纵观云计算和区块链

我们在研究区块链的过程中发现，区块链的发展和云计算有非常多的相似之处，因此便有了此文，带领大家从宏观的角度认识区块链和云计算。

概述，区块链与云计算相似的地方

	云计算	区块链
底层技术三要素	计算、网络、存储	算法、网络、账本
类型	公有云、私有云、混合云	公有链、私有链，联盟链
形态	IaaS、PaaS、BaaS、SaaS……	Smart Contract、DLT、Crypto Currency……

底层三要素

云计算的底层三要素

计算虚拟化

计算虚拟化就是在虚拟系统和底层硬件之间抽象出CPU和内存等，以供虚拟机使用。计算虚拟化技术需要模拟出一套操作系统的运行环境，在这个环境你可以安装Windows也是可以安装Linux，这些操作系统被称作Guest OS。他们相互独立，互不影响（相对的，因为当主机资源不足会出现竞争等问题，导致运行缓慢等问题）。计算虚拟化可以将主机单个物理核虚拟出多个vcpu，这些vcpu本质上就是运行的进程，考虑到系统调度，所以并不是虚拟的核数越多越好；内存相似的，把物理机上面内存进行逻辑划分出多个段，供不同的虚拟机使用，每个虚拟机看到的都是自己独立的一个内存。除了这些还需要模拟网络设备、BIOS等。这个虚拟化软件叫做hypervisor，著名的有ESXI、xen、KVM等，通常分为两种，第一种是直接部署到物理服务器上面的，如下图ESXI

由于直接部署到裸机上面，hypervisor需要自带各种硬件驱动，虚拟机的所有操作都需要经过hypervisor。还有另一种虚拟化hypervisor，以KVM最为流行（个人电脑上面安装的virtualbox以及workstations也是），它们依赖与宿主机操作系统，这样的好处就是可以充分利用宿主机的各种资源管理以及驱动，但效率上面会打一些折扣。下图是KVM的在使用IO时候的流程图。

当然也可以从全虚拟化、半虚拟化、硬件辅助虚拟化的角度去说，现在数据中心基本都是硬件辅助虚拟化了，全虚拟化就是完全靠软件模拟、半虚拟需要修改操作让其知道自己运行在虚拟环境中、硬件辅助由硬件为每个Guest OS提供一套寄存器、Guest OS可以直接运行在特权级，这样提高效率。

虽然当前数据中心商用的虚拟化软件仍然以VMware的ESXI为主，但在OpenStack的推动下，KVM正在慢慢追赶，并且KVM是开源的，下面简单介绍一下KVM。KVM是基于内核的，从内核2.6以后就自带了，可以运行在x86和power等主流架构上。 KVM主要是CPU和内存的虚拟化，其它设备的虚拟化和虚拟机的管理则需要依赖QEMU完成。一个虚拟机本质上就是一个进程，运行在QEMU-KVM进程地址空间，KVM（内核空间）和qemu（用户空间）相结合一起向用户提供完整的虚拟化环境。

网络虚拟化

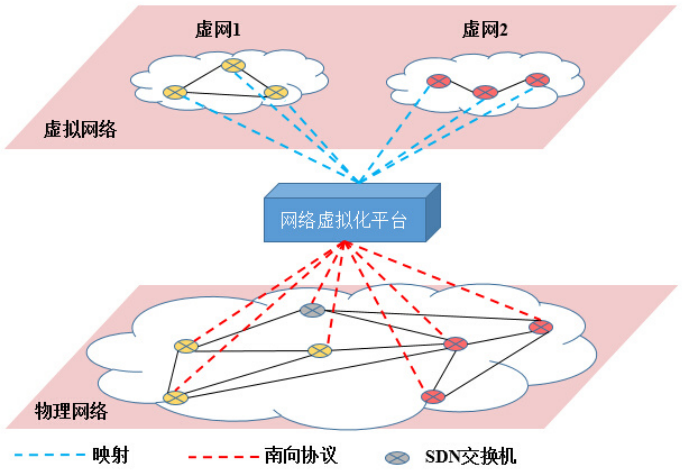
网络虚拟化是一种重要的网络技术，该技术可在物理网络上虚拟多个相互隔离的虚拟网络，不依赖于底层物理连接，能够动态变化网络拓扑，提供多租户隔离，从而使得不同用户之间使用独立的网络资源切片变成可能，从而提高网络资源利用率，实现弹性的网络。这里面目前最为火热的即软件定义网络（Software Defined Network, SDN），SDN的出现使得网络虚拟化的实现更加灵活和高效，同时网络虚拟化也成为SDN应用中的重量级应用。其核心技术OpenFlow通过将网络设备控制面与数据面分离开来，从而实现了网络流量的灵活控制，使网络作为管道变得更加智能。

通过SDN实现网络虚拟化包括物理网络管理，网络资源虚拟化和网络隔离三部分。而这三部分内容往往通过专门的中间层软件完成，我们称之为网络虚拟化平台。虚拟化平台需要完成物理网络的管理和抽象虚拟化，并分别提供给不同的租户。此外，虚拟化平台还应该实现不同租户之间的相互隔离，保证不同租户互不影响。虚拟化平台的存在使得租户无法感知到网络虚拟化的存在，也即虚拟化平台可实现用户透明的网络虚拟化。

虚拟化平台

虚拟化平台是介于数据网络拓扑和租户控制器之间的中间层。面向数据平面，虚拟化平面就是控制器；而面向租户控制器，虚拟化平台就是数据平面。所以虚拟化平台本质上具有数据平面和控制层面两种属性。在虚拟化的核心层，虚拟化平台需要完成物理网络资源到虚拟资源的虚拟化映射过程。面向租户控制器，虚拟化平台充当数据平面角色，将模拟出来的虚拟网络呈现给租户控制器。从租户控制器上往下看，只能看到属于自己的虚拟网络，而并不了解真实的物理网络。而在数据层面的角度看，虚拟化平台就是控制器，而交换机并不知道虚拟平面的存在。所以虚拟化平台的存在实现了面向租户和面向底层网络的透明虚拟化，其管理全部的物理网络拓扑，并向租户提供隔离的虚拟网络。

