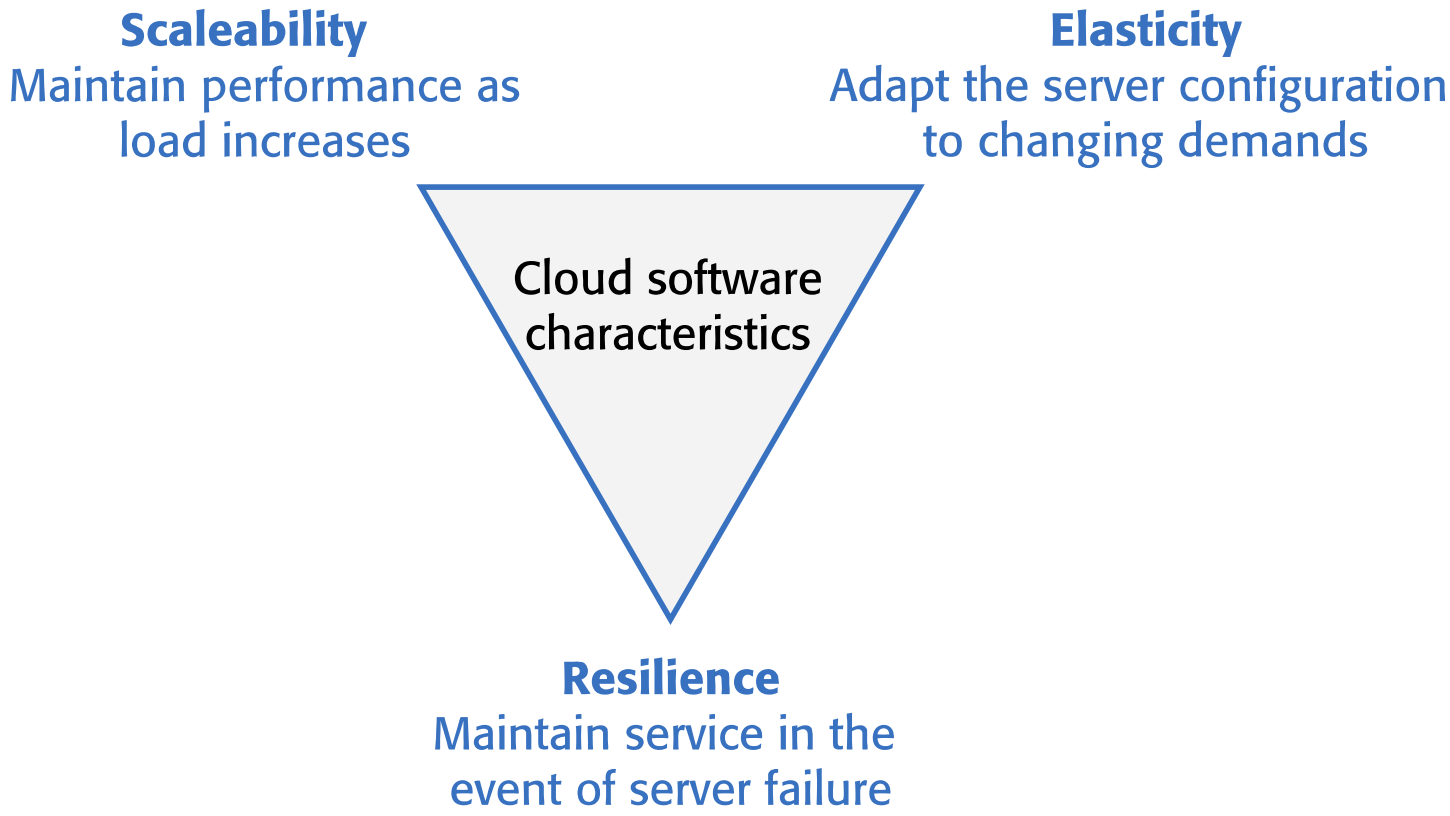
Облачни (Cloud) технологии и виртуализация

The cloud

- The cloud is made up of very large number of remote servers that are offered for rent by companies that own these servers.
  - Cloud-based servers are 'virtual servers', which means that they are implemented in software rather than hardware.
- You can rent as many servers as you need, run your software on these servers and make them available to your customers.
  - Your customers can access these servers from their own computers or other networked devices such as a tablet or a TV.
  - Cloud servers can be started up and shut down as demand changes.
- You may rent a server and install your own software, or you may pay for access to software products that are available on the cloud.

Scaleability, elasticity and resilience



Scaleability, elasticity and resilience

- **Scaleability** reflects the ability of your software to cope with increasing numbers of users.
  - As the load on your software increases, your software automatically adapts so that the system performance and response time is maintained.
- **Elasticity** is related to scaleability but also allows for scaling-down as well as scaling-up.
  - That is, you can monitor the demand on your application and add or remove servers dynamically as the number of users change.
- **Resilience** means that you can design your software architecture to tolerate server failures.
  - You can make several copies of your software concurrently available. If one of these fails, the others continue to provide a service.

Benefits of using the cloud for software development

- ***Cost***
  You avoid the initial capital costs of hardware procurement
- ***Startup time***
  You don't have to wait for hardware to be delivered before you can start work. Using the cloud, you can have servers up and running in a few minutes.
- ***Server choice***
  If you find that the servers you are renting are not powerful enough, you can upgrade to more powerful systems. You can add servers for short-term requirements, such as load testing.
- ***Distributed development***
  If you have a distributed development team, working from different locations, all team members have the same development environment and can seamlessly share all information.

