MODULE *PManager*

EXTENDS *Naturals*, *Sequences*

CONSTANTS A, P, *Uris*

ASSUME $A \in Nat$
ASSUME $P \in Nat$

$Permissions \triangleq 1 \,..\, P$
$Applications \triangleq 1 \,..\, A$

$NULL \triangleq$ "NULL"
$DENY \triangleq$ "DENY"
$GRANT \triangleq$ "GRANT"
$ALLOW \triangleq$ "ALLOW"
$REJECT \triangleq$ "REJECT"
$NORMAL \triangleq$ "NORMAL"
$SENSITIVE \triangleq$ "SENSITIVE"

$FLAG\_GRANT\_READ\_URI\_PERMISSION \triangleq$ "FLAG_GRANT_READ_URI_PERMISSION"
$FLAG\_GRANT\_WRITE\_URI\_PERMISSION \triangleq$ "FLAG_GRANT_WRITE_URI_PERMISSION"

$Boolean \triangleq \{\text{TRUE, FALSE}\}$
$Consent \triangleq \{ALLOW, REJECT, NULL\}$
$CustomPermission \triangleq \{NORMAL, SENSITIVE, NULL\}$
$PermissionRequestDecision \triangleq \{GRANT, DENY, NULL\}$
$UriPermModeFlag \triangleq \{FLAG\_GRANT\_READ\_URI\_PERMISSION, FLAG\_GRANT\_WRITE\_URI\_PERM$

** *this* is a comment containing the *PlusCal* code *

```
--algorithm PermissionManager
{
    variables CP = [a ∈ Applications ↦ NULL] ;
              installed = [p ∈ Applications ↦ FALSE] ;
              appPerms = [i ∈ Applications ↦ [p ∈ Permissions ↦ NULL]] ;
              permsInUse = [a ∈ Applications ↦ [i ∈ Permissions ↦ FALSE]] ;
              cpConsent = [a ∈ Applications ↦ [aa ∈ Applications ↦ NULL]] ;
              appPermConsents = [a ∈ Applications ↦ [i ∈ Permissions ↦ NULL]] ;
              appCustomPerms = [i ∈ Applications ↦ [i2 ∈ Applications ↦ NULL]] ;
              manifest = [i ∈ Applications ↦ NULL] ;

    procedure installApp( app ) { INSTALL_APP: installed[app] := TRUE       ; return ; }

    procedure defineCP( app ) { DEFINE_CP: either { CP[app] := NORMAL ; }
                                           or { CP[app] := SENSITIVE ; } ;
                                           return ; }

    procedure askCpFromUser( a1, a2 )
    {
```

```
    ASK_CP_FROM_USER:
    either { appCustomPerms[a1][a2] := GRANT; cpConsent[a1][a2] := ALLOW; }
    or { appCustomPerms[a1][a2] := DENY; cpConsent[a1][a2] := REJECT; };
    return;
}

procedure updateApp( app )
{
    UPDATE_APP: either { CP[app] := NULL; }
                or { call defineCP(app); };
    RETURNING:  return;
}

procedure askAppCP( a1, a2 )
{
    ASK_APP_CP: if ( appCustomPerms[a1][a2] = GRANT ) { return; }

                else if ( appCustomPerms[a1][a2] = DENY ) { return; }

                else
                {
                    if ( CP[a2] = NORMAL ) { appCustomPerms[a1][a2] := GRANT; };
                    else call askCpFromUser(a1, a2);

                    return;
                }
}

procedure uninstallApp( app )
{   Device.uninstallPackage ⇒ PM.runUninstall ⇒ PM.deletePackage ⇒ PackageManagerService.deletePackage ⇒ ...
    says nothing about URI permissions and the deleted package is lost upon entering updatePermissionsLP
    UNINSTALL_APP: installed[app] := FALSE;
                   permsInUse[app] := [p ∈ Permissions ↦ FALSE];
                   appPermConsents[app] := [p ∈ Permissions ↦ NULL];
                   return;
}

procedure systemArbitraryDecision( app, perm )
{
    SYSTEM_ARBITRARY_DECISION:
    either { appPerms[app][perm] := GRANT; appPermConsents[app][perm] := ALLOW; }
    or { appPerms[app][perm] := DENY; appPermConsents[app][perm] := REJECT; };
    return;
}

procedure grantUriPermission( app, uri, mode, perm )
{
    CHECKING_PERMISSION_URI: if ( manifest[app] ≠ uri ) return;
```

```
        ASK_PERMISSION:
        if ( appPerms[app][perm] = GRANT ) { permsInUse[app][perm] := TRUE; return }
        else if ( appPerms[app][perm] = DENY ) { permsInUse[app][perm] := FALSE; return }
        else
        {
            MAKE_DECISION: call systemArbitraryDecision(app, perm);
            USING_PERM: if ( appPerms[app][perm] = GRANT ) { permsInUse[app][perm] := TRUE; } ;
            return;
         }
    }

   fair process ( a ∈ Applications )
   {
        PLATFORM:- while ( TRUE )
        {
            if ( installed[self] = TRUE )
            {
                either { call updateApp(self); }
                or { either { call uninstallApp(self); }
                     or
                     {
                        either { if ( CP[self] = NULL ) { call defineCP(self); } }
                        or
                        {
                            either { with ( application ∈ (Applications \ {self}) ) { call askAppCP(self, ap
                            or { with ( a ∈ Applications \ {self} ) { with ( u ∈ Uris ) { with ( p ∈ Perm
                         }
                      }
                   }
             }
            else
            {
                with ( u ∈ Uris ) { manifest[self] := u; } ;
                call installApp(self);
             }
         } ;
    }
}
```