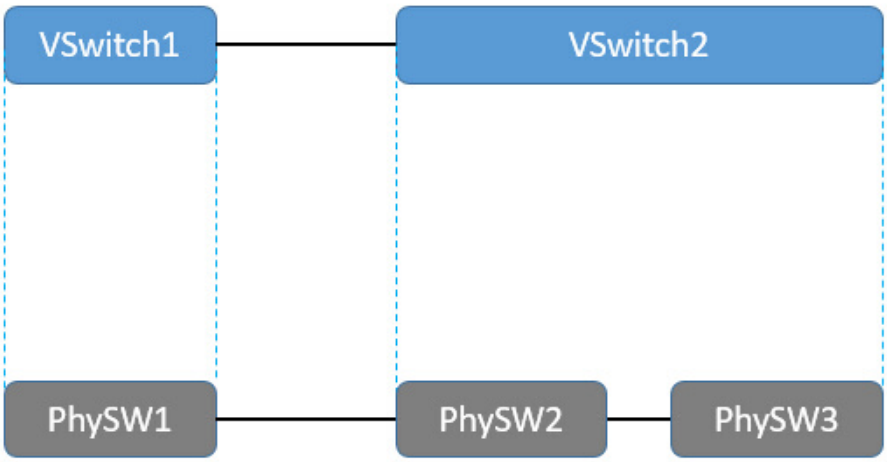
网络虚拟化平台示意图



虚拟化平台不仅可以实现物理拓扑到虚拟拓扑“一对一”的映射，也应该能实现物理拓扑“多对一”的映射。而由于租户网络无法独占物理平

面的交换机，所以本质上虚拟网络实现了“一虚多”和“多虚一”的虚拟化。此处的“一虚多”是指单个物理交换机可以虚拟映射成多个虚拟租户网中的逻辑交换机，从而被不同的租户共享；“多虚一”是指多个物理交换机和链路资源被虚拟成一个大型的逻辑交换机。即租户眼中的一个交换机可能在物理上由多个物理交换机连接而成。

单虚拟节点映射到多物理节点



网络资源虚拟化

为实现网络虚拟化，虚拟化平台需要对物理网络资源进行抽象虚拟化，其中包括拓扑虚拟化，节点资源虚拟化和链路资源虚拟化。

拓扑虚拟化

拓扑虚拟化是网络虚拟化平台最基本的功能。虚拟平台需要完成租户虚网中的虚拟节点和虚拟链路到物理节点和链路的映射。其中包括“一对一”和“一对多”的映射。“一对一”的映射中，一个虚拟节点将会映射成一个物理节点，同理虚拟链路也是。而在“一对多”的映射中，一个虚拟节点可以映射成由多个连接在一起的物理节点；一条逻辑链路也可能映射成由链接在一起的多条链路。而对于物理节点而言，一个物理节点可以被多个逻辑节点映射。

节点资源虚拟化

节点资源的虚拟化包括对节点Flow
tables（流表）、CPU等资源的抽象虚拟化。流表资源本身是交换机节点的稀缺资源，如果能对其进行虚拟化，然后由虚拟化平台对其进行分配，分配给不同的租户，那么就可以实现不同租户对节点资源使用的分配和限制。拓扑抽象仅仅完成了虚拟节点到物理节点的映射，而没有规定不同用户/租户对物理节点资源使用的分配情况。若希望进行更细粒度的网络虚拟化，节点资源虚拟化非常有必要。

链路资源虚拟化

和节点资源一样，链路资源也是网络中重要的资源，而拓扑抽象并没有规定某些用户可使用的链路资源的多少。所以在进行更细粒度的虚拟化时，有必要对链路资源进行虚拟化，从而实现链路资源的合理分配。可被抽象虚拟化的链路资源包括租户可使用的带宽以及端口的队列资源等等。

网络隔离

网络资源虚拟化仅仅完成了物理资源到虚拟资源的抽象过程，为实现完全的网络虚拟化，还需要对不同的租户提供隔离的网络资源。网络隔离需要对SDN的控制平面和数据平面进行隔离，从而保证不同租户控制器之间互补干扰，不同虚网之间彼此隔离。此外，为了满足用户对地址空间自定义的需求，虚拟化平台还需要对网络地址进行虚拟化。

控制面隔离

控制器的性能对SDN整体的性能产生极大的影响，所以虚拟化平台需保证租户的控制器在运行时不受其他租户控制器的影响，保证租户对虚拟化平台资源的使用。虚拟化平台在连接租户控制器时需保证该进程可以得到一定的资源保障，比如CPU资源。而虚拟化平台本身所处的位置就可以轻易实现租户的控制器之间的相互隔离。

数据面隔离

数据面的资源包括节点的CPU、Flow
Tables等资源以及链路的带宽，端口的队列资源等。为保证各个租户的正常使用，需对数据面的资源进行相应的隔离，从而保证租户的资源不被其他租户所占据。若在数据面上不进行资源的隔离，则会产生租户数据在数据面上的竞争，从而无法保障租户对网络资源的需求，所以很有必要在数据面对资源进行隔离。

地址隔离

为使租户能在自己的虚拟租户网中任意使用地址，虚拟化平台需要完成地址的隔离。实现地址隔离主要通过地址映射来完成。租户可任意定制地址空间，而这些地址对于虚拟化平台而言是面向租户的虚拟地址。虚拟化平台在转发租户控制器南向协议报文时，需要将虚拟地址转化成全网唯一的物理地址。租户的服务器的地址在发送到接入交换机时就会被修改成物理地址，然后数据包的转发会基于修改之后的物理地址进行转发。当数据到达租户目的地址主机出端口，控制器需将地址转换成原来租户设定的地址，从而完成地址的虚拟化映射。地址的虚拟化映射使得租户可以使用完全的地址空间，可以使用任意的FlowSpace（流空间：流表匹配项所组成的多维空间），而面向物理层面则实现了地址的隔离，使得不同的租户使用特定的物理地址，数据之间互不干扰。

