Access Control for Smart Home IoT: EGRBAC Model

Safwa Ameer

James Benson

Ravi Sandhu

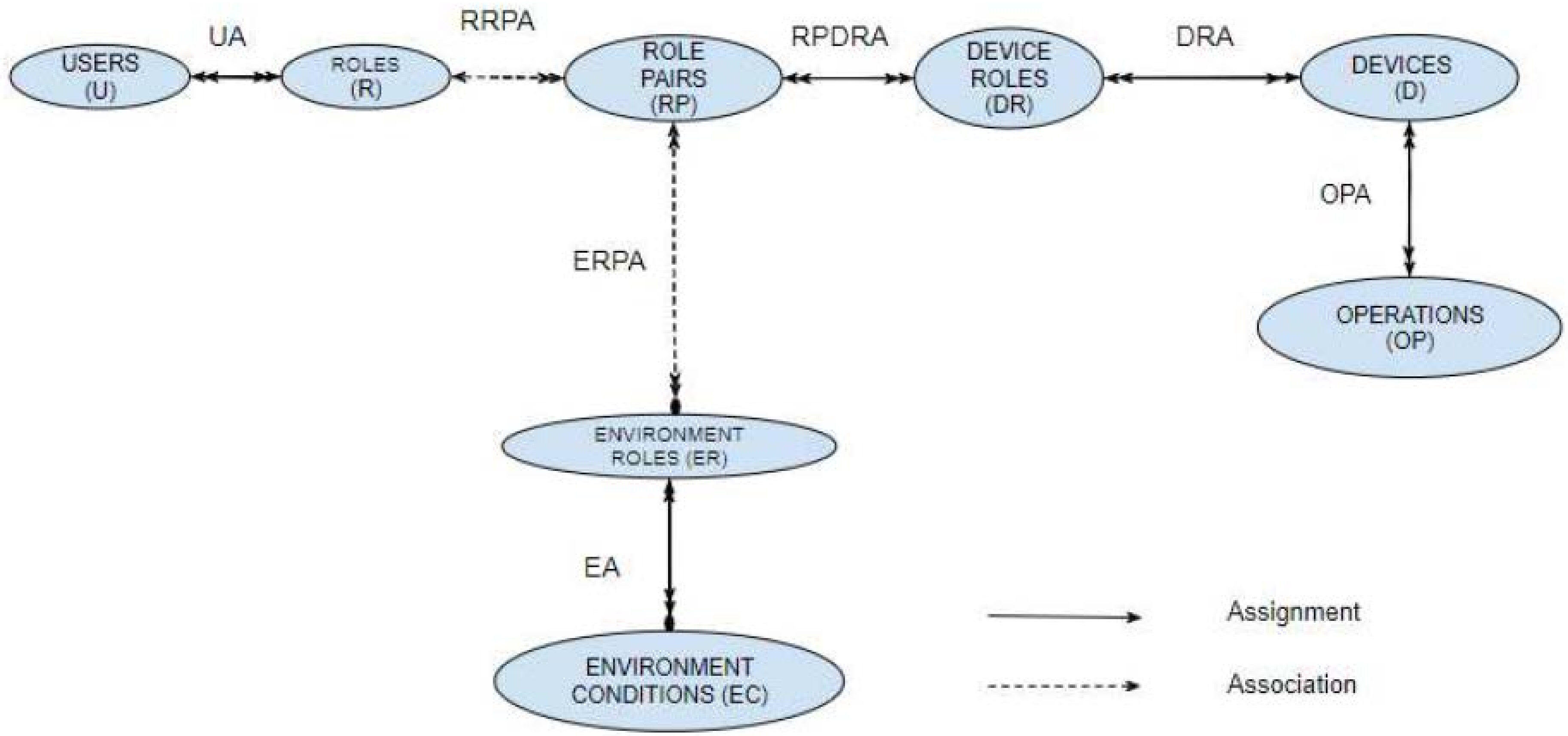
Institute for Cyber Security (ICS)
Center for Security and Privacy Enhanced Cloud Computing (C-SPECC)
Department of Computer Science
University of Texas at San Antonio