this ends the comment containing the *PlusCal* code
****

BEGIN TRANSLATION ($chksum(pcal) =$ "$c289dd5b$" $\wedge\ chksum(tla) =$ "$2bd339e2$")
Parameter *app* of procedure *installApp* at line 42 col 26 changed to *app_*
Parameter *app* of procedure *defineCP* at line 44 col 24 changed to *app_d*
Parameter $a1$ of procedure *askCpFromUser* at line 48 col 29 changed to $a1\_$

Parameter $a2$ of procedure *askCpFromUser* at line 48 col 33 changed to $a2\_$
Parameter *app* of procedure *updateApp* at line 56 col 25 changed to app_u
Parameter *app* of procedure *uninstallApp* at line 78 col 28 changed to app_un
Parameter *app* of procedure *systemArbitraryDecision* at line 87 col 39 changed to app_s
Parameter *perm* of procedure *systemArbitraryDecision* at line 87 col 44 changed to $perm\_$

CONSTANT $defaultInitValue$

VARIABLES CP, $installed$, $appPerms$, $permsInUse$, $cpConsent$, $appPermConsents$, $appCustomPerms$, $manifest$, pc, $stack$, $app\_$, app_d, $a1\_$, $a2\_$, app_u, $a1$, $a2$, app_un, app_s, $perm\_$, app, uri, $mode$, $perm$

$$vars \triangleq \langle CP, installed, appPerms, permsInUse, cpConsent, appPermConsents, appCustomPerms, manifest, pc, stack, app\_, app\_d, a1\_, a2\_, app\_u, a1, a2, app\_un, app\_s, perm\_, app, uri, mode, perm\rangle$$

$$ProcSet \triangleq (Applications)$$

$Init \triangleq$ Global variables
$\land\ CP = [a \in Applications \mapsto NULL]$
$\land\ installed = [p \in Applications \mapsto \text{FALSE}]$
$\land\ appPerms = [i \in Applications \mapsto [p \in Permissions \mapsto NULL]]$
$\land\ permsInUse = [a \in Applications \mapsto [i \in Permissions \mapsto \text{FALSE}]]$
$\land\ cpConsent = [a \in Applications \mapsto [aa \in Applications \mapsto NULL]]$
$\land\ appPermConsents = [a \in Applications \mapsto [i \in Permissions \mapsto NULL]]$
$\land\ appCustomPerms = [i \in Applications \mapsto [i2 \in Applications \mapsto NULL]]$
$\land\ manifest = [i \in Applications \mapsto NULL]$
Procedure *installApp*
$\land\ app\_ = [self \in ProcSet \mapsto defaultInitValue]$
Procedure *defineCP*
$\land\ app\_d = [self \in ProcSet \mapsto defaultInitValue]$
Procedure *askCpFromUser*
$\land\ a1\_ = [self \in ProcSet \mapsto defaultInitValue]$
$\land\ a2\_ = [self \in ProcSet \mapsto defaultInitValue]$
Procedure *updateApp*
$\land\ app\_u = [self \in ProcSet \mapsto defaultInitValue]$
Procedure *askAppCP*
$\land\ a1 = [self \in ProcSet \mapsto defaultInitValue]$
$\land\ a2 = [self \in ProcSet \mapsto defaultInitValue]$
Procedure *uninstallApp*
$\land\ app\_un = [self \in ProcSet \mapsto defaultInitValue]$
Procedure *systemArbitraryDecision*
$\land\ app\_s = [self \in ProcSet \mapsto defaultInitValue]$
$\land\ perm\_ = [self \in ProcSet \mapsto defaultInitValue]$
Procedure *grantUriPermission*
$\land\ app = [self \in ProcSet \mapsto defaultInitValue]$
$\land\ uri = [self \in ProcSet \mapsto defaultInitValue]$
$\land\ mode = [self \in ProcSet \mapsto defaultInitValue]$