存储虚拟化

存储虚拟化是一种贯穿于整个IT环境、用于简化本来可能会相对复杂的底层基础架构的技术。存储虚拟化的思想是将资源的逻辑映像与物理存储分开，从而为系统和管理员提供一幅简化、无缝的资源虚拟视图。在没有云计算之前存储虚拟化已经发展了很久，可以说和云计算没有特别关系，而云计算存储通常指的是亚马逊的S3存储或者EBS存储等，将统一的资源池划分给多个用户。

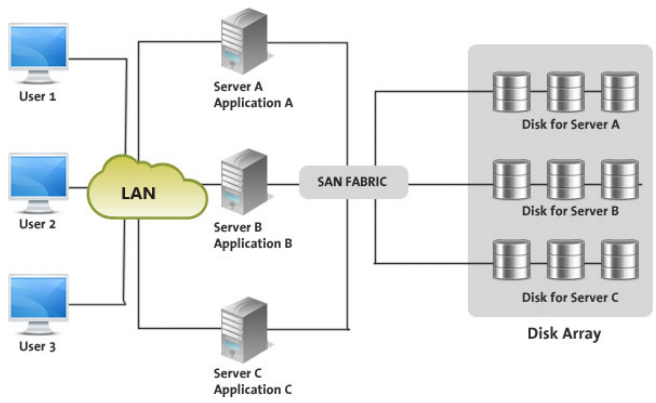
对于用户来说，虚拟化的存储资源就像是一个巨大的“存储池”，用户不会看到具体的磁盘、磁带，也不必关心自己的数据经过哪一条路径通往哪一个具体的存储设备。

从管理的角度来看，虚拟存储池是采取集中化的管理，并根据具体的需求把存储资源动态地分配给各个应用。值得特别指出的是，利用虚拟化技术，可以用磁盘阵列模拟磁带库，为应用提供速度像磁盘一样快、容量却像磁带库一样大的存储资源，这就是当今应用越来越广泛的虚拟磁带库（VTL, Virtual Tape Library），在当今企业存储系统中扮演着越来越重要的角色。

主流的存储虚拟化有以下三种技术，在云计算场景中通常会根据实际场景选择合适的技术。

SAN

先从高端存储说起，现在高端存储应该EMC、IBM和HDS的天下，这些年外置存储跟随着廉价磁盘不断提升容量和性能，推动了SAN网络、主机FC接口不断成熟，在数据中心变得很普遍，尤其在金融领域。

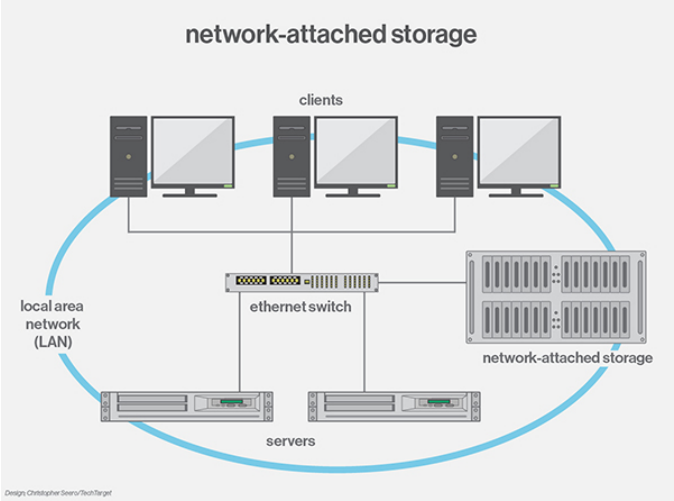


SAN提供的是块存储，譬如磁盘阵列里面有10块I
T的数据盘，然后可以通过做RAID或者逻辑卷（LVM）的方式划分出10个的数据盘，但这个10个数据盘已经和之前的物理盘不一样了，一个逻辑盘可能有第一个物理盘提供100G，第二个物理盘提供300G。对于操作系统来说，完全无法感知是物理盘还是逻辑盘，这是存储资源池的理念。通过RAID或者LVM不仅可以提供数据保护还能够重新划分盘的大小，提高读写速率。

但SAN也不是毫无缺点，它价格也是比较昂贵的，光纤口，光纤交换机价格高，所以才有了IPSAN存储，通过IP协议承载存储协议；无法提供数据共享，一个盘只能挂给一个主机，所以这就有了NAS存储。

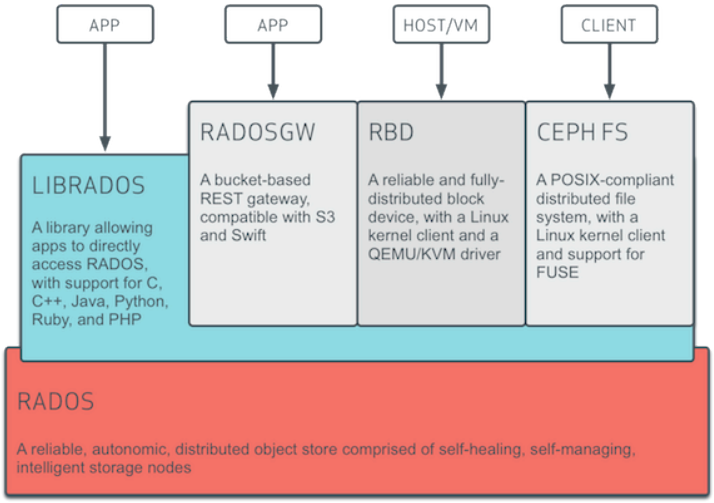
NAS

NAS是文件存储，文件存储相比块存储最大的优势是能共享数据，它基于标准的网络协议，SAN是有自己一套存储协议的。常见的NAS包括NFS、FTP和HTTP文件服务器等，由于这种设备通常都有一个IP，所以一般客户机充当数据网关服务器可以直接对其访问。NAS建立在传统网络之上，所以可以更远距离的传输，并且NAS具有安装容易易于维护的特点，但其速度通常要比SAN慢很多。



