## 链码概念

Chaincode：链上代码，简称链码，一般是指由开发人员使用Go语言（也支持Java等语言）编写的应用程序代码，提供分布式账本的状态处理逻辑。链码被部署在Fabric的网络节点中，能够独立运行在具有安全特性的受保护的 Docker 容器中，以 gRPC 协议与相应的 peer 节点进行通信，以操作（初始化或管理）分布式账本中的数据。可以根据不同的需求开发出不同的复杂的应用。

## 系统链码

负责 Fabric 节点自身的处理逻辑, 包括系统配置、背书、校验等工作

系统链码仅支持 Go 语言, 在 Peer 节点启动时会自动完成注册和部署

**系统链码共有五种类型：**

**1 配置系统链码(CSCC)**

CSCC：Configuration System Chaincode，负责处理 Peer 端的 Channel 配置。

**2 生命周期系统链码(LSCC)**

LSCC：Lifecycle System Chaincode，负责对用户链码的生命周期进行管理。

**3 查询系统链码(QSCC)**

QSCC：Query System Chaincode，提供账本查询 API。如获取区块和交易等信息。

**4 背书管理系统链码(ESCC)**

ESCC：Endorsement System Chaincode，负责背书(签名)过程, 并可以支持对背书策略进行管理对提交的交易提案的模拟运行结果进行签名,，之后创建响应消息返回给客户端。

**5 验证系统链码(VSCC)**

VSCC：Validation System Chaincode，处理交易的验证，包括检查背书策略以及多版本并发控制。

## 用户链码

由应用程序开发人员根据不同场景需求及成员制定的相关规则，使用 Golang（或Java等）语言编写的基于操作区块链分布式账本的状态的业务处理逻辑代码，运行在链码容器中，通过 Fabric 提供的接口与账本状态进行交互。

用户链码在整个应用程序中处于重要地位。因为它下可对账本数据进行操作，上可以给企业级应用程序提供调用接口。所以一个没有链码的企业级应用程序，不能称之为是基于区块链的企业级应用程序。

## 链码生命周期

这一系列的操作是由链码的生命周期来负责管理。

管理 Chaincode 的生命周期共有五个命令：

* install：将已编写完成的链码安装在网络节点中。
* instantiate：对已安装的链码进行实例化。
* upgrade：对已有链码进行升级。链代码可以在安装后根据具体需求的变化进行升级。
* package：对指定的链码进行打包的操作。
* singnpackage：签名。

安装、实例化、升级这三项操作不适用于系统链码