$\land\ perm = [self \in ProcSet \mapsto defaultInitValue]$
$\land\ stack = [self \in ProcSet \mapsto \langle\rangle]$
$\land\ pc = [self \in ProcSet \mapsto$ "PLATFORM"$]$

$INSTALL\_APP(self) \triangleq \land\ pc[self] =$ "INSTALL_APP"
$\land\ installed' = [installed$ EXCEPT $![app\_[self]] =$ TRUE$]$
$\land\ pc' = [pc$ EXCEPT $![self] = Head(stack[self]).pc]$
$\land\ app\_' = [app\_$ EXCEPT $![self] = Head(stack[self]).app\_]$
$\land\ stack' = [stack$ EXCEPT $![self] = Tail(stack[self])]$
$\land$ UNCHANGED $\langle CP, appPerms, permsInUse, cpConsent,$
$appPermConsents, appCustomPerms, manifest,$
$app\_d, a1\_, a2\_, app\_u, a1, a2, app\_un,$
$app\_s, perm\_, app, uri, mode, perm\rangle$

$installApp(self) \triangleq INSTALL\_APP(self)$

$DEFINE\_CP(self) \triangleq \land\ pc[self] =$ "DEFINE_CP"
$\land\ \lor\ \land\ CP' = [CP$ EXCEPT $![app\_d[self]] = NORMAL]$
$\lor\ \land\ CP' = [CP$ EXCEPT $![app\_d[self]] = SENSITIVE]$
$\land\ pc' = [pc$ EXCEPT $![self] = Head(stack[self]).pc]$
$\land\ app\_d' = [app\_d$ EXCEPT $![self] = Head(stack[self]).app\_d]$
$\land\ stack' = [stack$ EXCEPT $![self] = Tail(stack[self])]$
$\land$ UNCHANGED $\langle installed, appPerms, permsInUse, cpConsent,$
$appPermConsents, appCustomPerms, manifest,$
$app\_, a1\_, a2\_, app\_u, a1, a2, app\_un,$
$app\_s, perm\_, app, uri, mode, perm\rangle$

$defineCP(self) \triangleq DEFINE\_CP(self)$

$ASK\_CP\_FROM\_USER(self) \triangleq \land\ pc[self] =$ "ASK_CP_FROM_USER"
$\land\ \lor\ \land\ appCustomPerms' = [appCustomPerms$ EXCEPT $![a1\_[self]][a2\_[se$
$\land\ cpConsent' = [cpConsent$ EXCEPT $![a1\_[self]][a2\_[self]] = ALLOW$
$\lor\ \land\ appCustomPerms' = [appCustomPerms$ EXCEPT $![a1\_[self]][a2\_[se$
$\land\ cpConsent' = [cpConsent$ EXCEPT $![a1\_[self]][a2\_[self]] = REJEC$
$\land\ pc' = [pc$ EXCEPT $![self] = Head(stack[self]).pc]$
$\land\ a1\_' = [a1\_$ EXCEPT $![self] = Head(stack[self]).a1\_]$
$\land\ a2\_' = [a2\_$ EXCEPT $![self] = Head(stack[self]).a2\_]$
$\land\ stack' = [stack$ EXCEPT $![self] = Tail(stack[self])]$
$\land$ UNCHANGED $\langle CP, installed, appPerms, permsInUse,$
$appPermConsents, manifest, app\_,$
$app\_d, app\_u, a1, a2, app\_un, app\_s,$
$perm\_, app, uri, mode, perm\rangle$