分布式存储

伴随着x86性能提升，以x86芯片构建的小型存储系统在中端存储领域开始崭露头角。通过将X86本地的磁盘利用起来构建一个大存储集群。分布式存储通常能够同时提供块存储和文件存储的能力。这里不得介绍一个和OpenStack结合紧密的分布式存储ceph，下图是ceph官网的一个整体模块图，它提供了CEPH FS文件存储系统和POSIX接口、对象存储以及最常用的快存储。



它的基石是下面的RADOS，再下面就是系统组件，包括：

- CEPH OSDs：CEPH的OSD（Object Storage Device）守护进程。主要功能包括：存储数据、副本数据处理、数据恢复、数据回补，平衡数据分布。并将数据相关的一些监控信息提供给CEPH Moniter，以便CEPH Moniter来检查其他OSD的心跳状态。一个CEPH存储集群，要求至少两个CEPH OSDs，才能有效的保存两份数据。注意，这里的两个CEPH OSD是指运行在两台物理服务器上的，并不是在一台物理服务器上开两个CEPH OSD的守护进程。
- Moniters：CEPH的Moniter守护进程，主要功能是维护集群状态的表组，这个表组中包含了多张表，其中有Moniter map、OSD map 、PG(Placement Group) map、CRUSH map。
- MDSs：CEPH的MDS (Metadata Server)守护进程，主要保存的是CEPH Filesystem的元数据。注意，对于CEPH的块设备和CEPH对象存储都不需要CEPH MDS守护进程。CEPH MDS为基于POSIX文件系统的用户提供了一些基础命令的执行，比如ls、find等等，这样可以很大层度上降低CPEH 存储集群的压力。

还有一个开源的对象存储就是openstack的Swift，Swift的初衷就是用廉价的成本来存储容量特别大的数据，swift使用容器来管理对象，允许用户存储、检索和删除对象以及对象的元数据，而这些操作都是通过用户友好的RESTful风格的接口完成。

区块链的底层三要素

共享帐本

共享账本准确的说应该是分布式账本技术，这个技术从实质上说就是一个可以在多个站点、不同地理位置或者多个机构组成的网络里进行分享的资产数据库。在一个网络里的参与者可以获得一个唯一、真实账本的副本。账本里的任何改动都会在所有的副本中被反映出来，反应时间会在几分钟甚至是几秒内。在这个账本里存储的资产可以是金融、法律定义上的、实体的或是电子的资产。在这个账本里存储的资产的安全性和准确性是通过公私钥以及签名的使用去控制账本的访问权，从而实现密码学基础上的维护。根据网络中达成共识的规则，账本中的记录可以由一个、一些或者是所有参与者共同进行更新。

分布式账本技术使用密码哈希算法和数字签名来确保交易的完整性，同时确保共享账本是精确副本，并降低了发生交易欺诈的风险，因为篡改需要同时在许多地方同时执行。密码哈希算法（比如 SHA256 计算算法）能确保对交易输入的任何改动 — 甚至是最细微的改动 — 都会计算出一个不同的哈希值，表明交易输入可能被损坏。数字签名则确保交易源自发送方（已使用私钥签名）而不是冒名顶替者。

共识算法

这里主要讲述区块链在发展过程中出现的五种典型共识算法：PoW、PoS、DPoS、PBFT和联合共识。

早期，比特币Bitcoin作为区块链技术的第一个成功应用率先引入了工作量证明机制（PoW，Proof of Work），工作量证明机制利用了Hash算法在随机性上这个非常重要的特性。PoW机制俗称挖矿，这里挖的是比特币里的每一个区块。每个区块用包含的交易、时间、以及一个自定义数值来计算这个区块的Hash。一个合格的区块的Hash必须满足前N位为零，因此需要不断的调整刚才那三个参数来寻找满足条件的Hash。由于Hash算法足够随机，零的个数越多，算出这个Hash的概率越低。此时，要得到合理的Block Hash需要经过大量尝试计算，计算时间取决于机器的哈希运算速度。当某个节点提供出一个合理的Block Hash值，说明该节点确实经过了大量的尝试计算，这就是工作量证明。当然，并不能得出计算次数的绝对值，因为寻找合格的Hash是一个概率事件。当节点拥有占全网n%的算力时，该节点即有n%的概率率先发布一个合格的区块。

随后，由于PoW这种算法极其耗费计算资源，截至写本文时（2017年8月），据测算，比特币网络消耗的电力就已经高达15TW。因此，随后的NXT等新兴密码学货币提出了一种新的思路即股权证明（PoS，Proof of Stake）。这种模式会根据你持有数字货币的量和时间，决定你可以发布下一个区块的概率。在PoS模式下，有一个名词叫币龄，每个币每天产生1币龄，比如你持有100个币，总共持有了30天，那么，此时你的币龄就为3000，然后按照所有人的币龄根据一个随机算法决定谁来发布下一个区块。这个时候，如果你被选中发布了一个POS区块，你的币龄就会被清空为0重新再来。

PoS也不是没有缺点，最大的缺点就是在于效率上。因此，比特股BitShares提出了委托股权证明机制（DPoS，Delegated Proof of Stake）。它的原理是让每一个持有比特股的人进行投票，由此产生101位代表，
我们可以将其理解为101个超级节点或者矿池，而这101个超级节点彼此的权利是完全相等的。从某种角度来看，DPoS有点像是议会制度或人民代表大会制度。如果代表不能履行他们的职责（当轮到他们时，没能生成区块），他们会被除名，网络会选出新的超级节点来取代他们。

以上的这些共识机制都依赖密码学货币，因为不管是PoW还是PoS，驱动寻找区块的源动力都是发布新区块的货币奖励。对于无代币的系统如HyperLedger Fabric，如何选择共识机制？这时，我们可以回过头看看PBFT。BFT（Byzantine Fault Tolerance，拜占庭容错算法）是很早就提出的分布式容错算法，可以查找拜占庭问题来进一步了解，这里不做详述。PBFT作为BFT的一种实现，是一种状态机副本复制算法，即服务作为状态机进行建模，状态机在分布式系统的不同节点进行副本复制。每个状态机的副本都保存了服务的状态，同时也实现了服务的操作。将所有的副本组成的集合使用大写字母R表示，使用0到|R|-1的整数表示每一个副本。为了描述方便，假设|R|=3f+1，这里f是有可能失效的副本的最大个数。尽管可以存在多于3f+1个副本，但是额外的副本除了降低性能之外不能提高可靠性。

