# Slotleader选取流程

## 概述

Slot Leader选取主要用于选出当前epoch每个Slot所对应的唯一出块人。Slot Leader选取实际上选出的是一个顺序，用此顺序匹配上一个epoch生成的为本区块出块使用的epoch leaders，决定本个epoch内每个slot的出块人。

Slot Leader需要匿名出块，而出的块需要被所有人可以验证。

## 时间划分

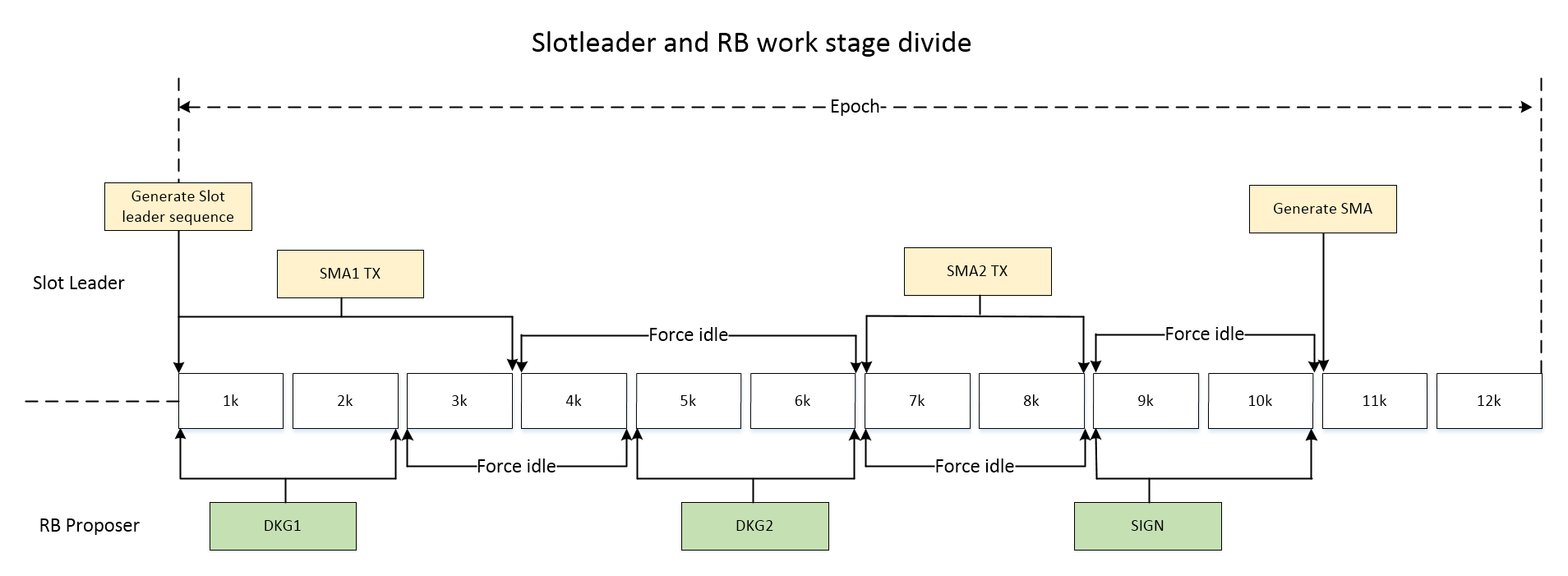
Slotleader选取以1个epoch为基准执行周期，每个epoch内行为相同。

将每个epoch中的slot划分成10份，每份k个slot。

选取周期设计为1k发送Stage1交易，2k强制空闲，3k~4k发送Stage2交易，5k强制空闲，6k计算SMA。此SMA值保存在本地的LevelDB中，供下个epoch计算slotLeader使用。因为RB计算随机数在7k时进行，所以slot leader选取生成SMA的过程要在7k前结束，以保证结果不可更改。

在强制空闲时间范围内到达预编译合约的交易，信息不写入StateDB。例如只有3k~4k过程中收到的Stage2交易允许写入数据到StateDB中，其它时间收到的Stage2交易不写入StateDB。