$askCpFromUser(self) \triangleq ASK\_CP\_FROM\_USER(self)$

$UPDATE\_APP(self) \triangleq \land\ pc[self] =$ "UPDATE_APP"
$\land\ \lor\ \land\ CP' = [CP$ EXCEPT $![app\_u[self]] = NULL]$

```
                    ∧ pc′  = [pc EXCEPT ![self] = "RETURNING"]
                    ∧ UNCHANGED ⟨stack, app_d⟩
                  ∨ ∧ ∧ app_d′ = [app_d EXCEPT ![self] = app_u[self]]
                      ∧ stack′ = [stack EXCEPT ![self] = ⟨[procedure ↦ "defineCP",
                                                          pc        ↦ "RETURNING",
                                                          app_d     ↦ app_d[self]]⟩
                                                         ∘ stack[self]]
                    ∧ pc′  = [pc EXCEPT ![self] = "DEFINE_CP"]
                    ∧ CP′ = CP
                  ∧ UNCHANGED ⟨installed, appPerms, permsInUse, cpConsent,
                               appPermConsents, appCustomPerms, manifest,
                               app_, a1_, a2_, app_u, a1, a2, app_un,
                               app_s, perm_, app, uri, mode, perm⟩

RETURNING(self) ≜ ∧ pc[self] = "RETURNING"
                  ∧ pc′ = [pc EXCEPT ![self] = Head(stack[self]).pc]
                  ∧ app_u′ = [app_u EXCEPT ![self] = Head(stack[self]).app_u]
                  ∧ stack′  = [stack EXCEPT ![self]  = Tail(stack[self])]
                  ∧ UNCHANGED ⟨CP, installed, appPerms, permsInUse,
                               cpConsent, appPermConsents, appCustomPerms,
                               manifest, app_, app_d, a1_, a2_, a1, a2,
                               app_un, app_s, perm_, app, uri, mode, perm⟩

updateApp(self) ≜ UPDATE_APP(self) ∨ RETURNING(self)

ASK_APP_CP(self) ≜ ∧ pc[self] = "ASK_APP_CP"
                   ∧ IF appCustomPerms[a1[self]][a2[self]] = GRANT
                     THEN ∧ pc′ = [pc EXCEPT ![self] = Head(stack[self]).pc]
                          ∧ a1′ = [a1 EXCEPT ![self] = Head(stack[self]).a1]
                          ∧ a2′ = [a2 EXCEPT ![self] = Head(stack[self]).a2]
                          ∧ stack′ = [stack EXCEPT ![self] = Tail(stack[self])]
                          ∧ UNCHANGED ⟨appCustomPerms, a1_, a2_⟩
                     ELSE ∧ IF appCustomPerms[a1[self]][a2[self]] = DENY
                             THEN ∧ pc′ = [pc EXCEPT ![self] = Head(stack[self]).pc]
                                  ∧ a1′ = [a1 EXCEPT ![self] = Head(stack[self]).a1]
                                  ∧ a2′ = [a2 EXCEPT ![self] = Head(stack[self]).a2]
                                  ∧ stack′ = [stack EXCEPT ![self] = Tail(stack[self])]
                                  ∧ UNCHANGED ⟨appCustomPerms, a1_,
                                               a2_⟩
                             ELSE ∧ IF CP[a2[self]] = NORMAL
                                     THEN ∧ appCustomPerms′ = [appCustomPerms EXC
                                          ∧ pc′ = [pc EXCEPT ![self] = "Error"]
                                          ∧ UNCHANGED ⟨stack,
                                                       a1_, a2_⟩
                                     ELSE ∧ ∧ a1_′ = [a1_ EXCEPT ![self] = a1[self]]
                                            ∧ a2_′ = [a2_ EXCEPT ![self] = a2[self]]
```

$\land\ stack' = [stack \text{ EXCEPT } ![self] = \langle [procedu$
pc
$a1\_$
$a2\_$
$\circ\ Tail($
$\land\ pc' = [pc \text{ EXCEPT } ![self] =$ "ASK_CP_FROM
$\land$ UNCHANGED $appCustomPerms$
$\land$ UNCHANGED $\langle a1,\ a2 \rangle$
$\land$ UNCHANGED $\langle CP,\ installed,\ appPerms,\ permsInUse,$
$cpConsent,\ appPermConsents,\ manifest,\ app\_,$
$app\_d,\ app\_u,\ app\_un,\ app\_s,\ perm\_,\ app,$
$uri,\ mode,\ perm \rangle$

