

External client support for Zebra specification

The specs simulates a call to the *create_account* grpc method as a starting point and then the grpc method calls the *create_account* procedure in the *zcash_client_backend* side. The grpc method then sends the key to the memory wallet and the memory wallet adds the key to the accounts set. The memory wallet then sends a block to the memory wallet and the memory wallet adds the block to the blocks set.

The memory wallet is a simple algorithm that listens for requests and sends adding requests to the scan task. The scan task listens for requests from the services process and adds tasks to the scan task set. The scan task also adds account to the memory wallet and either sends “scanned” blocks to the memory wallet or does nothing more.

The main process is the entry point of the model and calls the *create_account* grpc method.

EXTENDS *TLC*, *Integers*, *Sequences*, *Json*, *FiniteSets*

StatusWaiting \triangleq “waiting”

StatusAdding \triangleq “adding”

CreateAccountServiceRequest \triangleq “create_account”

--algorithm *client_integration*

variables

A string that will be used as a response to any of the *gRPC* method calls, initially empty.

response = “” ;

The current service request flag, initially listening for requests.

service_request = *StatusWaiting* ;

The current status of the scan task, initially listening for requests.

scan_task_status = *StatusWaiting* ;

The set of scan tasks that are currently being processed, initially empty.

scan_tasks = {} ;

The key that will be served to the client after a create account request.

key_to_be_served = “” ;

The block that will be served to the client after a scan task finds a relevant block, initially empty.

block_to_be_served = [*height* \mapsto 0, *hash* \mapsto “000000”] ;

The set of accounts that in the memory wallet, initially empty.

accounts = {} ;

The set of blocks in the memory wallet, initially empty.

blocks = {} ;

Keep track of the last inserted account id.

last_account_id = 0 ;

define

Ensure that whenever a block is available, it eventually gets inserted into the memory wallet.

LIVENESS_BLOCK_INSERTION \triangleq

\wedge *block_to_be_served.height* > 0

$\Rightarrow \Diamond(\forall b \in \text{blocks} : b = \text{block_to_be_served})$

Ensure that an account is not added twice.

$Safety_Account_Addition \triangleq$
 $\wedge \forall a \in accounts :$
 $\wedge a.account_id \geq 0$
 $\wedge \forall b \in accounts : b.account_id \neq a.account_id$
 Ensure that the account id is incremented properly.
 $Safety_Account_ID_Increment \triangleq$
 $\wedge \forall a, b \in accounts : a.account_id < b.account_id$
 Ensure that a block is not inserted multiple times.
 $Safety_Block_Insertion \triangleq$
 $\wedge \forall b \in blocks :$
 $\wedge b.height > 0$
 $\wedge \forall c \in blocks : c.height \neq b.height$
 Ensure that the service request always return to listening after adding.
 $Service_Request_Transition \triangleq$
 $\wedge service_request = StatusAdding$
 $\Rightarrow \Diamond(service_request = StatusWaiting)$
 Ensure that all accounts have a non empty *ufvk* and that blocks have non zero hash.
 $Inductive_Invariant \triangleq$
 $\wedge (\forall acc \in accounts : acc.ufvk \neq "")$
 $\wedge (\forall blk \in blocks : blk.hash \neq "000000")$
 $\wedge service_request \in \{StatusWaiting, StatusAdding, CreateAccountServiceRequest\}$
end define ;

UTILITY PROCEDURES:

– Procedure to initiate the *gRPC* call for account creation. This sets the *service_request* flag to signal that an account creation request has been received and is being processed.

```

procedure create_account_grpc()
begin
    CreateAccountGrpc:
        service_request := CreateAccountServiceRequest ;
end procedure ;

```

The *create_account* in the *zcash_client_backend* side.

```

procedure create_account_zcash_client_backend()
begin
    CreateAccountZcashClientBackend:
        response := "zxviews..." ;
        return ;
end procedure ;

```

The *put_block* in the *zcash_client_backend* side.

```

procedure put_block_zcash_client_backend()
begin
    PutBlockZcashClientBackend:
        blocks := blocks  $\cup$  {block_to_be_served} ;