Virtualization

- *"**Virtualization** is a process that allows for more efficient utilization of physical computer hardware and is the foundation of cloud computing."*

- *"**Virtualization** uses software to create an abstraction layer over computer hardware that allows the hardware elements of a single computer—processors, memory, storage and more—to be divided into multiple virtual computers, commonly called virtual machines (VMs). Each VM runs its own operating system (OS) and behaves like an independent computer, even though it is running on just a portion of the actual underlying computer hardware."*

Source: https://www.ibm.com/topics/virtualization

Virtual cloud servers

- A **virtual server** runs on an underlying physical computer and is made up of an operating system plus a set of software packages that provide the server functionality required.
- A **virtual server** is a stand-alone system that can run on any hardware in the cloud.
  - This 'run anywhere' characteristic is possible because the virtual server has no external dependencies.
- **Virtual machines (VMs),** running on physical server hardware, can be used to implement virtual servers.
  - A hypervisor provides hardware emulation that simulates the operation of the underlying hardware.
- If you use a virtual machine to implement virtual servers, you have exactly the same hardware platform as a physical server.

Types of Virtualization

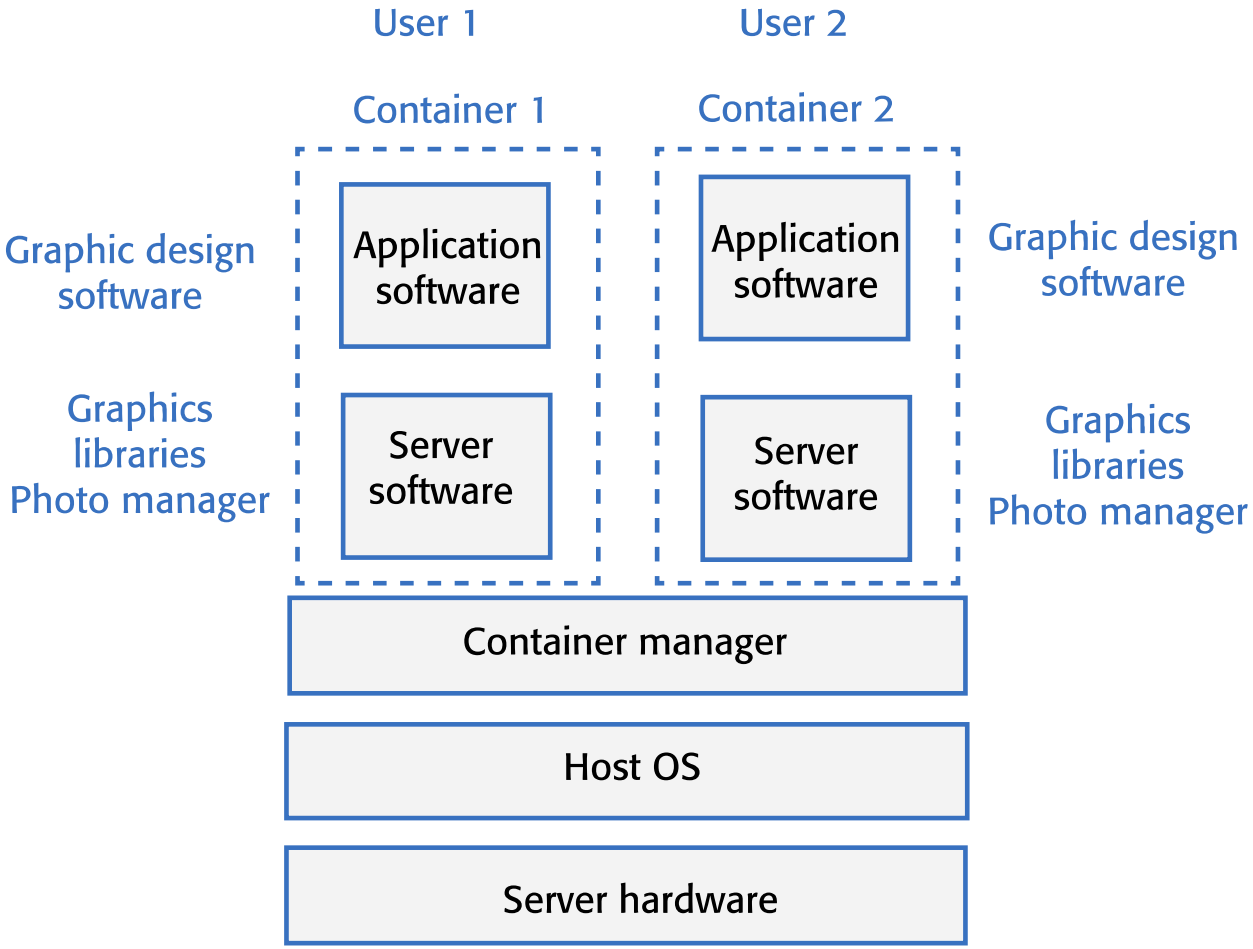
- Desktop virtualization
- Network virtualization
- Storage virtualization
- Data virtualization
- Application virtualization
- Data center virtualization
- CPU virtualization
- GPU virtualization
- Linux virtualization
- Cloud virtualization *(IaaS, PaaS, SaaS)*

Source: https://www.ibm.com/topics/virtualization

Container-based virtualization

- If you are running a cloud-based system with many instances of applications or services, these all use the same operating system, you can use a simpler virtualization technology called 'containers'.
- Using containers accelerates the process of deploying virtual servers on the cloud.
  - Containers are usually megabytes in size whereas VMs are gigabytes.
  - Containers can be started and shut down in a few seconds rather than the few minutes required for a VM.
- **Containers are an operating system virtualization technology that allows independent servers to share a single operating system.**
  - They are particularly useful for providing isolated application services where each user sees their own version of an application.