$askAppCP(self) \triangleq ASK\_APP\_CP(self)$

$UNINSTALL\_APP(self) \triangleq$
$\land\ pc[self] =$ "UNINSTALL_APP"
$\land\ installed' = [installed \text{ EXCEPT } ![app\_un[self]] = \text{FALSE}]$
$\land\ permsInUse' = [permsInUse \text{ EXCEPT } ![app\_un[self]] = [p \in Permissions \mapsto \text{FA}$
$\land\ appPermConsents' = [appPermConsents \text{ EXCEPT } ![app\_un[self]] = [p \in Perm$
$\land\ pc' = [pc \text{ EXCEPT } ![self] = Head(stack[self]).pc]$
$\land\ app\_un' = [app\_un \text{ EXCEPT } ![self] = Head(stack[self]).app\_un]$
$\land\ stack' = [stack \text{ EXCEPT } ![self] = Tail(stack[self])]$
$\land$ UNCHANGED $\langle CP,\ appPerms,\ cpConsent,\ appCustomPerms,$
$manifest,\ app\_,\ app\_d,\ a1\_,\ a2\_,\ app\_u,$
$a1,\ a2,\ app\_s,\ perm\_,\ app,\ uri,\ mode,$
$perm \rangle$

$uninstallApp(self) \triangleq UNINSTALL\_APP(self)$

$SYSTEM\_ARBITRARY\_DECISION(self) \triangleq$
$\land\ pc[self] =$ "SYSTEM_ARBITRARY_DECISION"
$\land\ \lor\ \land\ appPerms' = [appPerms \text{ EXCEPT } ![app\_s[self]][perm\_$
$\land\ appPermConsents' = [appPermConsents \text{ EXCEPT } ![ap$
$\lor\ \land\ appPerms' = [appPerms \text{ EXCEPT } ![app\_s[self]][perm\_$
$\land\ appPermConsents' = [appPermConsents \text{ EXCEPT } ![ap$
$\land\ pc' = [pc \text{ EXCEPT } ![self] = Head(stack[self]).pc]$
$\land\ app\_s' = [app\_s \text{ EXCEPT } ![self] = Head(stack[self]).app\_s$
$\land\ perm\_' = [perm\_ \text{ EXCEPT } ![self] = Head(stack[self]).perm\_$
$\land\ stack' = [stack \text{ EXCEPT } ![self] = Tail(stack[self])]$
$\land$ UNCHANGED $\langle CP,\ installed,\ permsInUse,$
$cpConsent,\ appCustomPerms,$
$manifest,\ app\_,\ app\_d,\ a1\_,$
$a2\_,\ app\_u,\ a1,\ a2,\ app\_un,$
$app,\ uri,\ mode,\ perm \rangle$

$systemArbitraryDecision(self) \triangleq SYSTEM\_ARBITRARY\_DECISION(self)$

$CHECKING\_PERMISSION\_URI(self) \triangleq \land\ pc[self] =$ "CHECKING_PERMISSION_URI"

$\land$ IF $manifest[app[self]] \neq uri[self]$
THEN $\land\ pc' = [pc$ EXCEPT $![self] = Head(stack[self]).pc]$
$\land\ app' = [app$ EXCEPT $![self] = Head(stack[self]).ap$
$\land\ uri' = [uri$ EXCEPT $![self] = Head(stack[self]).ur$
$\land\ mode' = [mode$ EXCEPT $![self] = Head(stack[self]$
$\land\ perm' = [perm$ EXCEPT $![self] = Head(stack[self]$
$\land\ stack' = [stack$ EXCEPT $![self] = Tail(stack[self])]$
ELSE $\land\ pc' = [pc$ EXCEPT $![self] =$ "ASK_PERMISSION"$]$
$\land$ UNCHANGED $\langle stack, app, uri,$
$mode, perm\rangle$
$\land$ UNCHANGED $\langle CP, installed, appPerms,$
$permsInUse, cpConsent,$
$appPermConsents,$
$appCustomPerms, manifest,$
$app\_, app\_d, a1\_, a2\_, app\_u,$
$a1, a2, app\_un, app\_s, perm\_\rangle$

