includes user applications such as a logging software. Specifically on Linux and Unix systems the userland also contains a package manager used to install other software.

Modern x86 server CPUs (and some desktop CPUs) also have a number of features that help with virtualization. Virtualization lets the server administrator run multiple guest operating systems efficiently and share the server resources between them.

Did you know?

You can find out if an Intel CPU supports hardware virtualization by looking for the `VT-x` feature on the Intel ARK. Unfortunately AMD does not have an easy to use list but you can look for the `AMD-V` feature on AMD CPUs.

Note

Virtualization is different from containerization (which we will talk about later) in that with virtualization each guest operating system has its own kernel whereas containers share a kernel between them.

There is one more important aspect of finally getting an application to run: the runtime environment. Except for a few rare occasions applications need a runtime environment. If the application is compiled to machine code they still need so-called shared libraries. Shared libraries are common across multiple applications and can be installed and updated independently from the application itself. This makes for a more efficient update process, but also means that the right set of libraries need to be installed for applications.

If the applications are written higher level languages like Java, Javascript, PHP, etc. they need the appropriate runtime environment for that language.

One notable exception to the runtime environment requirement is the programming language Go. Go compiles everything normally located in libraries into a single binary along with the application. This makes it exceptionally simple to deploy Go applications into containers.

The Cloud

In a hurry?

Typical cloud features:

- API
- Dynamic scaling
- Can be classified into IaaS, PaaS and SaaS
- IaaS service offerings typically include:
    - Virtualization.
    - Network infrastructure.
    - Everything required to run the above.
- PaaS service offerings typically include a managed service ready to be consumed by a developer.
- SaaS service offerings typically include a managed service ready to be consumed by a non-technical end user.

All of the previously discussed things were available before the “cloud”. You could pay a provider to give you access to a virtual machine where you could run your applications. What changed with the cloud, however, is the fact that you no longer had to write a support ticket for changed and everything became self service.

The cloud age started with an infamous e-mail from Jeff Bezos to his engineers in 2002 forcing them to use APIs to exchange data between teams. The exact e-mail is no longer available but it went along these lines:

> 1) All teams will henceforth expose their data and functionality through service interfaces.
>
> 2) Teams must communicate with each other through these interfaces.
>
> 3) There will be no other form of interprocess communication allowed: no direct linking, no direct reads of another team’s data store, no shared-memory model, no back-doors whatsoever. The only communication allowed is via service interface calls over the network.
>
> 4) It doesn’t matter what technology is used. HTTP, Corba, Pubsub, custom protocols — doesn’t matter.
>
> 5) All service interfaces, without exception, must be designed from the ground up to be externalizable. That is to say, the team must plan and design to be able to expose the interface to developers in the outside world. No exceptions.
>
> 6) Anyone who doesn’t do this will be fired.

This marked the beginning of Amazon Web Services the first and also the most successful public cloud offering. The first public release of AWS was in 2004 with SQS their message queue service, and got completely overhauled in 2006 where the Elastic Compute (EC2) and the Simple Storage Service (S3) service made its first public appearance.

The APIs provided by cloud providers allow for a large amount of flexibility. If new servers are needed they can be launched within a few minutes. If there are too many servers they can be deleted. The same goes for other services: with the API (and the appropriate billing model) comes flexibility to adapt to change.

The other factor that makes it easy to adapt to change is of course the fact that these services are managed. The cloud customer doesn't need to hire a team of engineers to build a database service, for example, it can be consumed without knowing how exactly the database is set up.

Generally speaking, cloud services can be classified into three categories: Infrastructure as a Service (IaaS) providing virtual machines and network infrastructure, Platform as a Service (PaaS) offering services for developers to use, and Software as a Service (SaaS) offering end-user services.

A comparison of different cloud models. With self-managed the customer needs to take care of the application, runtime, operating, system, virtualization, hardware, network, power, cooling, fire suppression, and housing. With IaaS onl the application, runtime, and the operating system. With PaaS the customer only takes care of the application. With SaaS the customer takes care of nothing and only consumes the service.