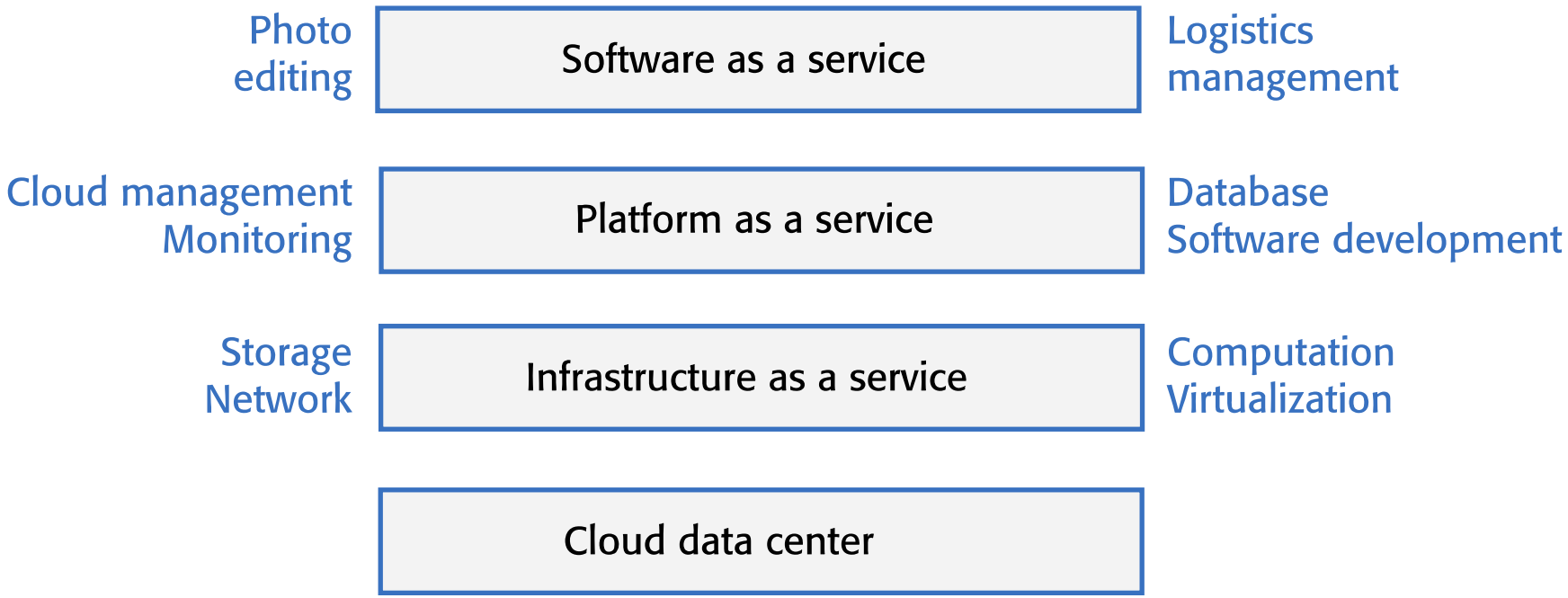
Using containers to provide isolated services