```

end procedure ;

SERVICES PROCESS :

Listen for requests and send adding requests to scan task.

process *services* = "SERVICES"

begin

Services:

We only have one service request in this algorithm.

if *service_request* = *CreateAccountServiceRequest* **then**

CallZcashClientBackend:

call *create_account_zcash_client_backend()* ;

SendKey:

key_to_be_served := *response* ;

CreateAccount:

scan_task_status := *StatusAdding* ;

end if ;

ServicesLoop:

goto *Services* ;

end process ;

SCAN TASK PROCESS :

Listen for requests from the services process and :

– *Add tasks to the scan task set.*

– *Add account to the memory wallet.*

– *Either send "scanned" blocks to the memory wallet or do nothing more.*

process *scantask* = "SCAN TASK"

variables *inner_state* = {}, *inner_accounts* = {}, *inner_blocks* = {}, *inner_last_account_id* = 0 ;

begin

GetGlobals:

inner_state := *scan_tasks* ;

inner_accounts := *accounts* ;

inner_last_account_id := *last_account_id* ;

ScanTask:

if *scan_task_status* = *StatusAdding* **then**

AddingAccount:

accounts := *inner_accounts* \cup {*account_id* \mapsto *last_account_id* + 1, *ufvk* \mapsto *key_to_be_served*}

scan_tasks := *inner_state* \cup {*key_to_be_served*} ;

scan_task_status := *StatusWaiting* ;

last_account_id := *inner_last_account_id* + 1 ;

end if ;

SendBlock:

either

block_to_be_served := [*height* \mapsto 1, *hash* \mapsto "111111"] ;

```

        call put_block_zcash_client_backend();
    or
        skip;
    end either ;
    ScanTaskLoop:
        goto ScanTask;
end process ;

MAIN PROCESS :

process Main = "MAIN"
begin
    CreateAccountCall:
        The grpc is the entry point of the model.
        call create_account_grpc();
    End:
        skip;
end process ;

end algorithm ;

BEGIN TRANSLATION(chksum(pcal) = "83f024a5"  $\wedge$  chksum(tla) = "971a54d7")
VARIABLES response, service_request, scan_task_status, scan_tasks,
           key_to_be_served, block_to_be_served, accounts, blocks,
           last_account_id, pc, stack

define statement
LIVENESS_BLOCK_INSERTION  $\triangleq$ 
     $\wedge$  block_to_be_served.height > 0
     $\Rightarrow \Diamond(\forall b \in \text{blocks} : b = \text{block\_to\_be\_served})$ 

SAFETY_ACCOUNT_ADDITION  $\triangleq$ 
     $\wedge \forall a \in \text{accounts} :$ 
         $\wedge a.\text{account\_id} \geq 0$ 
         $\wedge \forall b \in \text{accounts} : b.\text{account\_id} \neq a.\text{account\_id}$ 

SAFETY_ACCOUNT_ID_INCREMENT  $\triangleq$ 
     $\wedge \forall a, b \in \text{accounts} : a.\text{account\_id} < b.\text{account\_id}$ 

SAFETY_BLOCK_INSERTION  $\triangleq$ 
     $\wedge \forall b \in \text{blocks} :$ 
         $\wedge b.\text{height} > 0$ 
         $\wedge \forall c \in \text{blocks} : c.\text{height} \neq b.\text{height}$ 

SERVICE_REQUEST_TRANSITION  $\triangleq$ 
     $\wedge \text{service\_request} = \text{StatusAdding}$ 
     $\Rightarrow \Diamond(\text{service\_request} = \text{StatusWaiting})$ 

INDUCTIVE_INVARIANT  $\triangleq$ 