> **Note**
>
> SaaS will not be discussed in this course.

Infrastructure as a Service (IaaS)

The most basic of cloud service offerings is IaaS. IaaS means that the cloud provider will manage the infrastructure used by the customer. Infrastructure in this sense means the ability to manage (provision, start,stop) virtual machines. In very rare cases some providers also offer “bare metal” machines in this fashion. However, most bare metal providers do not offer a true IaaS as machines cannot be ordered using an API, have an up-front fee and are billed on a monthly basis.

IaaS also includes the network connectivity to the Internet. Almost all IaaS providers also offer a built-in firewall, virtual private networks (VPC) that can be used to connect virtual machines together without sending the traffic over the Internet, and other network services.

Note that network services can differ greatly. For example, some providers implement private networks on a regional basis while other providers offer private networks that can be used to connect virtual machines that are located in different regions.

Managed Services (PaaS)

In a hurry?

Managed services:

- The provider is typically responsible for provisioning, backups, monitoring and restoring the service if needed.
- Low entry cost.
- Little in-house know-how required.
- Vendor lock-in for non-standard services.
- If problems arise they can be hard to debug.

Apart from offering virtual machines and network infrastructure as a service many cloud providers also offer additional services that are typically used by developers such as managed databases. These services are managed in the sense that the developer does not need to run the operating system and the database software itself. However, the developer has to configure the database such that it works according to the needs of the application. The provider takes over duties like installing the service, making backups, monitoring, and restoring the service in case of an outage.

Did you know?

There are purely PaaS providers that do not offer IaaS, only a developer-friendly platform to run software on. Your run of the mill average PHP web hosting provider does qualify if you can order the service using an API. One notable example that rose to prominence is Heroku.

The pricing of these managed services varies greatly but they usually follow the cloud model. Most cloud providers offer at least a low cost or even free entry version that has a small markup on top of the IaaS costs.

The massive benefit of using these managed services is, of course, that you do not need to have in-house knowledge about how to operate them. You don't need an in-house database operator, you *“just”* need to know how to set up the database in the cloud. This lets your company focus on the core business they are trying to build and outsource the know-how required to build these services on top of the IaaS layer.

As you might imagine managed servies also have downsides. Most importantly they present a clear vendor lock-in. This means that if you want to move to a different cloud provider you will have a hard time doing so if you are using one of the more specialized services. In other words the level of standardization across providers matters.

It is also worth mentioning that managed services tend to work well for a large number of customers but a few number of customers can run into hard to debug problems. This debugging

difficulty arises out of the inherent opacity of the services: you, the customer, don't see what's happening on the IaaS layer. If a database, for example, fills the available bandwidth may not be notified and are left guessing why your database misbehaves.

Business Models

> **In a hurry?**
>
> **Billing models:**
>
> - IaaS is typically priced per-second based on the instance size.
> - PaaS can be priced similar to IaaS but also per-request or data volume.
> - Data volume is typically priced based on usage. Some providers charge for data volume even for internal traffic.

The flexibility of cloud providers comes from their usage-based pricing. This can vary depending on the type of service used. For example, virtual machines (IaaS) is typically priced on a per-second basis. When you start a virtual machine for only 5 minutes you will only pay for 5 minutes of runtime. This lets you optimize the usage costs of the cloud *if* you use **automation** to start and stop machines or services on demand.

Typically the cost savings are realized in these scenarios:

On-off usage

A graph showing zero usage with two spikes when the service is used.

This usage type is typical for batch processing. Machines are only started when there is a workload (e.g. video conversion, machine learning training job, etc.)

Growth usage

A graph showing exponential growth.

Projects that see a growth curve often opt to use a cloud provider as well since buying hardware in larger quantities typically takes 2-4 months.

Sudden spike usage

A graph showing a sudden usage spike.