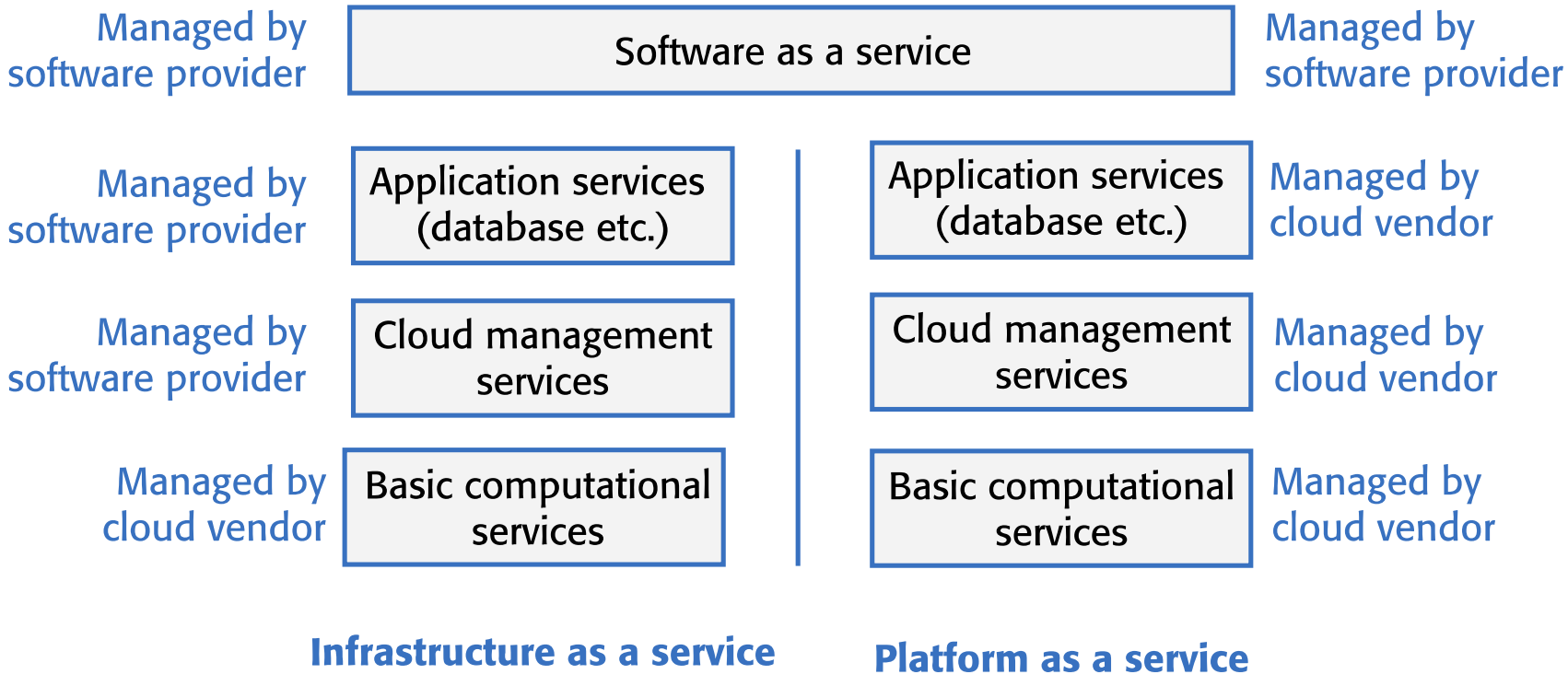
Everything as a service

- The idea of a service that is rented rather than owned is fundamental to cloud computing.
- **Infrastructure as a service (IaaS)**
  - Cloud providers offer different kinds of infrastructure service such as a compute service, a network service and a storage service that you can use to implement virtual servers.
- **Platform as a service (PaaS)**
  - This is an intermediate level where you use libraries and frameworks provided by the cloud provider to implement your software. These provide access to a range of functions, including SQL and NoSQL databases.
- **Software as a service (SaaS)**
  - Your software product runs on the cloud and is accessed by users through a web browser or mobile app.

Everything as a service



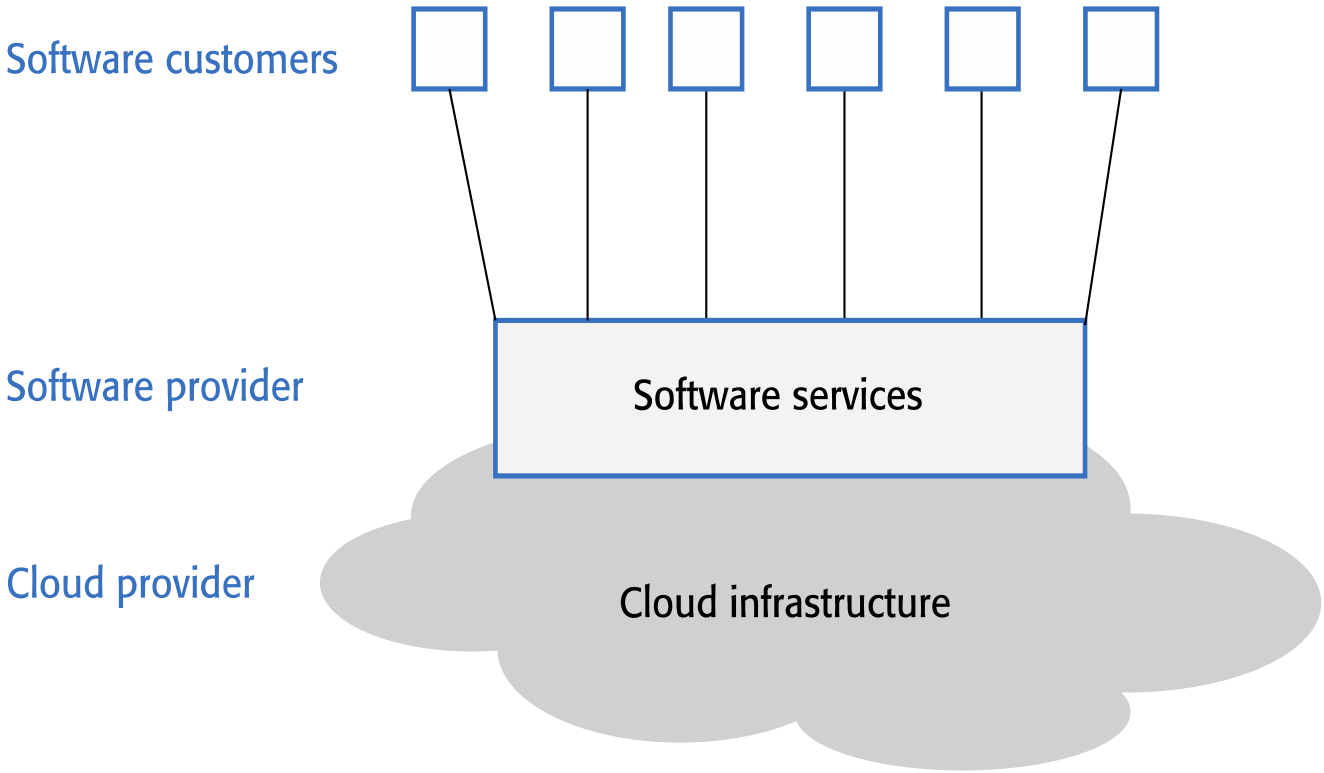
Management responsibilities for IaaS and PaaS



Software as a service

- Increasingly, software products are being delivered as a service, rather than installed on the buyer's computers.
- If you deliver your software product as a service, you run the software on your servers, which you may rent from a cloud provider.
- Customers don't have to install software and they access the remote system through a web browser or dedicated mobile app.
- The payment model for software as a service is usually a subscription model.
  - Users pay a monthly fee to use the software rather than buy it outright.

Software as a service



Benefits of SaaS for software product providers

- ***Cash flow***
  Customers either pay a regular subscription or pay as they use the software. This means you have a regular cash flow, with payments throughout the year. You don't have a situation where you have a large cash injection when products are purchased but very little income between product releases.