```

$$\begin{aligned}
& \wedge (\forall acc \in accounts : acc.ufvk \neq "") \\
& \wedge (\forall blk \in blocks : blk.hash \neq "000000") \\
& \wedge service_request \in \{StatusWaiting, StatusAdding, CreateAccountServiceRequest\} \\
\\
& \text{VARIABLES } inner_state, inner_accounts, inner_blocks, inner_last_account_id \\
\\
& vars \triangleq \langle response, service_request, scan_task_status, scan_tasks, \\
& \quad key_to_be_served, block_to_be_served, accounts, blocks, \\
& \quad last_account_id, pc, stack, inner_state, inner_accounts, \\
& \quad inner_blocks, inner_last_account_id \rangle \\
\\
& ProcSet \triangleq \{ "SERVICES" \} \cup \{ "SCAN TASK" \} \cup \{ "MAIN" \} \\
\\
& Init \triangleq \text{Global variables} \\
& \quad \wedge response = "" \\
& \quad \wedge service_request = StatusWaiting \\
& \quad \wedge scan_task_status = StatusWaiting \\
& \quad \wedge scan_tasks = \{ \} \\
& \quad \wedge key_to_be_served = "" \\
& \quad \wedge block_to_be_served = [height \mapsto 0, hash \mapsto "000000"] \\
& \quad \wedge accounts = \{ \} \\
& \quad \wedge blocks = \{ \} \\
& \quad \wedge last_account_id = 0 \\
& \quad \text{Process } scantask \\
& \quad \wedge inner_state = \{ \} \\
& \quad \wedge inner_accounts = \{ \} \\
& \quad \wedge inner_blocks = \{ \} \\
& \quad \wedge inner_last_account_id = 0 \\
& \quad \wedge stack = [self \in ProcSet \mapsto \langle \rangle] \\
& \quad \wedge pc = [self \in ProcSet \mapsto \text{CASE } self = "SERVICES" \rightarrow "Services" \\
& \quad \quad \quad \square self = "SCAN TASK" \rightarrow "GetGlobals" \\
& \quad \quad \quad \square self = "MAIN" \rightarrow "CreateAccountCall"] \\
\\
& CreateAccountGrpc(self) \triangleq \wedge pc[self] = "CreateAccountGrpc" \\
& \quad \wedge service_request' = CreateAccountServiceRequest \\
& \quad \wedge pc' = [pc \text{ EXCEPT } ![self] = "Error"] \\
& \quad \wedge \text{UNCHANGED } \langle response, scan_task_status, \\
& \quad \quad scan_tasks, key_to_be_served, \\
& \quad \quad block_to_be_served, accounts, \\
& \quad \quad blocks, last_account_id, stack, \\
& \quad \quad inner_state, inner_accounts, \\
& \quad \quad inner_blocks, inner_last_account_id \rangle \\
\\
& create_account_grpc(self) \triangleq CreateAccountGrpc(self) \\
\\
& CreateAccountZcashClientBackend(self) \triangleq \wedge pc[self] = "CreateAccountZcashClientBackend" \\
& \quad \wedge response' = "zxviews..."
\end{aligned}$$

$$\begin{aligned}
& \wedge pc' = [pc \text{ EXCEPT } ![self] = \text{Head}(\text{stack}[self]).pc] \\
& \wedge \text{stack}' = [\text{stack} \text{ EXCEPT } ![self] = \text{Tail}(\text{stack}[self])] \\
& \wedge \text{UNCHANGED } \langle \text{service_request}, \\
& \quad \text{scan_task_status}, \\
& \quad \text{scan_tasks}, \\
& \quad \text{key_to_be_served}, \\
& \quad \text{block_to_be_served}, \\
& \quad \text{accounts}, \text{blocks}, \\
& \quad \text{last_account_id}, \\
& \quad \text{inner_state}, \\
& \quad \text{inner_accounts}, \\
& \quad \text{inner_blocks}, \\
& \quad \text{inner_last_account_id} \rangle
\end{aligned}$$

$$\text{create_account_zcash_client_backend}(self) \triangleq \text{CreateAccountZcashClientBackend}(self)$$