Safwa.ameer@my.utsa.edu

L11-2

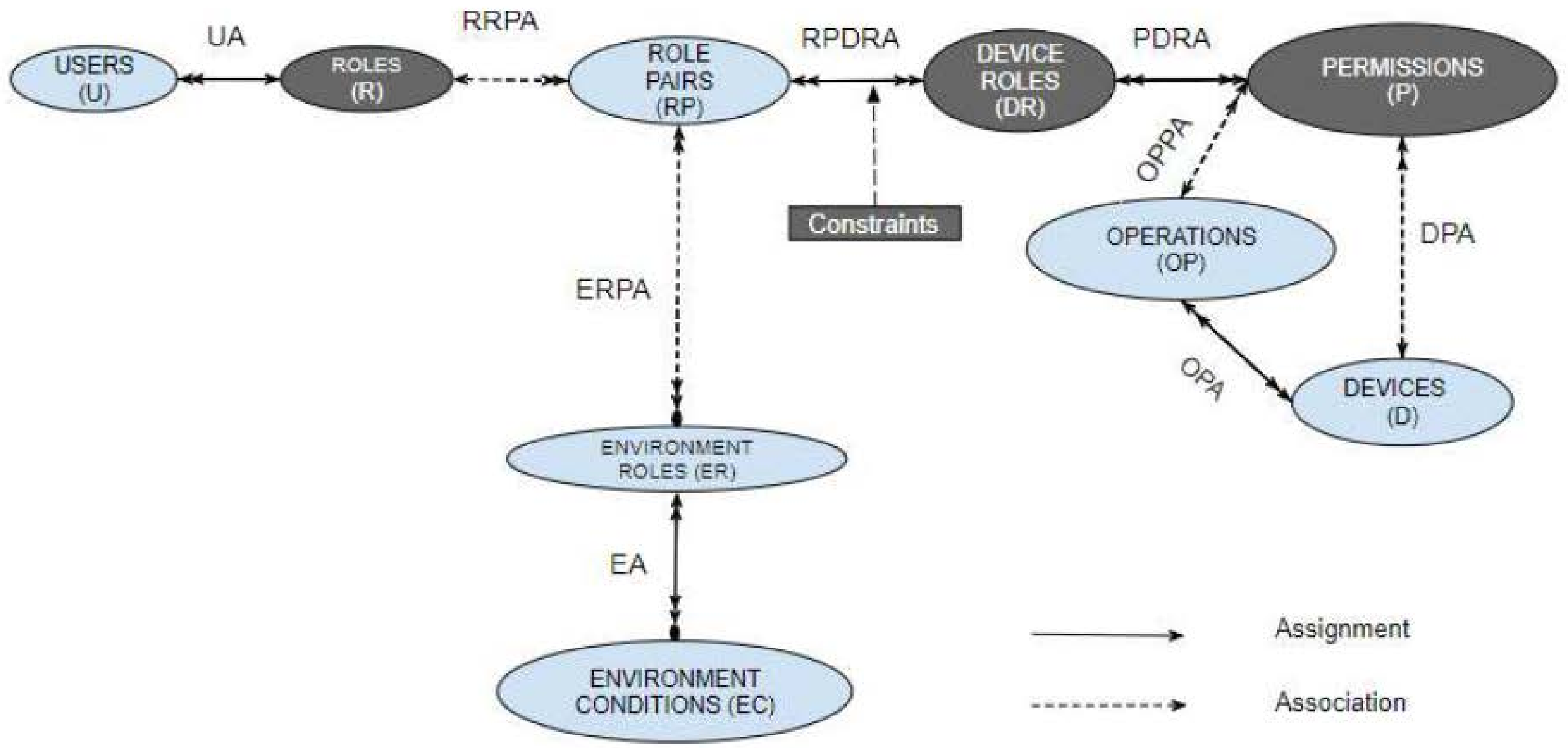
Spring 2020

Our view of GRBAC



World-Leading Research with Real-World Impact!

The EGRBAC Model



EGRBAC Model Formalization

Users, Roles and Devices

–U, R, D, P, OP, DR and $Constraints$ are sets of users, roles, devices, permissions, operations, device roles, and constraints respectively

–$UA \subseteq U \times R$, many to one users to role assignment (home owner specified)

–$OPA \subseteq OP \times D$, a many to many assignment between operations and its target device (manufacturer specified)

–$P \subseteq D \times OP$, every permission is a mapping between an operation and its device (manufacturer specified)

– For convenience for every $p = (d_i, op_j) \in P$, let $p.d = d_i$ and $p.op = op_j$

– $OPPA \subseteq OP \times P$, a one to many operation to permissions association, where: $OPPA = \{(op_i, p_j) \mid p_j.op = op_i \wedge p_j \in P\}$

– $DPA \subseteq D \times P$, a one to many device to permissions association, where: $DPA = \{(d_i, p_j) \mid p_j.d = d_i \wedge p_j \in P\}$

– $PDRA \subseteq P \times DR$, a many to many permissions to device roles assignment (home owner specified)