除此之外，还有一种基于投票的联合共识（Voting），以Ripple为代表。这种共识使网络能够基于特殊节点列表达成共识。初始特殊节点列表就像一个俱乐部，要接纳一个新成员，必须由51%的该俱乐部会员投票通过。共识遵循这核心成员的51%权力，外部人员则没有影响力。这种共识方式同样极大的提高了效率，但是却需要确保特殊节点中恶意节点不能超过51%，牺牲的是整个网络的去中心化。

P2P网络

P2P为大众所熟知主要要归功于BitTorrent及BT的流行，而P2P网络的核心概念即彼此连接的多台计算机之间都处于对等的地位，各台计算机有相同的功能，无主从之分，一台计算机既可作为服务器，设定共享资源供网络中其他计算机所使用，又可以作为工作站，整个网络一般来说不依赖专用的集中服务器，也没有专用的工作站。网络中的每一台计算机既能充当网络服务的请求者，又对其它计算机的请求做出响应，提供资源、服务和内容。通常这些资源和服务包括：信息的共享和交换、计算资源（如CPU计算能力共享）、存储共享（如缓存和磁盘空间的使用）、网络共享、打印机共享等。

区块链为了实现分布式账本的能力，同样也采用了P2P网络。分布式账本会分发给网络中的所有成员节点，同时可以阻止任何单个或一组参与者控制底层基础架构或破坏整个系统。网络中的参与者是平等的，都遵守相同的协议。

类型

云计算

公有云

公有云通常指第三方提供商为用户提供的能够使用的云，比如我们经常使用阿里云即是一种公有云。公有云可通过internet使用，价格非常的低廉，核心的属性是共享服务资源。

公有云被认为是云计算的主要形态，目前市场上公有云也是占据了较大的市场份额的，在国内公有云可以分为以下几类。

1. 传统的电信基础设施运营商，比如中国移动、中国联通、中国电信等提供的公有云服务
2. 一类是政府主导的地方性云计算平台，也就是常说的政府云
3. 互联网巨头打造的公有云平台 如盛大云
4. 部分IDC运营商 如世纪互联
5. 一类为具有国外技术背景或引进国外云计算技术的国内企业

由于目前国内并未开放外国公司在中国直接进行云计算业务，因此像亚马逊、IBM、Joyent、Rackspaces等国外已有多年云计算业务经验的厂商在进入中国市场途中仍障碍重重。2012年11月1日，微软终于实现旗下公有云计算平台Windows Azure在中国的落地，这将掀开外资企业进军中国云计算市场的序幕。

私有云

私有云是为一个用户单独使用而构建的，因而在数据安全性以及服务质量上自己可以有效的管控，私有云的基础是首先你要拥有基础设施并可以控制在此设施上部署应用程序的方式，私有云可以部署在企业数据中心的防火墙内，核心属性是专有资源。

私有云可以搭建在公司的局域网上，与公司内部的公司的监控系统、资产管理系统等相关系统进行打通，从而更有利于公司内部系统的集成管理。

私有云虽然数据安全性方面比公有云高，但是维护的成本也相对较大（对于中小企业而言），因此一般只有大型的企业会采用这类的云平台，因为对于这些企业而言，业务数据这条生命线不能被任何其他的市场主体获取到，与此同时，一个企业尤其是互联网企业发展到一定程度之后，自身的运维人员以及基础设施都已经比较充足完善了，搭建自己的私有云有时候成本反而会比公有云来得低（所谓的规模经济）。

混合云

混合云则是融合了公有云与私有云的优劣势，近几年来混合云模式也得以快速发展起来。混合云综合了数据安全性以及资源共享性双重方面的考虑，个性化的方案达到了省钱安全的目的，从而获得越来越多企业的青睐。但混合云也并不是完美无缺的，以下几个问题需要格外注意。

1. 数据冗余能力：混合云缺少数据冗余，对于数据而言，做好冗余以及容灾备份是非常有必要的，因此若缺乏数据冗余能力的话，实际上数据安全性也不能得到很好的保证。
2. 法律方面：由于是两个云的集合，因此在法律法规上必须确保公有云和私有云提供商符合法律规范，而且你必须要证明两个云之间是顺从的。
3. SLA（服务质量）相比于私有云而言有可能会略差，这里的SLA指的是标准统一性（统一）。在你的私有云的可用性和性能的显示工作负载下收集数据。集成公有云和私有云寻求潜在的问题都会破坏服务。比如：如果一个私有云的关键业务驱动在本地保持敏感和机密数据，然后你的SLA应该体现出在公有云中使用这些服务的限制性。
4. 风险成本或者学习成本较高。从安全角度而言，混合云虽然兼有了私有云的安全性，但是随之带来的却是应由于API带来的复杂网络配置使得传统系统管理员的知识经验及能力受到挑战，随之带来的并是高昂的学习成本或者系统管理员能力不足带来的额外风向。

区块链

公有链

公有链是指全世界任何人都可读取的、任何人都能发送交易且交易能获得有效确认的、任何人都能参与其中共识过程的区块链——共识过程决定哪个区块可被添加到区块链中和明确当前状态。公有链通常被认为是“完全去中心化”的。

公有链的特点：

保护用户免受开发者的影响

在公有链中程序开发者无权干涉用户，所以区块链可以保护使用他们开发的程序的用户。

访问门槛低

任何拥有足够技术能力的人都可以访问，也就是说，只要有一台能够联网的计算机就能够满足访问的条件。

所有数据默认公开

尽管所有关联的参与者都隐藏自己的真实身份，这种现象十分的普遍。他们通过他们的公共性来产生自己的安全性，在这里每个参与者可以看到所有的账户余额和其所有的交易活动。

私有链

私有链是指其写入权限仅在一个组织手里的区块链。读取权限或者对外开放，或者被任意程度地进行了限制。

私有链的特点：

交易速度非常之快

一个私有链的交易速度可以比任何其他的区块链都快，甚至接近了并不是一个区块链的常规数据库的速度。

这是因为就算少量的节点也都具有很高的信任度，并不需要每个节点来验证一个交易。

给隐私更好的保障

私有链使得在那个区块链上的数据隐私政策像在另一个数据库中似的完全一致；不用处理访问权限和使用所有的老办法，但至少说，这个数据不会公开地被拥有网络连接的任何人获得。