$$\begin{aligned}
\text{PutBlockZcashClientBackend}(self) \triangleq & \wedge pc[self] = \text{"PutBlockZcashClientBackend"} \\
& \wedge \text{blocks}' = (\text{blocks} \cup \{\text{block_to_be_served}\}) \\
& \wedge pc' = [pc \text{ EXCEPT } ![self] = \text{"Error"}] \\
& \wedge \text{UNCHANGED } \langle \text{response}, \text{service_request}, \\
& \quad \text{scan_task_status}, \\
& \quad \text{scan_tasks}, \\
& \quad \text{key_to_be_served}, \\
& \quad \text{block_to_be_served}, \\
& \quad \text{accounts}, \text{last_account_id}, \\
& \quad \text{stack}, \text{inner_state}, \\
& \quad \text{inner_accounts}, \\
& \quad \text{inner_blocks}, \\
& \quad \text{inner_last_account_id} \rangle
\end{aligned}$$

$$\text{put_block_zcash_client_backend}(self) \triangleq \text{PutBlockZcashClientBackend}(self)$$

$$\begin{aligned}
\text{Services} \triangleq & \wedge pc[\text{"SERVICES"}] = \text{"Services"} \\
& \wedge \text{IF } \text{service_request} = \text{CreateAccountServiceRequest} \\
& \quad \text{THEN } \wedge pc' = [pc \text{ EXCEPT } ![\text{"SERVICES"}] = \text{"CallZcashClientBackend"}] \\
& \quad \text{ELSE } \wedge pc' = [pc \text{ EXCEPT } ![\text{"SERVICES"}] = \text{"ServicesLoop"}] \\
& \wedge \text{UNCHANGED } \langle \text{response}, \text{service_request}, \text{scan_task_status}, \\
& \quad \text{scan_tasks}, \text{key_to_be_served}, \text{block_to_be_served}, \\
& \quad \text{accounts}, \text{blocks}, \text{last_account_id}, \text{stack}, \\
& \quad \text{inner_state}, \text{inner_accounts}, \text{inner_blocks}, \\
& \quad \text{inner_last_account_id} \rangle
\end{aligned}$$

$$\begin{aligned}
\text{CallZcashClientBackend} \triangleq & \wedge pc[\text{"SERVICES"}] = \text{"CallZcashClientBackend"} \\
& \wedge \text{stack}' = [\text{stack} \text{ EXCEPT } ![\text{"SERVICES"}] = \langle [\text{procedure} \mapsto \text{"create_account_zcash_client_backend"}, \\
& \quad pc \mapsto \text{"SendKey"}] \rangle \\
& \quad \circ \text{stack}[\text{"SERVICES"}]]
\end{aligned}$$

$$\begin{aligned}
& \wedge pc' = [pc \text{ EXCEPT } !["SERVICES"] = \text{"CreateAccountZcashClientBackend"}] \\
& \wedge \text{UNCHANGED } \langle response, service_request, \\
& \quad scan_task_status, scan_tasks, \\
& \quad key_to_be_served, block_to_be_served, \\
& \quad accounts, blocks, last_account_id, \\
& \quad inner_state, inner_accounts, \\
& \quad inner_blocks, inner_last_account_id \rangle \\
\\
SendKey & \triangleq \wedge pc["SERVICES"] = \text{"SendKey"} \\
& \wedge key_to_be_served' = response \\
& \wedge pc' = [pc \text{ EXCEPT } !["SERVICES"] = \text{"CreateAccount"}] \\
& \wedge \text{UNCHANGED } \langle response, service_request, scan_task_status, \\
& \quad scan_tasks, block_to_be_served, accounts, blocks, \\
& \quad last_account_id, stack, inner_state, inner_accounts, \\
& \quad inner_blocks, inner_last_account_id \rangle \\
\\
CreateAccount & \triangleq \wedge pc["SERVICES"] = \text{"CreateAccount"} \\
& \wedge scan_task_status' = StatusAdding \\
& \wedge pc' = [pc \text{ EXCEPT } !["SERVICES"] = \text{"ServicesLoop"}] \\
& \wedge \text{UNCHANGED } \langle response, service_request, scan_tasks, \\
& \quad key_to_be_served, block_to_be_served, \\
& \quad accounts, blocks, last_account_id, stack, \\
& \quad inner_state, inner_accounts, inner_blocks, \\
& \quad inner_last_account_id \rangle \\
\\
ServicesLoop & \triangleq \wedge pc["SERVICES"] = \text{"ServicesLoop"} \\
& \wedge pc' = [pc \text{ EXCEPT } !["SERVICES"] = \text{"Services"}] \\
& \wedge \text{UNCHANGED } \langle response, service_request, scan_task_status, \\
& \quad scan_tasks, key_to_be_served, \\
& \quad block_to_be_served, accounts, blocks, \\
& \quad last_account_id, stack, inner_state, \\
& \quad inner_accounts, inner_blocks, \\
& \quad inner_last_account_id \rangle \\
\\
services & \triangleq Services \vee CallZcashClientBackend \vee SendKey \vee CreateAccount \\
& \quad \vee ServicesLoop \\
\\
GetGlobals & \triangleq \wedge pc["SCAN TASK"] = \text{"GetGlobals"} \\
& \wedge inner_state' = scan_tasks \\
& \wedge inner_accounts' = accounts \\
& \wedge inner_last_account_id' = last_account_id \\
& \wedge pc' = [pc \text{ EXCEPT } !["SCAN TASK"] = \text{"ScanTask"}] \\
& \wedge \text{UNCHANGED } \langle response, service_request, scan_task_status, \\
& \quad scan_tasks, key_to_be_served, block_to_be_served, \\
& \quad accounts, blocks, last_account_id, stack, \\
& \quad inner_blocks \rangle
\end{aligned}$$