Environment Roles and Environment Conditions

–ER and EC are sets of environment roles and environment conditions respectively

–$EA \subseteq 2^{EC} \times ER$, many to many subsets of environment conditions to environment roles assignment (home owner specified)

Role Pairs

–$RP \subseteq R \times 2^{ER}$, a set of role pairs specifying all permissible combinations of a user role and subsets of environment roles (home owner specified)

– For convenience for every $rp = (r_i, ER_j) \in RP$, let $rp.r = r_i$ and $rp.ER = ER_j$

–$RRPA \subseteq R \times RP$, one to many role to role pairs association, where $RRPA = \{(r_m, rp_n) \mid rp_n \in RP \wedge rp_n.r = r_m\}$

–$ERPA \subseteq ER \times RP$, many to many environment roles to role pairs association, where $ERPA = \{(er_m, rp_n) \mid rp_n \in RP \wedge er_m \in rp_n.ER\}$

Role Pair Assignment

–$RPDRA \subseteq RP \times DR$, many to many role pairs to device roles assignment (home owner specified)

Authorization Predicate

– For a user u_i to perform operation op_k on device d_j when the set of environment conditions EC_m is active:

$(op_k, d_j) \in OPA \wedge$

$(\exists\, r_x \in R, rp_y \in RP, dr_l \in DR)$, where:

$(u_i, r_x) \in UA \wedge rp_y.r = r_x \wedge (rp_y, dr_l) \in RPDRA \wedge ((op_k, d_j), dr_l) \in PDRA \wedge$

$rp_y.ER \subseteq \{er \in ER \mid (\exists EC'_m \subseteq EC_m)[(EC'_m, er) \in EA]\}$

EGRBAC Vs GRBAC

- In EGRBAC, a Role (R) specifically represents the relationship between the user and the family.

- Unlike GRBAC, EGRBAC requires each user to have a single role.

- There are two newly introduced elements:
  1. Permissions, an approval to perform an operation on one device, in other words it is a mapping between an operation and its owner device.
  2. Constraints, which are conditions that need to be satisfied when assigning device roles to role pairs, will be discussed later.

- Unlike in GRBAC where a device role is a way of categorizing devices, in EGRBAC a Device Role is a mean of categorizing permissions of different devices.