Cloud providers can also be useful if a service encounters a sudden load spike. This is typically the case for webshops around Black Friday and other sales events.

Periodic spike usage

A graph showing periodic spikes.

Almost every service has usage spikes depending on the time of day. This can be used to scale the service up and down.

Per-request billing

IaaS services are typically priced based on an allocated amount of resources determined at the start of the virtual machines. Some PaaS services also use this billing model. (Typically Databases as a Service.) Other PaaS services often opt for a per-request or a data volume based billing approach.

Cost planning with the cloud

Cost planning is an important part of the job of a cloud architect. Depending on the billing model the cloud provider adopts this can be fairly simple to almost impossible. As a rule of thumb the billing model of larger cloud providers (AWS, Azure, Google Cloud) is more complex than smaller cloud providers (DigitalOcean, Exoscale, Upcloud, etc).

It is worth noting that data transfer is typically charged based on volume. Some cloud providers even charge for internal traffic between availability zones.

Cost-comparison with on premises systems

One of the most important cases for cost analysis will be the comparison with on premises architecture. It is quite common to create a cost comparison where the investment required for a hardware purchase is compared with the cloud cost over 3 or 5 years.

These comparisons can be quite misleading because they often don't contain any cost attribution for the required work and they also do not use the advantages of the cloud in terms of scaling. Without these factors a hardware purchase will almost always be cheaper when calculated over 5 years and be roughly equal when compared over 3 years.

Private vs. Public Cloud

In a hurry?

Private cloud:

- Hosted on-premises or in the public cloud using only private connections.
- Large cloud providers offer their services “in a box” to host yourself.

Most cloud applications will reside on a public cloud and be accessible from the Internet. However, for some usecases access over the public Internet is not desirable or even strictly forbidden by laws or regulations. In order cases companies may decide that it is their policy that certain systems must never be connected to the public Internet, or only connect via a self-hosted gateway.

In these cases a private cloud is desirable to minimize the risk of cross-contamination or data exposure due to other tenants being on the same infrastructure. Such cross-contamination is not uncommon, for example in recent years there have been a litany of CPU bugs such as Meltdown and Spectre.

Public cloud providers have extended their offers to address these concerns. Their service offerings now include dedicated hypervisors to mitigate CPU bugs, the ability to connect the cloud bypassing the Internet, or accessing the cloud API's from a private network. Some cloud providers even went as far offering a self-hosted setup where the public cloud offering can be brought on-premises.

These features, especially the connectivity-related features, also allow for the creation of a hybrid cloud where a traditional infrastructure can be connected to the cloud. This gives a company the ability to leverage the flexibility of the cloud but also keep static / legacy systems for financial or engineering reasons.

Automation

> **In a hurry?**
>
> **Automation:**
>
> - Documents how a cloud is set up.
> - Gives a reproducible environment.
> - Allows for spinning up multiple copies of the same environment.
> - Not all tools are equal. Some tools allow for manual changes after running them (e.g. Ansible), others don't (e.g. Terraform).

As you can see from the previous sections cloud computing doesn't necessarily make it simpler to deploy an application. In this regard setting up a server and not documenting it is the same as setting up a cloud and not documenting it. In the end in both cases modifications will be hard to carry out later since details of the implementation are lost.

It will also be hard to create a near-identical copy of the environment for development, testing, etc. purposes. This is partially due to the lack of documentation, and partially because of the

manual work involved. In the past this problem was addressed by *cloning* virtual machines which was an approach with limited success as customizations were hard to make.

The rise of API's brought a much welcome change: automation tools such as Puppet, Ansible and Terraform not only *automate* the installation of a certain software but also *document* how the setup works.

When engineers first go into automation they tend to reserve a part of the work to be done manually. For example they might use Ansible to create virtual machines and install some basic tools and install the rest of the software stack by hand. This approach is workable where the software stack cannot be automated but should be avoided for software where this is not the case as the lack of automation usually also means a lack of documentation as described above.