$$\begin{aligned}
\text{ScanTask} &\triangleq \wedge pc[\text{"SCAN TASK"}] = \text{"ScanTask"} \\
&\wedge \text{IF } scan_task_status = StatusAdding \\
&\quad \text{THEN } \wedge pc' = [pc \text{ EXCEPT } ![\text{"SCAN TASK"}] = \text{"AddingAccount"}] \\
&\quad \text{ELSE } \wedge pc' = [pc \text{ EXCEPT } ![\text{"SCAN TASK"}] = \text{"SendBlock"}] \\
&\wedge \text{UNCHANGED } \langle response, service_request, scan_task_status, \\
&\quad scan_tasks, key_to_be_served, block_to_be_served, \\
&\quad accounts, blocks, last_account_id, stack, \\
&\quad inner_state, inner_accounts, inner_blocks, \\
&\quad inner_last_account_id \rangle \\
\\
\text{AddingAccount} &\triangleq \wedge pc[\text{"SCAN TASK"}] = \text{"AddingAccount"} \\
&\wedge accounts' = (inner_accounts \cup \{[account_id \mapsto last_account_id + 1, ufvk \mapsto key_to_be_served]\}) \\
&\wedge scan_tasks' = (inner_state \cup \{key_to_be_served\}) \\
&\wedge scan_task_status' = StatusWaiting \\
&\wedge last_account_id' = inner_last_account_id + 1 \\
&\wedge pc' = [pc \text{ EXCEPT } ![\text{"SCAN TASK"}] = \text{"SendBlock"}] \\
&\wedge \text{UNCHANGED } \langle response, service_request, key_to_be_served, \\
&\quad block_to_be_served, blocks, stack, \\
&\quad inner_state, inner_accounts, inner_blocks, \\
&\quad inner_last_account_id \rangle \\
\\
\text{SendBlock} &\triangleq \wedge pc[\text{"SCAN TASK"}] = \text{"SendBlock"} \\
&\wedge \vee \wedge block_to_be_served' = [height \mapsto 1, hash \mapsto \text{"111111"}] \\
&\quad \wedge stack' = [stack \text{ EXCEPT } ![\text{"SCAN TASK"}] = \langle [procedure \mapsto \text{"put_block_zcash_client_"}, \\
&\quad \quad \quad pc \mapsto \text{"ScanTaskLoop"}] \rangle \\
&\quad \quad \quad \circ stack[\text{"SCAN TASK"}]] \\
&\quad \wedge pc' = [pc \text{ EXCEPT } ![\text{"SCAN TASK"}] = \text{"PutBlockZcashClientBackend"}] \\
&\vee \wedge \text{TRUE} \\
&\quad \wedge pc' = [pc \text{ EXCEPT } ![\text{"SCAN TASK"}] = \text{"ScanTaskLoop"}] \\
&\quad \wedge \text{UNCHANGED } \langle block_to_be_served, stack \rangle \\
&\wedge \text{UNCHANGED } \langle response, service_request, scan_task_status, \\
&\quad scan_tasks, key_to_be_served, accounts, blocks, \\
&\quad last_account_id, inner_state, inner_accounts, \\
&\quad inner_blocks, inner_last_account_id \rangle \\
\\
\text{ScanTaskLoop} &\triangleq \wedge pc[\text{"SCAN TASK"}] = \text{"ScanTaskLoop"} \\
&\wedge pc' = [pc \text{ EXCEPT } ![\text{"SCAN TASK"}] = \text{"ScanTask"}] \\
&\wedge \text{UNCHANGED } \langle response, service_request, scan_task_status, \\
&\quad scan_tasks, key_to_be_served, \\
&\quad block_to_be_served, accounts, blocks, \\
&\quad last_account_id, stack, inner_state, \\
&\quad inner_accounts, inner_blocks, \\
&\quad inner_last_account_id \rangle \\
\\
scantask &\triangleq \text{GetGlobals} \vee \text{ScanTask} \vee \text{AddingAccount} \vee \text{SendBlock} \\
&\vee \text{ScanTaskLoop}
\end{aligned}$$