$ASK\_PERMISSION(self) \triangleq$ $\land\ pc[self] =$ "ASK_PERMISSION"
$\land$ IF $appPerms[app[self]][perm[self]] = GRANT$
THEN $\land\ permsInUse' = [permsInUse$ EXCEPT $![app[self]][perm[self]] =$ T
$\land\ pc' = [pc$ EXCEPT $![self] = Head(stack[self]).pc]$
$\land\ app' = [app$ EXCEPT $![self] = Head(stack[self]).app]$
$\land\ uri' = [uri$ EXCEPT $![self] = Head(stack[self]).uri]$
$\land\ mode' = [mode$ EXCEPT $![self] = Head(stack[self]).mode]$
$\land\ perm' = [perm$ EXCEPT $![self] = Head(stack[self]).perm]$
$\land\ stack' = [stack$ EXCEPT $![self] = Tail(stack[self])]$
ELSE $\land$ IF $appPerms[app[self]][perm[self]] = DENY$
THEN $\land\ permsInUse' = [permsInUse$ EXCEPT $![app[self]][pe$
$\land\ pc' = [pc$ EXCEPT $![self] = Head(stack[self]).pc]$
$\land\ app' = [app$ EXCEPT $![self] = Head(stack[self]).app$
$\land\ uri' = [uri$ EXCEPT $![self] = Head(stack[self]).uri$
$\land\ mode' = [mode$ EXCEPT $![self] = Head(stack[self]).$
$\land\ perm' = [perm$ EXCEPT $![self] = Head(stack[self]).$
$\land\ stack' = [stack$ EXCEPT $![self] = Tail(stack[self])]$
ELSE $\land\ pc' = [pc$ EXCEPT $![self] =$ "MAKE_DECISION"$]$
$\land$ UNCHANGED $\langle permsInUse,$
$stack, app, uri,$
$mode, perm\rangle$
$\land$ UNCHANGED $\langle CP, installed, appPerms, cpConsent,$
$appPermConsents, appCustomPerms,$
$manifest, app\_, app\_d, a1\_, a2\_, app\_u,$
$a1, a2, app\_un, app\_s, perm\_\rangle$

$MAKE\_DECISION(self) \triangleq$ $\land\ pc[self] =$ "MAKE_DECISION"
$\land\ \land\ app\_s' = [app\_s$ EXCEPT $![self] = app[self]]$

$\land\ perm\_' = [perm\_ \text{ EXCEPT } ![self] = perm[self]]$
$\land\ stack' = [stack \text{ EXCEPT } ![self] = \langle [procedure \mapsto$ "systemArbitraryDecision
$pc \mapsto$ "USING_PERM",
$app\_s \mapsto app\_s[self]$,
$perm\_ \mapsto perm\_[self]] \rangle$
$\circ\ stack[self]]$
$\land\ pc' = [pc \text{ EXCEPT } ![self] =$ "SYSTEM_ARBITRARY_DECISION"$]$
$\land$ UNCHANGED $\langle CP, installed, appPerms, permsInUse,$
$cpConsent, appPermConsents,$
$appCustomPerms, manifest, app\_, app\_d,$
$a1\_, a2\_, app\_u, a1, a2, app\_un, app,$
$uri, mode, perm\rangle$

$USING\_PERM(self) \triangleq \land\ pc[self] =$ "USING_PERM"
$\land$ IF $appPerms[app[self]][perm[self]] = GRANT$
THEN $\land\ permsInUse' = [permsInUse \text{ EXCEPT } ![app[self]][perm[self]] =$ TRUE
ELSE $\land$ TRUE
$\land$ UNCHANGED $permsInUse$
$\land\ pc' = [pc \text{ EXCEPT } ![self] = Head(stack[self]).pc]$
$\land\ app' = [app \text{ EXCEPT } ![self] = Head(stack[self]).app]$
$\land\ uri' = [uri \text{ EXCEPT } ![self] = Head(stack[self]).uri]$
$\land\ mode' = [mode \text{ EXCEPT } ![self] = Head(stack[self]).mode]$
$\land\ perm' = [perm \text{ EXCEPT } ![self] = Head(stack[self]).perm]$
$\land\ stack' = [stack \text{ EXCEPT } ![self] = Tail(stack[self])]$
$\land$ UNCHANGED $\langle CP, installed, appPerms, cpConsent,$
$appPermConsents, appCustomPerms, manifest,$
$app\_, app\_d, a1\_, a2\_, app\_u, a1, a2,$
$app\_un, app\_s, perm\_\rangle$