Automation tools are also not equally suited for each task. Terraform, for example, expects full control of the infrastructure that is created by it. Manual installation steps afterwards are not supported. This approach gives Terraform the advantage that it can also *remove* the infrastructure it creates. Removing the infrastructure is paramount when there are multiple temporary environments deployed such as a dev or testing environment.

Ansible, on the other hand, allows for manual changes but does not automatically implement a tear-down procedure for the environment that has been created. This makes Ansible environments, and Ansible code harder to test and maintain but more suited for environments where traditional IT is still a concern.

Regulation

In a hurry?

GDPR:

- Applies to all companies, world-wide, that handle personal data of EU citizens.
- Companies must keep track of why and how data is handled.
- Data subjects have wide ranging, but not unlimited rights to request information, correction and deletion of data.
- Data breaches have to be reported and may carry a fine.
- The right of deletion means that backups have to be set up in such a way that data can be deleted.

CLOUD Act:

- US authorities can access data stored by US companies in Europe.

DMCA:

- Copyright infringements can be solved by sending a DMCA takedown notice to the provider.
- Providers have to restore the content if the uploader sends a counter-notification.

> **CDA Section 230:**
>
> • Shields providers from liability if their systems host illegal content which they don't know about.
>
> **Privacy Shield:**
>
> • Establishes a legal basis for transfering data from Europe to the US.

General Data Protection Regulation (G.D.P.R., EU)

The GDPR (or DSGVO in German-speaking countries) is the general overhaul of privacy protections in the EU. The GDPR replaces much of the previously country-specific privacy protections present in the EU.

JURISDICTION

The GDPR applies to all companies that deal with the data of EU citizens around the globe. This is made possible by trade agreements previously already in place. All companies that handle the data of EU citizens even if the companies are not located in the EU.

STRUCTURE

- **Data subject:** the person whose data is being handled.
- **Personal identifiable data:** (PI) data that makes it possible to uniquely identify a data subject. PI can also be created when previously non-PI data is combined to build a profile.
- **Data controller:** The company that has a relationship with the data subject and is given the data for a certain task.
- **Data processor:** A company that processes data on behalf of the data controller. The data processor must have a **data processing agreement** (DPA) with the data controller. IaaS and PaaS cloud providers are data processors.

PURPOSES OF DATA PROCESSING

One new limitation the GDPR brings to privacy regulation is the fact that there are fundamental limits to data processing. You, the data controller, cannot simply collect data for one purpose and then use that same data for another purpose. The purposes to which data is collected have to be clearly defined and in several cases the data subject has to provide their explicit consent. (In other words it is not enough to add a text to the signup form that says that they agree to everything all at once.)

Legal grounds to data processing are detailed in Article 6 of the GDPR. These are:

1. If the data subject has **given their explicit consent** to process their data for a specific purpose.

2. If the processing is necessary to **perform a contract** with the data subject. In other words you do not need additional consent if you are processing data in order to fulfill a service to the data subject.
3. To perform a **legal obligation**. For example, you have to keep an archive of your invoices to show in case of a tax audit.
4. To protect the **vial interests of the data subject or another natural person**.
5. To perform a task which is in the **public interest**, or if an official authority has been vested in the data controller.
6. For the purposes of a **legitimate interest** of the data controller (with exceptions). This might seem like a catch-all but courts have defined legitimate interests very narrowly. The data controller must show that the legitimate interest exists and cannot be achieved in any other way than with the data processing in question. One good example for the legitimate interests clause would be a webshop that is collecting data about their visitors in order to spot fraudulent orders.