$$\begin{aligned}
\text{CreateAccountCall} &\triangleq \wedge pc["MAIN"] = \text{"CreateAccountCall"} \\
&\wedge stack' = [stack \text{ EXCEPT } ![["MAIN"]] = \langle [procedure \mapsto \text{"create_account_grpc"}, \\
&\quad pc \mapsto \text{"End"}] \rangle \\
&\quad \circ stack["MAIN"]] \\
&\wedge pc' = [pc \text{ EXCEPT } ![["MAIN"]] = \text{"CreateAccountGrpc"}] \\
&\wedge \text{UNCHANGED } \langle response, service_request, \\
&\quad scan_task_status, scan_tasks, \\
&\quad key_to_be_served, block_to_be_served, \\
&\quad accounts, blocks, last_account_id, \\
&\quad inner_state, inner_accounts, inner_blocks, \\
&\quad inner_last_account_id \rangle \\
\\
\text{End} &\triangleq \wedge pc["MAIN"] = \text{"End"} \\
&\wedge \text{TRUE} \\
&\wedge pc' = [pc \text{ EXCEPT } ![["MAIN"]] = \text{"Done"}] \\
&\wedge \text{UNCHANGED } \langle response, service_request, scan_task_status, scan_tasks, \\
&\quad key_to_be_served, block_to_be_served, accounts, blocks, \\
&\quad last_account_id, stack, inner_state, inner_accounts, \\
&\quad inner_blocks, inner_last_account_id \rangle \\
\\
\text{Main} &\triangleq \text{CreateAccountCall} \vee \text{End} \\
\\
&\text{Allow infinite stuttering to prevent deadlock on termination.} \\
\text{Terminating} &\triangleq \wedge \forall self \in ProcSet : pc[self] = \text{"Done"} \\
&\wedge \text{UNCHANGED } vars \\
\\
\text{Next} &\triangleq services \vee scantask \vee Main \\
&\quad \vee (\exists self \in ProcSet : \vee create_account_grpc(self) \\
&\quad \vee create_account_zcash_client_backend(self) \\
&\quad \vee put_block_zcash_client_backend(self)) \\
&\quad \vee \text{Terminating} \\
\\
\text{Spec} &\triangleq Init \wedge \Box [Next]_{vars} \\
\\
\text{Termination} &\triangleq \Diamond (\forall self \in ProcSet : pc[self] = \text{"Done"}) \\
\\
&\text{END TRANSLATION}
\end{aligned}$$