$grantUriPermission(self) \triangleq CHECKING\_PERMISSION\_URI(self)$
$\lor\ ASK\_PERMISSION(self)$
$\lor\ MAKE\_DECISION(self) \lor USING\_PERM(self)$

$PLATFORM(self) \triangleq \land\ pc[self] =$ "PLATFORM"
$\land$ IF $installed[self] =$ TRUE
THEN $\land \lor \land \land\ app\_u' = [app\_u \text{ EXCEPT } ![self] = self]$
$\land\ stack' = [stack \text{ EXCEPT } ![self] = \langle [procedure \mapsto$ "updateApp",
$pc \mapsto$ "PLATFORM
$app\_u \mapsto app\_u[self]] \rangle$
$\circ\ stack[self]]$
$\land\ pc' = [pc \text{ EXCEPT } ![self] =$ "UPDATE_APP"$]$
$\land$ UNCHANGED $\langle app\_d, a1, a2, app\_un, app, uri, mode, perm\rangle$
$\lor \land \lor \land \land\ app\_un' = [app\_un \text{ EXCEPT } ![self] = self]$
$\land\ stack' = [stack \text{ EXCEPT } ![self] = \langle [procedure \mapsto$ "uninstallA
$pc \mapsto$ "PLATFO
$app\_un \mapsto app\_un[se$

$\circ\ stack[self]]$

$\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“UNINSTALL\_APP”}]$

$\land \text{ UNCHANGED } \langle app\_d,\ a1,\ a2,\ app,\ uri,\ mode,\ perm \rangle$

$\lor\ \land\ \lor\ \land \text{ IF } CP[self] = NULL$

$\text{THEN } \land\ \land\ app\_d' = [app\_d \text{ EXCEPT } ![self] = self]$

$\land\ stack' = [stack \text{ EXCEPT } ![self] = \langle [proc$

pc

$app\_$

$\circ\ sta$

$\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“DEFINE\_CP”}]$

$\text{ELSE } \land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“PLATFORM”}]$

$\land \text{ UNCHANGED } \langle stack,$

$app\_d \rangle$

$\land \text{ UNCHANGED } \langle a1,\ a2,\ app,\ uri,\ mode,\ perm \rangle$

$\lor\ \land\ \lor\ \land\ \exists\, application \in (Applications \setminus \{self\}) :$