如下图所示：



主要步骤包括：

1、使用上个Epoch生成的Epoch Leader作为基础匹配slotLeader顺序，计算当前epoch每个slot的出块人；

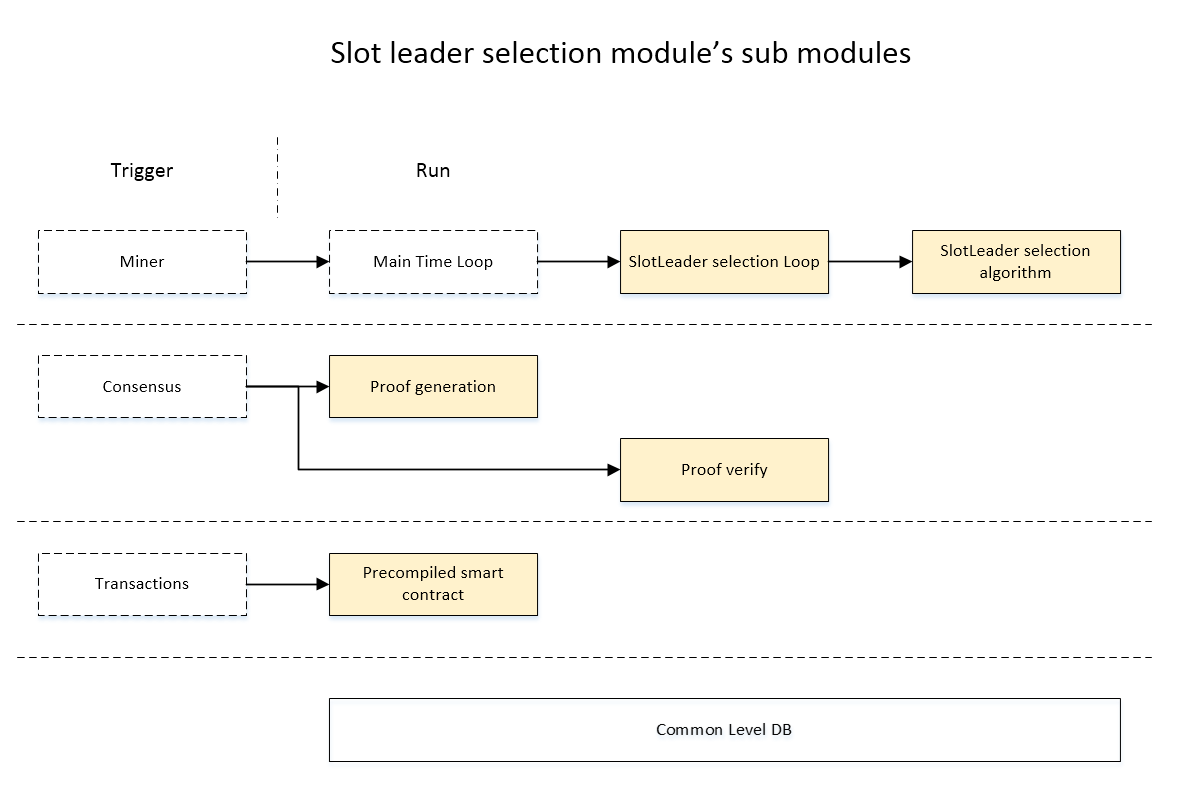
2、epoch Leader选取模块（外部）生成当前epoch的epoch Leader，用于交易发送和下个epoch出块；

3、当前epoch的epoch Leader 发送交易；

4、在生成RB随机数开始前，计算SMA用于下个Epoch的slot出块选择；

## 代码结构

slotLeader功能代码主要由如下几个部分组成：工作循环、选取算法、预编译合约、Proof生成、proof验证及POS内部共用的LevelDB封装。



所对应的代码分别在如下目录下：

|  |
| --- |
| pos/slotleader/\*.go  core/vm/slot\_leader\_selection\_contract.go  pos/posdb/\*.go  pos/postools/slottools/\*.go |

## 工作循环

工作循环周期性的被Main Time loop 触发运行。在运行过程中，获取当前的epochID和slotID，针对每个epochID，根据slotID的不同，划分了4个执行阶段。分别为：