交易成本大幅降低甚至为零

私有链上可以进行完全免费或者至少说是非常廉价的交易。如果一个实体机构控制和处理所有的交易，那么他们就不再需要为工作而收取费用。

然而，即使交易的处理是由多个实体机构完成的，例如竞争性银行，进一步举例来说，因为同样的原因，它们可以如此之快的处理交易，所以费用仍然是非常小的；这并不需要节点之间的完全协议，所以很少的节点需要为任何一个交易而工作。

有助于保护其基本的产品不被破坏

正是这一点使得银行等金融机构能在目前的环境中欣然接受私有链，银行和政府在看管他们的产品上拥有既得利益，用于跨国贸易的国家法定货币仍然是有价值的。

由于公有链的直接应用是保护像比特币这样新型的非国家性质的货币，对核心利润流或组织构成了破坏性的威胁，这些实体机构应该会不惜一切代价去避免损害。

联盟链

联盟链是指其共识过程受到预选节点控制的区块链；例如，不妨想象一个有15个金融机构组成的共同体，每个机构都运行着一个节点，而且为了使每个区块生效需要获得其中10个机构的确认。区块链或许允许每个人都可读取，或者只受限于参与者，或走混合型路线，例如区块的根哈希及其API（应用程序接口）对外公开，API可允许外界用来作有限次数的查询和获取区块链状态的信息。这些区块链可视为“部分去中心化”。

形态

云计算

IaaS（Infrastructure-as-a-Service，基础设施即服务）

第一层叫做IaaS，有时候也叫做Hardware-as-a-Service，几年前如果你想在办公室或者公司的网站上运行一些企业应用，你需要去买服务器，或者别的高昂的硬件来控制本地应用，让你的业务运行起来。但是现在有IaaS，你可以将硬件外包到别的地方去。IaaS公司会提供场外服务器，存储和网络硬件，你可以租用。节省了维护成本和办公场地，公司可以在任何时候利用这些硬件来运行其应用。

一些大的IaaS公司包括Amazon, Microsoft, VMWare, Rackspace和Red Hat.不过这些公司又都有自己的专长，比如Amazon和微软给你提供的不只是IaaS，他们还会将其计算能力出租给你来host你的网站。

作用

通过IaaS这种模式，用户可以从供应商那里获得他所需要的虚拟机或者存储等资源来装载相关的应用，同时这些基础设施的繁琐的管理工作将由IaaS供应商来处理。IaaS能通过它上面对虚拟机支持众多的应用。IaaS主要的用户是系统管理员。

产品

主要产品包括：Amazon EC2，Linode，Joyent，Rackspace，IBM Blue Cloud和Cisco UCS等。

功能

IaaS供应商需要在那些方面对基础设施进行管理以给用户提供资源?或者说IaaS云有那些功能?在《虚拟化与云计算》中列出了IaaS的七个基本功能：

资源抽象：使用资源抽象的方法(比如，资源池)能更好地调度和管理物理资源。

资源监控：通过对资源的监控，能够保证基础实施高效率的运行。

负载管理：通过负载管理，不仅能使部署在基础设施上的应用运能更好地应对突发情况，而且还能更好地利用系统资源。

数据管理：对云计算而言，数据的完整性，可靠性和可管理性是对IaaS的基本要求。

资源部署：也就是将整个资源从创建到使用的流程自动化。

安全管理：IaaS的安全管理的主要目标是保证基础设施和其提供的资源能被合法地访问和使用。

计费管理：通过细致的计费管理能使用户更灵活地使用资源。

PaaS（Platform-as-a-Service，平台即服务）

第二层就是所谓的PaaS，某些时候也叫做中间件。你公司所有的开发都可以在这一层进行，节省了时间和资源。

PaaS公司在网上提供各种开发和分发应用的解决方案，比如虚拟服务器和操作系统。这节省了你在硬件上的费用，也让分散的工作室之间的合作变得更加容易。网页应用管理，应用设计，应用虚拟主机，存储，安全以及应用开发协作工具等。

作用

通过PaaS这种模式，用户可以在一个包括SDK，文档和测试环境等在内的开发平台上非常方便地编写应用，而且不论是在部署，或者在运行的时候，用户都无需为服务器，操作系统，网络和存储等资源的管理操心，这些繁琐的工作都由PaaS供应商负责处理，而且PaaS在整合率上面非常惊人，比如一台运行Google App
Engine的服务器能够支撑成千上万的应用，也就是说，PaaS是非常经济的。PaaS主要的用户是开发人员。

产品

一些大的PaaS提供者有Google App Engine，Microsoft Azure，Force.com，Heroku，Engine
Yard。新兴的公司有AppFog，Mendix和Standing Cloud

功能

为了支撑着整个PaaS平台的运行，供应商需要提供那么功能?主要有四大功能：

友好的开发环境：通过提供SDK和IDE等工具来让用户能在本地方便地进行应用的开发和测试。

丰富的服务：PaaS平台会以API的形式将各种各样的服务提供给上层的应用。

自动的资源调度：也就是可伸缩这个特性，它将不仅能优化系统资源，而且能自动调整资源来帮助运行于其上的应用更好地应对突发流量。

精细的管理和监控：通过PaaS能够提供应用层的管理和监控，比如，能够观察应用运行的情况和具体数值(比如，吞吐量和反映时间)来更好地衡量应用的运行状态，还有能够通过精确计量应用使用所消耗的资源来更好地计费。