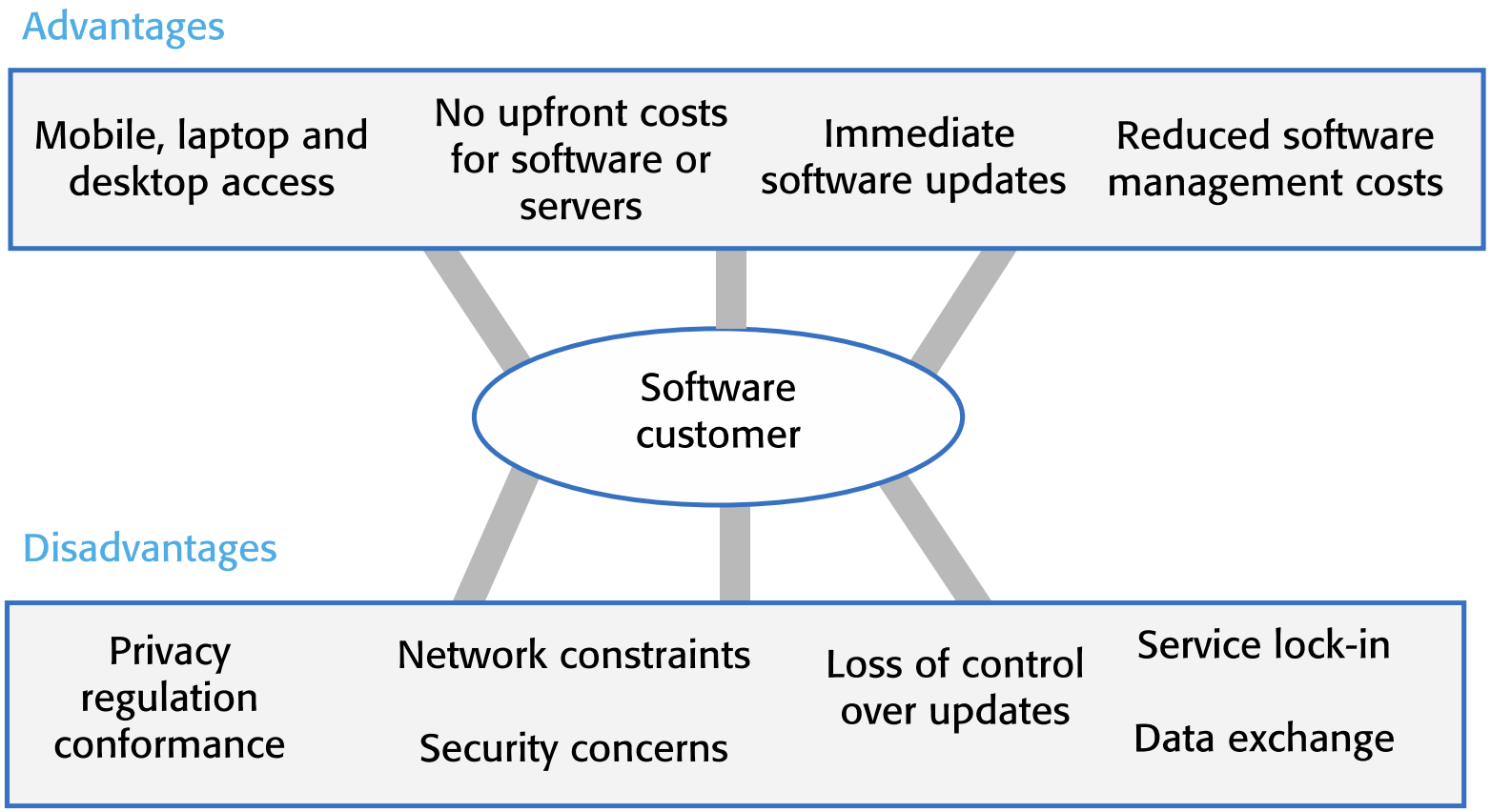
- ***Update management***
  You are in control of updates to your product and all customers receive the update at the same time. You avoid the issue of several versions being simultaneously used and maintained. This reduces your costs and makes it easier to maintain a consistent software code base.

- ***Continuous deployment***
  You can deploy new versions of your software as soon as changes have been made and tested. This means you can fix bugs quickly so that your software reliability can continuously improve.

Benefits of SaaS for software product providers

- ***Payment flexibility***
  You can have several different payment options so that you can attract a wider range of customers. Small companies or individuals need not be discouraged by having to pay large upfront software costs.
- ***Try before you buy***
  You can make early free or low-cost versions of the software available quickly with the aim of getting customer feedback on bugs and how the product could be approved.
- ***Data collection***
  You can easily collect data on how the product is used and so identify areas for improvement. You may also be able to collect customer data that allows you to market other products to these customers.