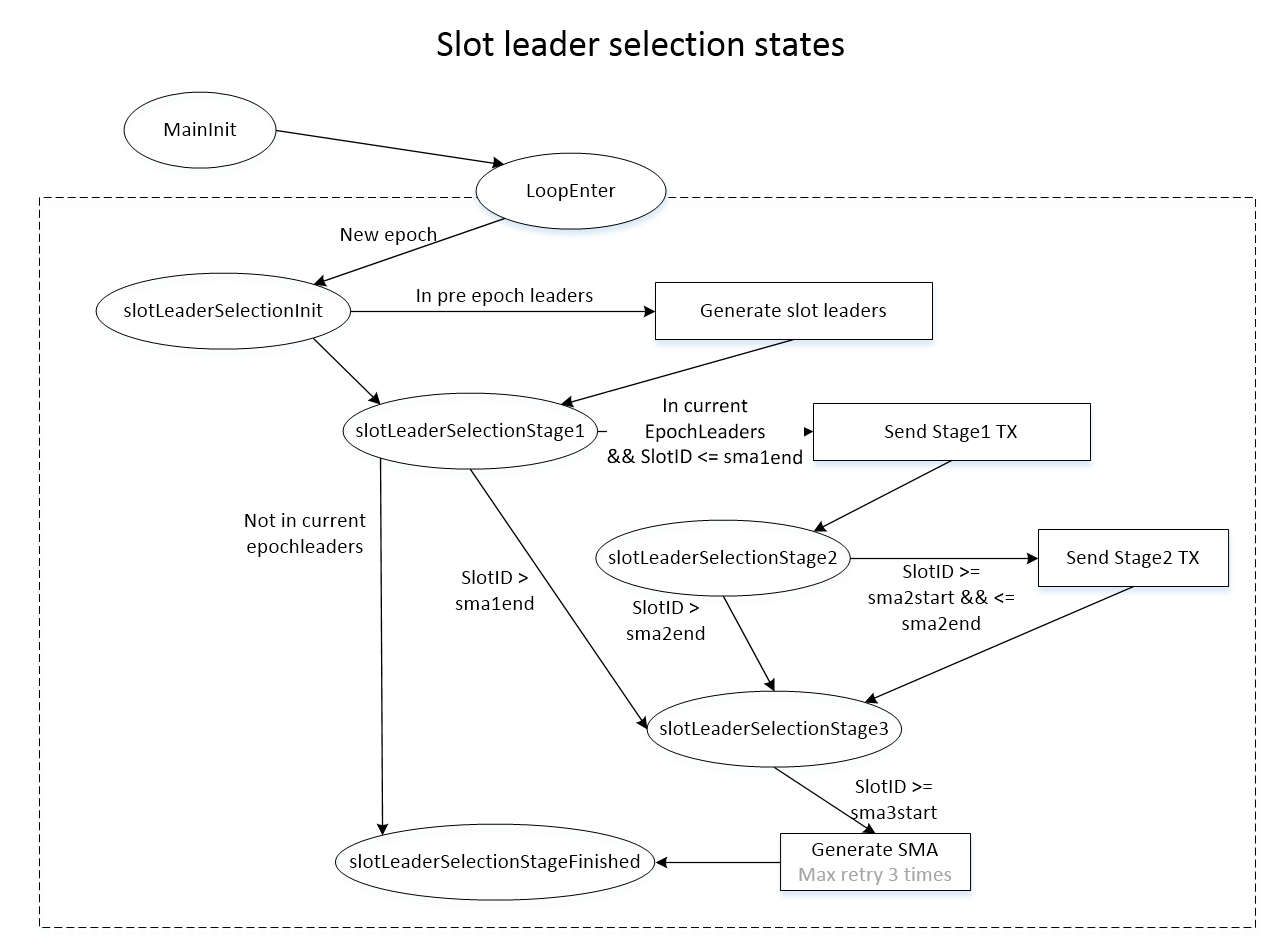
Stage 1: slot leader selection initial;

Stage 2: generate slot leader; Send Stage1 TX;

Stage 3: Send Stage2 TX;

Stage 4: Compute and Generate SMA;

Stage 5: Complete, wait for the next epoch.