$\land\ \land\ a1' = [a1 \text{ EXCEPT } ![self] = self]$

$\land\ a2' = [a2 \text{ EXCEPT } ![self] = application]$

$\land\ stack' = [stack \text{ EXCEPT } ![self] = \langle [procedure$

pc

$a1$

$a2$

$\circ\ stack[se$

$\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“ASK\_APP\_CP”}]$

$\land \text{ UNCHANGED } \langle app,\ uri,\ mode,\ perm \rangle$

$\lor\ \land\ \exists\, a \in Applications \setminus \{self\} :$

$\exists\, u \in Uris :$

$\exists\, p \in Permissions :$

$\land\ \land\ app' = [app \text{ EXCEPT } ![self] = a]$

$\land\ mode' = [mode \text{ EXCEPT } ![self] = FLAG$

$\land\ perm' = [perm \text{ EXCEPT } ![self] = p]$

$\land\ stack' = [stack \text{ EXCEPT } ![self] = \langle [proce$

pc

app

uri

$mode$

$perm$

$\circ\ stac$

$\land\ uri' = [uri \text{ EXCEPT } ![self] = u]$

$\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“CHECKING\_PE}$

$\land \text{ UNCHANGED } \langle a1,\ a2 \rangle$

$\land\ app\_d' = app\_d$

$\land \text{ UNCHANGED } app\_un$

$\land\ app\_u' = app\_u$

$\land \text{ UNCHANGED } \langle manifest,\ app\_ \rangle$

$\text{ELSE } \land\ \exists\, u \in Uris :$

$$
\begin{array}{l}
\quad manifest' = [manifest \text{ EXCEPT } ![self] = u] \\
\land\ \land\ app\_' = [app\_ \text{ EXCEPT } ![self] = self] \\
\quad\ \land\ stack' = [stack \text{ EXCEPT } ![self] = \langle [procedure \mapsto \text{“installApp”}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad pc \mapsto \text{“PLATFORM”}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad app\_ \mapsto app\_[self]] \rangle \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \circ\ stack[self]] \\
\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“INSTALL\_APP”}] \\
\land \text{ UNCHANGED } \langle app\_d,\ app\_u,\ a1,\ a2,\ app\_un,\ app, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad uri,\ mode,\ perm \rangle \\
\land \text{ UNCHANGED } \langle CP,\ installed,\ appPerms,\ permsInUse, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad cpConsent,\ appPermConsents,\ appCustomPerms, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad a1\_,\ a2\_,\ app\_s,\ perm\_ \rangle
\end{array}
$$

$$a(self) \triangleq PLATFORM(self)$$

$$
\begin{array}{l}
Next \triangleq (\exists\, self \in ProcSet : \ \lor\ installApp(self) \lor defineCP(self) \\
\qquad\qquad\qquad\qquad\qquad\qquad\ \lor\ askCpFromUser(self) \lor updateApp(self) \\
\qquad\qquad\qquad\qquad\qquad\qquad\ \lor\ askAppCP(self) \lor uninstallApp(self) \\
\qquad\qquad\qquad\qquad\qquad\qquad\ \lor\ systemArbitraryDecision(self) \\
\qquad\qquad\qquad\qquad\qquad\qquad\ \lor\ grantUriPermission(self)) \\
\qquad\quad \lor\ (\exists\, self \in Applications : a(self))
\end{array}
$$

$$
\begin{array}{l}
Spec \triangleq \ \land\ Init \land \Box[Next]_{vars} \\
\qquad\ \land\ \forall\, self \in Applications : \ \land \mathrm{WF}_{vars}((pc[self] \neq \text{“PLATFORM”}) \land a(self)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\ \land \mathrm{WF}_{vars}(updateApp(self)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\ \land \mathrm{WF}_{vars}(uninstallApp(self)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\ \land \mathrm{WF}_{vars}(defineCP(self)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\ \land \mathrm{WF}_{vars}(askAppCP(self)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\ \land \mathrm{WF}_{vars}(grantUriPermission(self)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\ \land \mathrm{WF}_{vars}(installApp(self)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\ \land \mathrm{WF}_{vars}(askCpFromUser(self)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\ \land \mathrm{WF}_{vars}(systemArbitraryDecision(self))
\end{array}
$$

END TRANSLATION

$$TypeOK \triangleq \ \land\ appCustomPerms \in [Applications \to [Applications \to PermissionRequestDecision]]$$

$$
\begin{array}{l}
CpConsent \triangleq \ \Box\neg(\ \land \exists\, m \in Applications : CP[m] = SENSITIVE \\
\qquad\qquad\qquad\qquad\ \land \exists\, n \ \ \in Applications : \\
\qquad\qquad\qquad\qquad\quad \land\ m \neq n \\
\qquad\qquad\qquad\qquad\quad \land\ appCustomPerms[n][m] = GRANT \\
\qquad\qquad\qquad\qquad\quad \land\ cpConsent[n][m] \neq ALLOW)
\end{array}
$$

$$
\begin{array}{l}
UriPermConsent \triangleq \ \Box\neg(\ \land \exists\, application \in Applications : \exists\, permission \in Permissions : \quad \text{Bagheri} \\
\qquad\qquad\qquad\qquad\qquad\ \land\ appPermConsents[application][permission] \neq ALLOW \\
\qquad\qquad\qquad\qquad\qquad\ \land\ appPerms[application][permission] = GRANT \\
\qquad\qquad\qquad\qquad\qquad\ \land\ permsInUse[application][permission] = \text{TRUE})
\end{array}
$$

$$Authorized \triangleq \Box\neg(\ \wedge \exists\, application \in Applications : \exists\, permission \in Permissions : \quad \text{System consent}$$
$$\wedge\ appPerms[application][permission] \neq GRANT$$
$$\wedge\ permsInUse[application][permission] = \text{TRUE})$$

* Modification History
* Last modified *Wed Mar* 29 21:46:56 *GMT*+03:30 2023 by *Amirhosein*