**一、MVC**

MVC（Model-View-Controller）的特点包括：

1.分层结构：MVC将应用程序分为三个主要部分，即模型、视图和控制器，每个部分都按照不同的目的和职责进行设计和实现。

2.模块化设计：MVC采用模块化的设计，让不同的模块分工协作，避免了代码冗余和模块之间的紧密耦合。

3.易于维护：MVC使得应用程序的各个部分的职责划分明确，理清了各自的职能，使得应用程序易于维护。

4.高扩展性：MVC的设计使得应用程序的各个部分可以独立扩展和修改，降低了修改或扩展应用程序时的风险和代价。

5.代码重用性：由于MVC的代码架构清晰、逻辑独立，所以提高了代码的重用性。

6.可移植性：MVC的设计使得应用程序的各个部分可以在不同的平台上运行和实现，提高了应用程序的可移植性。

**二、存算分离MVC**

存算分离MVC相较于普通MVC，主要强调数据（存储）和业务逻辑（算法）的分离，所以它的特点包括：

1.存算分离：将数据和算法分离，使得数据处理和业务逻辑的修改可以独立进行，提高了代码复用性和可维护性。

2.模块化设计：存算分离MVC同样采用模块化的设计，让不同的模块分工协作，避免了代码冗余和模块之间的紧密耦合。

3.高效性：由于数据和算法的分离，存算分离MVC能够更快的处理数据和计算，提高了应用程序的性能。

4.易于测试：存算分离MVC的数据和算法部分都可以独立测试，这使得开发人员更容易发现并修复错误，提高了开发效率。

5.可维护性：存算分离MVC的设计使得算法的修改不会影响数据存储，数据存储的修改不会影响算法，使得应用程序易于维护。

6.增加可扩展性：存算分离MVC对于数据和业务逻辑的分离，增加了扩展的可能性，新的数据存储方案和算法可以很容易地被添加到应用程序中。

**三、P2P**

点对点网络（P2P Network）作为区块链的底层技术之一，其特点包括：

1.分布式：点对点网络是一种去中心化的网络结构，在网络中各个节点之间没有主从关系，它们通过协议和算法相互通信和协作。

2.可扩展性：P2P网络的节点可以动态加入和退出，这使得它天生适应于分布式系统中的数据和服务的扩展和缩减。

3.高可用性：由于P2P网络的节点没有中心化的控制节点，因此它的鲁棒性和可靠性非常高，即使有一些节点因故障或攻击而失效，整个网络也可继续稳定运行。

4.安全性：P2P网络使用加密算法和数字签名等技术，来确保通信的机密性、完整性和认证性，保证了信息的安全性。

5.低成本：P2P网络不需要在节点之间建立复杂的信任体系或高级的网络设备，节点之间通信和协作的成本较低，因此非常适合开展小规模和中规模的数据和服务共享。

6.无需第三方信任：P2P网络中的节点相互平等，没有中心化的信任机构，这使得它无需借助信任第三方就可以实现可靠的数据和服务传输和共享。这种特点也是区块链的重要特点之一。

**四、微服务**

微服务是一种将应用程序拆分成小型、自治的服务的体系结构风格。每个服务都是独立部署、独立运行和独立维护的，它们可以使用不同的编程语言和技术栈。这种架构风格的优点是可以提高应用程序的可伸缩性、可靠性和灵活性。

**五、管道过滤器**

管道过滤器式风格是一种将应用程序分解成一系列处理单元的体系结构风格。每个处理单元执行特定的任务，它们按照一定的顺序连接起来，形成处理管道。每个处理单元将输入转换为输出，并将其传递给下一个处理单元。这种架构风格的优点是可以提高应用程序的可维护性、可重用性和可扩展性。

微服务和管道过滤器是当前软件体系结构中较为流行的两种风格。

这两种架构风格通常可以结合使用，以创建更加灵活、可靠和高效的应用程序。例如，微服务可以使用管道过滤器式风格来实现服务间的通信和数据传输。