Advantages and disadvantages of SaaS for customers



Data storage and management issues for SaaS

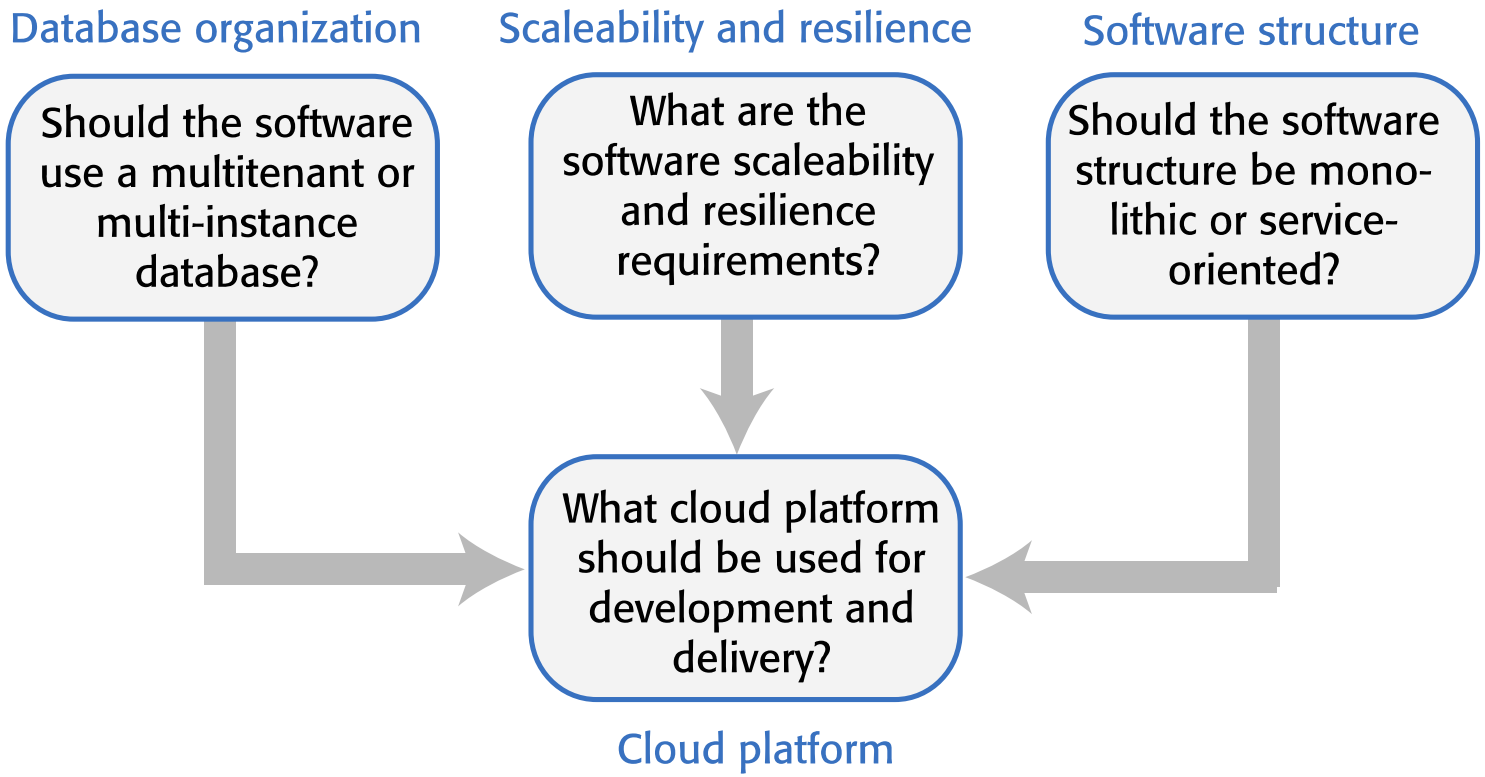
- ***Regulation***
  Some countries, such as EU countries, have strict laws on the storage of personal information. These may be incompatible with the laws and regulations of the country where the SaaS provider is based. If a SaaS provider cannot guarantee that their storage locations conform to the laws of the customer's country, businesses may be reluctant to use their product.

- ***Data transfer***
  If software use involves a lot of data transfer, the software response time may be limited by the network speed. This is a problem for individuals and smaller companies who can't afford to pay for very high speed network connections.

- ***Data security***
  Companies dealing with sensitive information may be unwilling to hand over the control of their data to an external software provider. As we have seen from a number of high profile cases, even large cloud providers have had security breaches. You can't assume that they always provide better security than the customer's own servers.

- ***Data exchange***
  If you need to exchange data between a cloud service and other services or local software applications, this can be difficult unless the cloud service provides an API that is accessible for external use.

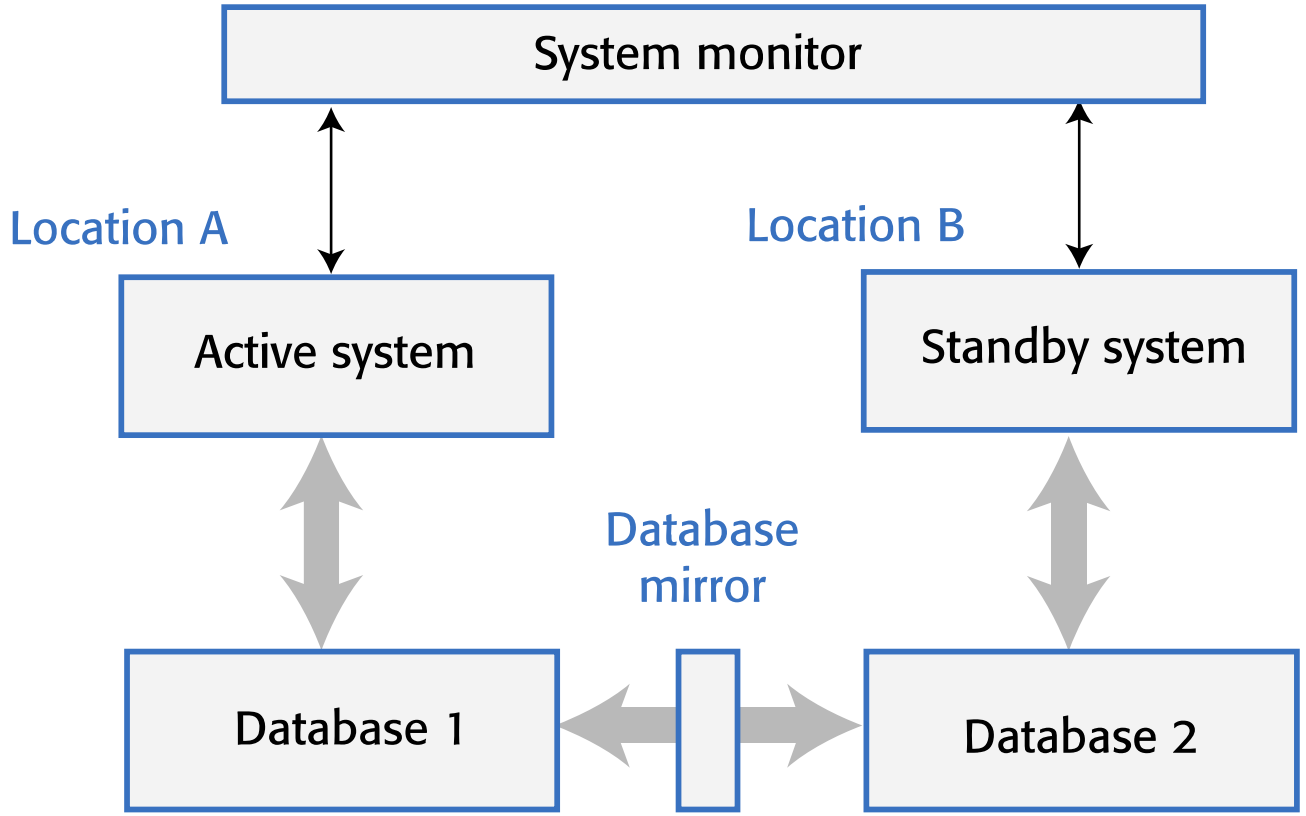
Architectural decisions for cloud software engineering