所对应的代码在如下位置：

|  |
| --- |
| pos/slotleader/workflow.go |

## 选取算法（赵晓峰）

选取算法主要分为：Stage1，Stage2，SMA生成三个部分。

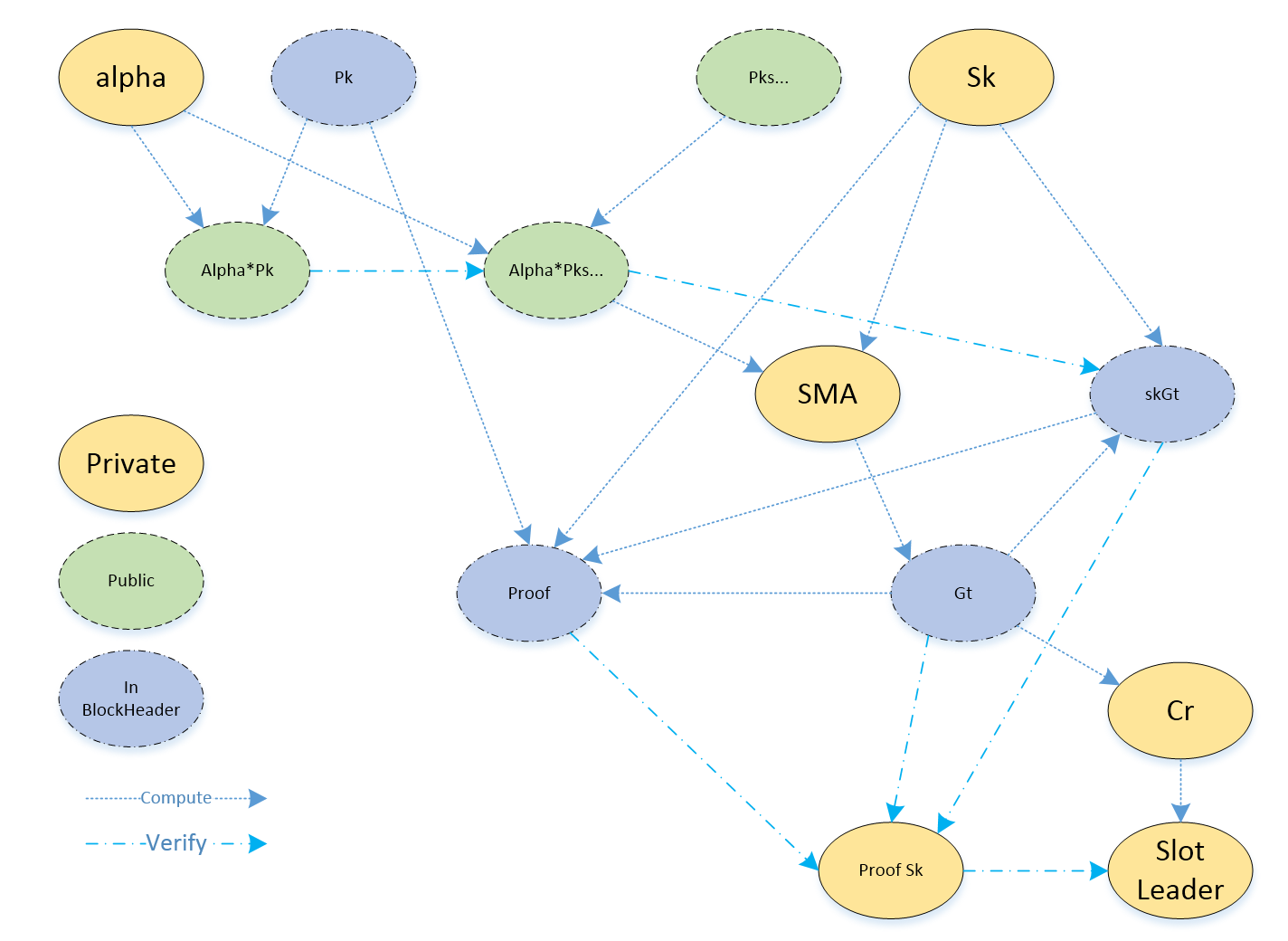
Stage1：每个在当前epoch选出，并将在下一个epoch出块的参与人，生成随机数α，并在合适的时机发送α\*Pk到链上；其中epoch 0的α值为公钥的hash。

Stage2：（赵晓峰）

SMA生成：（赵晓峰）

## 预编译合约

## Proof生成（赵晓峰）



## Proof验证（赵晓峰）

## 共用LevelDB封装