## Producer\_plugin

控制各种黑白名单

注册多个timer，用其控制该什么时候检查是否要出块，并控制chain\_plugin出块。

管理priv-key

处理incoming\_block, incoming\_transaction

处理chain.accepted\_block， chain.irreversible\_block

## 主要数据结构

std::set<chain::account\_name> \_producers，如果要出块的话，出块producer们的名字

\_production\_enabled，是否允许出块

## 主要流程

在initialize中做各种初始化，在start\_up中调用schedule\_production\_loop开始工作：

调用start\_block，针对结果，可能会等待某个时间后重试，或者调用maybe\_produce\_block来生产区块。

注意这里的等待某个时间后重试，是共识的一部分，因为producer是按照固定顺序出块的，这里的判断逻辑还挺复杂。

在start\_block里，会调用 chain.abort\_block和chain.start\_block来调整chain\_plugin的block状态，并把chain.get\_unapplied\_transactions取出来push\_transaction，以及scheduled\_trxs，\_pending\_incoming\_transactions安排执行。逻辑挺复杂。

maybe\_produce\_block->produce\_block

-> chain.finalize\_block();

chain.sign\_block

chain.commit\_block();

->emit( self.accepted\_block, pending->\_pending\_block\_state );

->net\_plugin. accepted\_block

->dispatcher->bcast\_block，会检查所有connection里，是否需要向peer发送该block，如果需要，就发送过去。