SaaS（Software-as-a-Service，软件即服务）

第三层也就是所谓SaaS。这一层是和你的生活每天接触的一层，大多是通过网页浏览器来接入。任何一个远程服务器上的应用都可以通过网络来运行，就是SaaS了。

你消费的服务完全是从网页如Netflix, MOG, Google Apps, Box.NET,
Dropbox或者苹果的iCloud那里进入这些分类。尽管这些网页服务是用作商务和娱乐或者两者都有，但这也算是云技术的一部分。

一些用作商务的SaaS应用包括Citrix的GoToMeeting，Cisco的WebEx，Salesforce的CRM，ADP，Workday和SuccessFactors。

作用

通过SaaS这种模式，用户只要接上网络，并通过浏览器，就能直接使用在云端上运行的应用，而不需要顾虑类似安装等琐事，并且免去初期高昂的软硬件投入。SaaS主要面对的是普通的用户。

产品

主要产品包括：Salesforce Sales Cloud，Google Apps，Zimbra，Zoho和IBM Lotus Live等。

功能

谈到SaaS的功能，也可以认为是要实现SaaS服务，供应商需要完成那些功能?主要有四个方面：

随时随地访问：在任何时候或者任何地点，只要接上网络，用户就能访问这个SaaS服务。

支持公开协议：通过支持公开协议(比如HTML4/5)，能够方便用户使用。

安全保障：SaaS供应商需要提供一定的安全机制，不仅要使存储在云端的用户数据处于绝对安全的境地，而且也要在客户端实施一定的安全机制(比如HTTPS)来保护用户。

多租户(Multi-Tenant)机制：通过多住户机制，不仅能更经济地支撑庞大的用户规模，而且能提供一定的可定制性以满足用户的特殊需求。

BaaS（Backend as a Service，后端即服务）

BaaS 是一种新型的云服务，旨在为移动和 Web
应用提供后端云服务，包括云端数据/文件存储、账户管理、消息推送、社交媒体整合等。BaaS
是垂直领域的云服务，随着移动互联网的持续火热，BaaS
也受到越来越多的开发者的亲睐。它作为应用开发的新模型，可以降低开发者成本，让开发者只需专注于具体的开发工作。

BaaS是移动中间件的替代品（或者说备选方案），它使用统一的API和SDK来连接移动应用到后端云存储，传统的移动中间件通过本地的物理服务把后端服务集成到应用中。而BaaS通过云来集成后端服务。中间件和BaaS的最大不同是它们是否包含或者提供云的服务，BaaS可以说是PaaS平台在移动垂直领域的延伸，更可以说是移动中间件和云的融合。而现在它们都在以不同的形式来存在，云的优势很明显，那就是简单、成本低廉，中间件的优势是数据安全、易于扩展。所以从现在的趋势来看，它们不存在明显的取代关系，只不过可能以后BaaS的体量会更大。移动中间件将更多的被有能力的企业使用，同时也会有越来越多的中小型企业、开发者选择使用BaaS。

虽然BaaS属于PaaS的范畴，但两者也有区别。Quora上有人简要描述了二者的不同，BaaS简化了应用开发流程，而PaaS简化了应用部署流程。PaaS是一个执行代码以及管理应用运行环境的开发平台，用户通过SVN或者Git之类的代码版本管理工具与平台交互，对于开发者来说，PaaS就像是一个容器，输入是代码和配置文件，输出是一个可访问应用的URL。而BaaS平台进一步将用户需求进行了抽象，比如用户管理，开发者希望创建用户数据库表（模型）后，客户端就可以通过Restful接口直接操作对应的模型，所有的操作都可以被抽象为CRUD。之前，开发者需要创建表、写接口、写校验，而在BaaS平台中，开发者只需要定义模型，平台就会自动生成对应的接口，这可以让开发者更加专注具体的客户端代码。专门针对手机端的BaaS服务称为MBaaS，目前大多的BaaS平台都属于这一类。

随着移动互联网的发展，移动行业的分工也会像其它行业一样逐渐细化，后端服务就是这样被抽象出来，它统一向开发者提供文件存储、数据存储、推送服务等实现难度较高的功能，以帮助开发者快速开发移动应用。在国外，BaaS服务已经受到巨头的重视，2013年4月，Facebook收购Parse；2014年6月，苹果发布了CloudKit；2014年10月，Google收购了Firebase。Parse、CloudKit、Filrebase都是国外知名的BaaS类产品，苹果和谷歌通过BaaS服务可以更好的完善其生态圈，Parse也可以帮助Facebook建立它在移动端的地位，从巨头们在BaaS方面的布局也可以看出BaaS的价值。总体来说，BaaS平台的优势包括（来自搜狗百科）：

提高效率：减少移动APP开发中各个环节的成本，提高效率。

缩短上市时间：减少从构思到制作过程中的阻碍，并降低上线后的运营成本。

减少交付APP所需的资源：BaaS需要的开发者和IT资源更少。

针对手机和平板优化：BaaS供应商在优化移动APP数据和网络上花费了大量时间和资源，减少了跨平台和移动终端的碎片化的问题。

安全和弹性的基础设施：BaaS提供捆绑的基础设施，解决了弹性、安全性和性能等运营难题，让开发者专注开发。

大量的常用API资源：BaaS将常用和必要的第三方API资源汇总，省去开发者单独收集的麻烦。

它们主要提供的服务包括：

数据存储。用户可以通过可视化的界面设计数据库，包括创建Class、定义字段、录入数据等。同时，BaaS平台可以自动生成对应的Restful API，用户可以通过任何语言操作已有的API，另外，平台也内置用户系统、角色系统、文件系统、权限控制等模块。

数据推送。结合APP中的标签设置，针对不同属性的用户推送差异化信息，包括定时推送、离线推送等。

版本管理。支持iOS及Android版本的同步或异步管理，在控制台内流程化进行开发和版本管理。支持增量更新，终端用户可在应用内进行更新。

数据统计。平台可以查看应用的新增用户以及活跃用户数据，并支持自定义事件统计。

区块链

Crypto Currency

区块链最为人熟知的应用即密码学货币，也被经常称为虚拟货币。然而，密码学货币并不等同于数字货币（Digital Currency），也不等同于虚拟货币（Virtual Currency）。密码学货币指依靠密码技术和校验技术来创建，分发和维持的数字货币，如比特币（BTC）、莱特货币（LTC）等。

