Cita-Gossip 各模块数据结构及接口 1.0
2018/8/7

对外接口为 Gossip 模块接口：

Spread：接收并散布 cita 想要发布的消息（账本）

Hear：听闻收到的消息（账本）

---

pub mod discovery;　　[外接 mDNS 库]
//提供配置文件的接口，用以发现新节点（两种方式）
接口：发现新节点

pub struct *NetworkMember{}* //标识 Id ， 性质 Properties
方法：
pub fn *ApplyBySeed()*　　//从配置文件中获取 seed 节点
pub fn *Listen_mDNS()*　　//监听 mDNS
pub fn *send_mDNS()*　　//新节点上线激活 mDNS
pub fn *send()*　　//传入 filter 激活相应操作

pub mod gossip;
//综合调用以下模块。
接口：接收并散布 cita 想要发布的消息（账本）、听闻收到的消息（账本）

pub fn *Spread(){}*
//接收并散布 cita 想要发布的消息（账本）
pub fn *Hear(){}*
//听闻接收到的消息（账本）
pub fn *Gossip(){*
//控制 Gossip 过程
//例如周期性发送 Alive 消息，处理各种消息等等
pub fn send_recurrent(){}　//周期性发送
pub fn send_unicast(){}　　//单播支持
pub fn send_gossip(){}　　//随机发送支持
pub fn get_discovery(){}　//discovery 模块传入
pub fn decap(){}　　//对 comm 模块进来的消息解封装，调用 msg 模块中的相应消息类型进行操作

}

pub mod comm;　[外接 gRPC 库]
//底层通信模块，调用通信机制，进行包括：建立连接、发送信息、中断连接、连接确认等功能，具体参考 fabric，使用 gRPC 方式实现通信。
接口：将本地产生的消息发送、接收消息并交由本地处理、调用 gRPC

pub mod SSL; [外接库]
//提供加密传输服务和验证信息服务，是白名单的体现（有白名单的能正确加密和解密信息）
pub mod identity;//暂时包括 CA 模块管理证书
//保留节点 id 到公钥证书的映射

pub mod msg;
//数据结构中声明消息类型，封装逻辑操作
接口：各消息的发送、接收函数

pub struct *Empty_msg{}*
源节点 src ，目标节点 des
pub struct *List_msg{}*
源节点 src ，目标节点 des ， 消息内容 list
包含 *Alive_msg*、*Dead_msg*、 *ListRequest_msg*、*ListResponse_msg*
pub struct *Gossip_msg{}*
源节点 src ，目标节点 des ， 消息内容 data
pub struct *HeartBeat_msg{}*
源节点 src ，目标节点 des
方法：每种消息各自的发送、接收函数

pub mod storage;
//声明修改本地数据的规则，提供数据接口
接口：修改与获取 list 的函数

pub fn ChangeList(){}
pub fn GetList(){}

pub mod filter;
//调节发送对象，实现随机或单播等功能
接口：filter 函数，实现将逻辑需求转化为具体通信目标的功能

pub fn *filter() -> ip{*
//需要从 id 列表中随机选择一个 id（本机除外）并转换为 ip 发送
}
pub fn *nonrandom_filter((target:发送目标) -> ip{}*

pub mod service；
//对外提供同步账本用的函数接口，实现对账本的随机发送
接口：
pub struct buf{} //暂存要发送的账本
方法：
pub fn gossip(){} //调用 filter，读取 storage，随机发送