Scalability and resilience

- The **scaleability** of a system reflects its ability to adapt automatically to changes in the load on that system.
- The **resilience** of a system reflects its ability to continue to deliver critical services in the event of system failure or malicious system use.
- You achieve **scaleability** in a system by making it possible to **add new virtual servers** (**scaling-out)** or **increase the power of a system server (scaling-up)** in response to increasing load.
  - In cloud-based systems, scaling-out rather than scaling-up is the normal approach used. Your software has to be organized so that individual software components can be replicated and run in parallel.
- To achieve **resilience**, you need to be able to restart your software quickly after a hardware or software failure.

Using a standby system to provide resilience



Resilience

- Resilience relies on redundancy:
  - Replicas of the software and data are maintained in different locations.
  - Database updates are mirrored so that the standby database is a working copy of the operational database.
  - A system monitor continually checks the system status. It can switch to the standby system automatically if the operational system fails.
- You should use redundant virtual servers that are not hosted on the same physical computer and locate servers in different locations.
  - Ideally, these servers should be located in different data centers.
  - If a physical server fails or if there is a wider data center failure, then operation can be switched automatically to the software copies elsewhere.

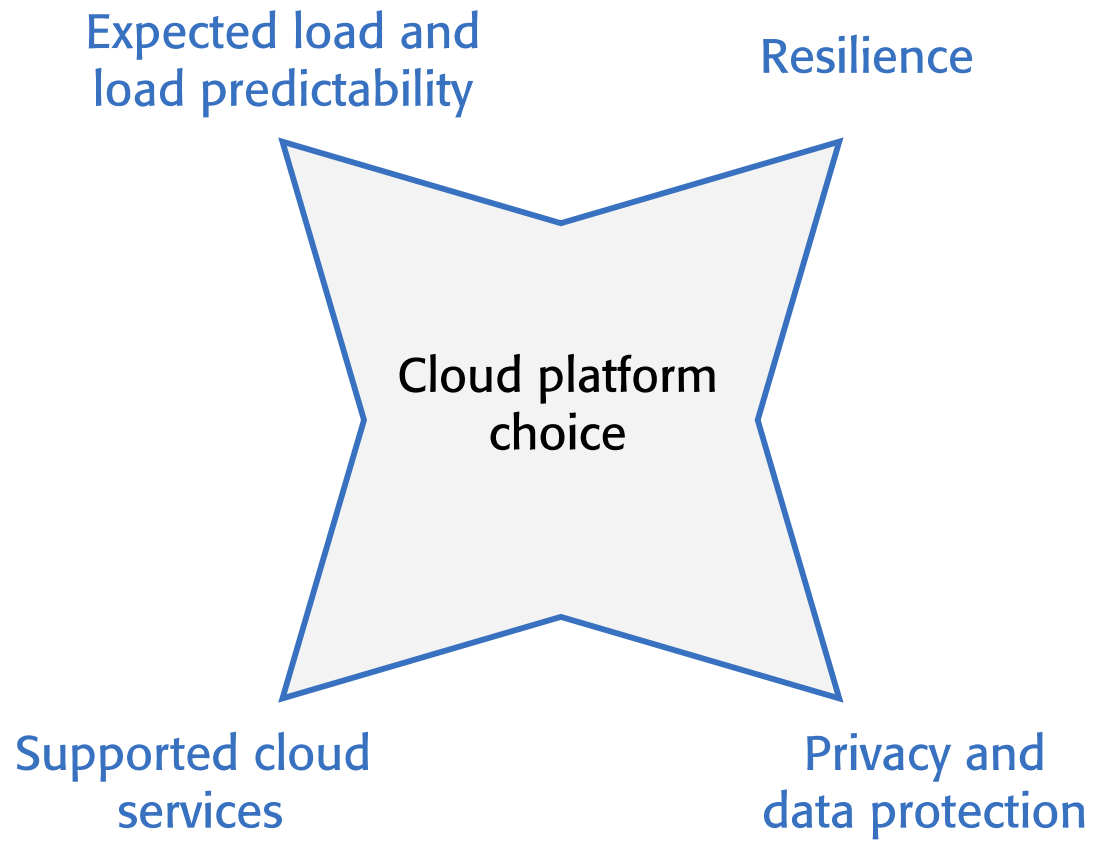
System structure

- An object-oriented approach to software engineering has been that been extensively used for the development of client-server systems built around a shared database.
- The system itself is, logically, a monolithic system with distribution across multiple servers running large software components. The traditional multi-tier client server architecture is based on this distributed system model.
- The alternative to a monolithic approach to software organization is a service-oriented approach where the system is decomposed into fine-grain, stateless services.
  - Because it is stateless, each service is independent and can be replicated, distributed and migrated from one server to another.
  - The service-oriented approach is particularly suitable for cloud-based software, with services deployed in containers.