DATA SUBJECTS' RIGHTS

Chapter 3 of the GDPR deals with the rights of the data subject. These are:

1. **The right to be informed.** This right lets the data subject request several pieces of information:
    - The purposes of processing.
    - The categories of personal data concerned.
    - Who received their data, in particular recipients in third countries or international organisations.
    - How long the data will be stored.
    - The right to request modification, deletion or restricting the processing of the personal data in question.
    - The right to log a complaint.
    - The source of the personal data if not provided by the data subject themselves.
    - If and how automated decision-making, profiling is taking place.
2. **The right to fix incorrect data.**
3. **The right of deletion.** *"The right to be forgotten."* This right puts several architectural limites on cloud systems as, for example, backups must be built in such a way that deletion requests are repeated after a restore. Note that this right is not without limits, legal obligations, for example, override it.

4. **The right to restrict processing.** If a deletion cannot be requested the data subject can request that their data should only be processed to the purposes that are strictly required.
5. **The right to data portability.** The data subject has to be provided a copy of their data in a machine-readable form.
6. **The right to object automated individual decision-making and profiling.**

DATA BREACHES

Chapter IV Section 2 is probably the first legislation around the world that explicitly requires security of data processing from companies that handle personal data. Article 33 of this chapter specifically requires that data breaches must be disclosed to the supervisory authorities. Supervisory authorities in turn have the right to impose fines up to 4% or 20.000.000 € of the global revenue of a company.

This means that companies can no longer sweep data breaches under the rug and must spend resources to secure their data processing. In the scope of the cloud this means that our cloud environment has to be set up in a secure way to avoid data breaches.

Clarifying Lawful Overseas Use of Data Act (C.L.O.U.D., 2018, USA)

The CLOUD Act, or House Rule 4943 is the latest addition to the US legislation pertaining to cloud providers. This act says that **a cloud provider must hand over data to US authorities if requested even if that data is stored in a different country.**

This act has been widely criticized and is, according to several legal scholars, in contradiction of the GDPR. The large US cloud providers have opened up subsidiaries in the EU in order to try and shield their european customers from this act and a few purely EU cloud providers have also capitalized on this. It remains to be seen how effective this move is.

Digital Millennium Copyright Act (D.M.C.A., 1998, USA)

The Digital Millenium Copyright Act clarifies how copyright works in the USA. Since the USA is part of the WIPO copyright treaty and a significant amount of providers are based in the USA nowadays all countries align themselves with the DMCA when it comes to dealing with online copyright infringement.

Title II of the DMCA creates a safe harbor for online service providers against copyright infringement committed by their users. However, they have to remove infringing content as soon as they are properly notified of it via a DMCA takedown notice. The author of a DMCA takedown notice must swear by the penalty of perjury that they are, or are acting on behalf of the copyright owner. (If they were to make this pledge in bad faith they could end up in prison.)

If the DMCA takedown notice has been sent erroneously the original uploader is free to send a counter-notification, also swearing under the penalty of perjury. In this case the provider notifies

the original sender and restores the content. The original sender then has to take the uploader to court to pursue the matter further.

Communications Decency Act, Section 230 (C.D.A., 1996, USA)

This section of of the CDA is very short:

> No provider or user of an interactive computer service shall be treated as the publisher or speaker of any information provided by another information content provider.

To someone not well versed in US legal lingo this may be sound like gibberish but it is, in fact, one of the most important pieces of legislation pertaining to operating platforms in the cloud.

In US legal history the speaker or publisher of certain information is responsible for the information being provided. Before the CDA any provider hosting illegal content would be responsible for said content. The CDA changed that by shielding providers from liability. This included web hosting providers as well as newspapers that had a content section.

The CDA is not limitless, providers still have to remove infringing content if notified. They may also employ moderation proactively.

Privacy Shield (2016, EU-US)

The Privacy Shield agreement is the successor to the failed International Safe Harbor Privacy Principles framework and is intend to protect the privacy of EU citizens when using US services or their data is transferred to the USA.

The Privacy Shield has been enhanced in 2017 by the EU–US Umbrella Agreement which fixes many of the issues with the Privacy Shield. While there are still valid criticisms it remains to be seen if this agreement will also be declared invalid as the predecessor.