2009年出现的比特币是第一个去中心化的密码学货币。之后，大量的密码学货币涌现，这些密码学货币通常被称为山寨币（altcoins）。多数密码学货币都设计成通缩的形态，即货币总量的增加速度会逐渐变慢，最终会到达一个固定值，类似贵金属的产出。这类密码学货币相对于中心化的数字货币有一个共同点即去中心化。这类去中心化的实现都是依托于区块链技术种的去中心化属性。和法定货币相比，密码学货币没有一个集中的发行方，而是由网络节点的计算生成，谁都有可能参与制造密码学货币，而且可以全世界流通，可以在任意一台接入互联网的电脑上交易，不管身处何方，任何人都可以挖掘、收取或转出密码学货币，并且在交易过程中接收方仅需依靠密码学算法来确认交易有效性，无需辨认用户身份信息或发送途径等。

密码学货币的特点有：

- 去中心化：多数密码学货币是一种分布式的虚拟货币，整个网络由用户构成，没有中央银行。去中心化是多数密码学货币安全与自由的保证。
- 全世界流通：密码学货币可以在任意一台接入互联网的电脑或手机上管理，前提是你有证明所有权的私钥。
- 专属所有权：密码学货币依靠私钥确认所有权，它可以被隔离保存在任何存储介质。除了用户自己之外无人可以获取。

但密码学货币也有自己的弊端，比如：

1. 51%攻击，由于密码学货币的去中心化属性，如果有人控制了整个网络上的51%的算力，篡改一段时间之前交易的几率就会变高。
2. 多数密码学货币的交易确认时间较长，交易时，为了确认数据准确性，需要和p2p网络进行交互，得到全网确认后，交易才算完成。此外初次启动时，也需要消耗大量时间下载历史交易数据。

DLT

密码学货币的火热高速推动了区块链技术的发展，金融机构越来越多的关注到了这个领域。随着金融机构的关注度上升，密码学货币中的去中心化属性也成了热门话题，公有链、私有链、联盟链的划分也逐渐展露。此时，分布式账本技术（Distributed Ledger Technology，DLT）这个名词也应运而生。DLT的诞生主要是为了区别于区块链概念，由于密码学货币的去中心化是最大的亮点，其背后的区块链技术也通常被认为自带去中心化属性，金融机构为了移除这种去中心化的属性，就非常需要DLT这个概念。

目前，DLT可以说是区块链技术应用在金融机构中最重要的形态，该技术可以移除当前市场基础设施中的效率极低和成本高昂的部分。

DLT从实质上说就是一个可以在多个站点、不同地理位置或者多个机构组成的网络里进行分享的资产数据库。在一个网络里的参与者可以获得一个唯一、真实账本的副本。账本里的任何改动都会在所有的副本中被反映出来，反应时间会在几分钟甚至是几秒内。在这个账本里存储的资产可以是金融、法律定义上的、实体的或是电子的资产。在这个账本里存储的资产的安全性和准确性是通过公私钥以及签名的使用去控制账本的访问权，从而实现密码学基础上的维护。根据网络中达成共识的规则，账本中的记录可以由一个、一些或者是所有参与者共同进行更新。

DLT的这些特点都使得其有可能称为全新的基础架构模式，因此，很有潜力帮助政府征税、发放福利、发行护照、登记土地所有权、保证货物供应链的运行，并从整体上确保政府记录和服务的正确性。例如在英国国民健康保险制度（NHS）里，这项技术通过改善和验证服务的送达以及根据精确的规则去安全地分享记录，有潜力改善医疗保健系统。对这些服务的消费者来说，这项技术根据不同的情况，有潜力让消费者们去控制个人记录的访问权并知悉其他机构对其记录的访问情况。

Smart Contract

在区块链技术发展的初期，密码学货币或者DLT中的数据变化都是基于明确的几种逻辑。显然，这种模式大家是不会满足的。因此，很快就有人将智能合约和区块链技术进行了结合。智能合约是能够自动执行合约条款的计算机程序。未来的某一天，这些程序可能取代处理某些特定金融交易的律师和银行。智能合约的潜能不只是简单的转移资金，一辆汽车或者一所房屋的门锁，都能够被连接到物联网上的智能合约被打开。

智能合约其实在区块链之前就有，但是偏概念，直到区块链到来后才开始有了实用价值的应用，特别是金融和物联网领域。智能合约的理念可以追溯到1994年，几乎与互联网（world wide web）同时出现。为比特币打下基础而受到广泛赞誉的密码学家尼克萨博（Nick Szabo）首次提出了“智能合约”这一术语。从本质上讲，这些自动合约的工作原理类似于其它计算机程序的if-then语句。智能合约只是以这种方式与真实世界的资产进行交互。当一个预先编好的条件被触发时，智能合约执行相应的合同条款。

区块链技术的出现和被广泛使用，则改变阻碍智能合约实现的现状，从而萨博的理念有了重生的机会。让我们举一个简单的例子，以足球比赛为例。假如你赌A队赢，你的朋友赌B队赢。此时，你和你的朋友可以将赌注写成一个区块链上的智能合约。当比赛结束时，智能合约能够通过媒体确认比赛结果然后自动结算，此时你们不需要依赖任何第三方机构。这就是智能合约的一个例子。

智能合约的最大特点就是代码的执行是自动的：要么成功执行，或者所有的状态变化都撤消（包括从当前失败的合约中已经送或接收的信息。）这是很重要的，因为它避免了合约部分执行的情况（例如，在证券购买交易中，证券所有者已经转移发送了证券，但是密码学货币的支付转移却失败了）。在区块链环境中，这尤为重要，因为没有办法来撤消执行错误所带来的不好的后果（而且如果对手不配合的话，根本就没有办法逆转交易）。

智能合约理论上可以给所有使用合约的场景带来变革。例如，证券的登记和清算无需再通过证券交易所，购买保险和理赔无需通过保险公司，投融资无需通过代理公司等等。