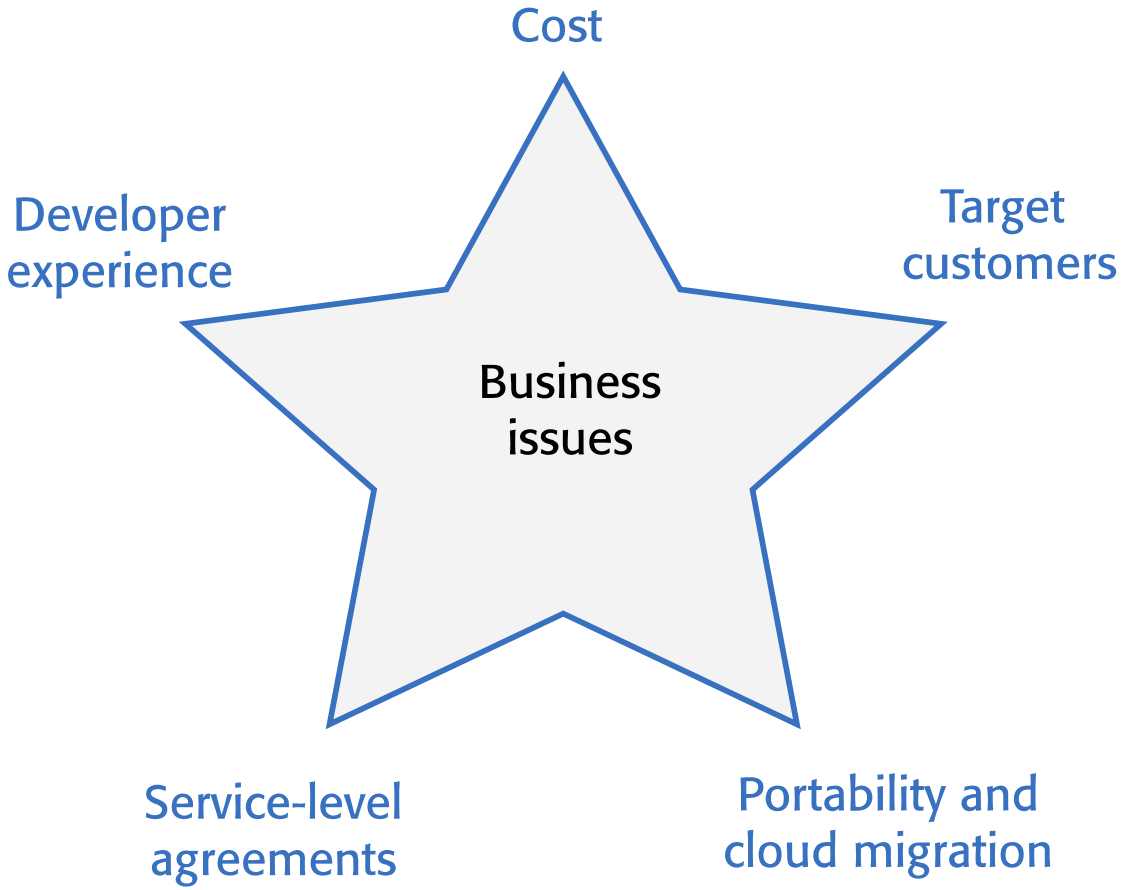
Cloud platform

- Cloud platforms include general-purpose clouds such as Amazon Web Services or lesser known platforms oriented around a specific application, such as the SAP Cloud Platform. There are also smaller national providers that provide more limited services but who may be more willing to adapt their services to the needs of different customers.
- There is no 'best' platform and you should choose a cloud provider based on your background and experience, the type of product that you are developing and the expectations of your customers.
- You need to consider both technical issues and business issues when choosing a cloud platform for your product.

Technical issues in cloud platform choice



Business issues in cloud platform choice



So technically Moses was the first man to download files from the cloud using a tablet.

Source: https://www.facebook.com/photo/?fbid=754174989668221&set=a.349158056836585

Conclusion: Key points 1

- The cloud is made up of a large number of virtual servers that you can rent for your own use. You and your customers access these servers remotely over the internet and pay for the amount of server time used.
- Virtualization is a technology that allows multiple server instances to be run on the same physical computer. This means that you can create isolated instances of your software for deployment on the cloud.
- Virtual machines are physical server replicas on which you run your own operating system, technology stack and applications.
- Containers are a lightweight virtualization technology that allow rapid replication and deployment of virtual servers. All containers run the same operating system.
- A fundamental feature of the cloud is that 'everything' can be delivered as a service and accessed over the internet. A service is rented rather than owned and is shared with other users.

Conclusion: Key points 2

- Infrastructure as a service (IaaS) means computing, storage and other services are available over the cloud. There is no need to run your own physical servers.
- Platform as a service (PaaS) means using services provided by a cloud platform vendor to make it possible to auto-scale your software in response to demand.
- Software as a service (SaaS) means that application software is delivered as a service to users. This has important benefits for users, such as lower capital costs, and software vendors, such as simpler deployment of new software releases.

References

- *Sommerville, I. Software Engineering. 10th edition, Published by Pearson Education, ISBN: 978-1-292-09613-1 (2016)*
- *Pressman, R., Maxim, B. Software Engineering: A Practitioner's Approach. 9th edition, Published by McGraw-Hill Education, ISBN: 9781260548006, (2019)*
- *Virtualization Overview - https://www.ibm.com/topics/virtualization*