I·C·S
The Institute for Cyber Security

Use Case 1.B Configuration in EGRBAC

C·SPECC
Center for Security and Privacy Enhanced Cloud Computing

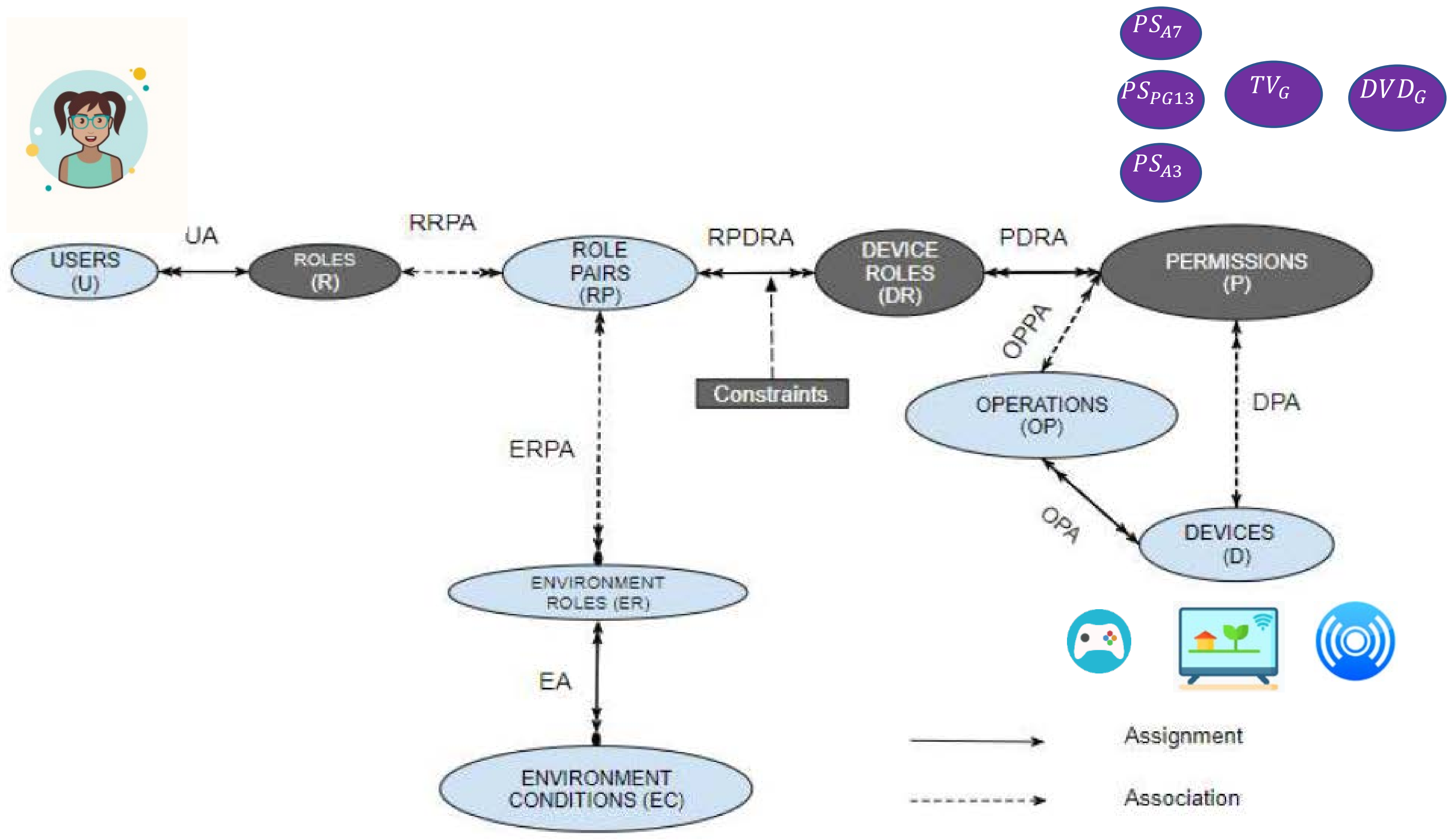
$$
\begin{aligned}
&U = \{alex, bob, \dots\},\ R = \{kids, parents, \dots\}\\
&UA = \{(alex, kids), (bob, parents), \dots\}\\
&D = \{TV, DVD, Playstation, \dots\}\\
&OP_{TV} = \{G, PG, PG13, R, NC-17, \dots\}\\
&OP_{DVD} = \{G, PG, PG13, R, NC-17, \dots\}\\
&OP_{Playstation} = \{A3, A7, PG12, A16, A18, BuyGames,\\
&\qquad InternetBrowsing, Texting, VoiceMessaging, \dots\}\\
&OP = OP_{TV} \cup OP_{DVD} \cup OP_{Playstation} \cup \dots\\
&P_{TV} = \{TV_G, TV_{PG}, TV_{PG13}, TV_R, TV_{NC-17}, \dots\}\\
&P_{DVD} = \{DVD_G, DVD_{PG}, DVD_{PG13}, DVD_R, DVD_{NC-17}, \dots\}\\
&P_{Playstation} = \{PS_{A3}, PS_{A7}, PS_{PG12}, PS_{A16}, PS_{A18}, PS_{BuyGames},\\
&\qquad PS_{InternetBrowsing}, PS_{Texting}, PS_{VoiceMessaging}, \dots\}\\
&P = P_{TV} \cup P_{DVD} \cup P_{Playstation} \cup \dots\\
&DR = \{Entertainment\_Devices, Kids\_Friendly\_Contents, \dots\}\\
&PDRA = (P_{TV} \times \{Entertainment\_Devices\}) \cup\\
&\qquad (P_{DVD} \times \{Entertainment\_Devices\}) \cup\\
&\qquad (P_{Playstation} \times \{Entertainment\_Devices\}) \cup\\
&\qquad \{(TV_G, Kids\_Friendly\_Content),\\
&\qquad (DVD_G, Kids\_Friendly\_Contents)\} \cup\\
&\qquad (\{PS_{A3}, PS_{A7}, PS_{PG12}\} \times\\
&\qquad\qquad \{Kids\_Friendly\_Contents\}) \cup \dots
\end{aligned}
$$

$$
\begin{aligned}
&EC = \{weekends, evenings, TRUE \dots\}\\
&ER = \{Entertainment\_Time, Any\_Time, \dots\}\\
&EA = \{(\{weekends, evenings\}, Entertainment\_Time),\\
&\qquad (TRUE, Any\_Time), \dots\}
\end{aligned}
$$

$$
\begin{aligned}
&RP = \{(kids, \{Entertainment\_Time\}), (parents, \{Any\_Time\}), \dots\}\\
&RPDRA = \{((kids, \{Entertainment\_Time\}),\\
&\qquad\qquad Kids\_Friendly\_Contents),\\
&\qquad ((parents, \{Any\_Time\}),\\
&\qquad\qquad Entertainment\_Devices), \dots\}
\end{aligned}
$$

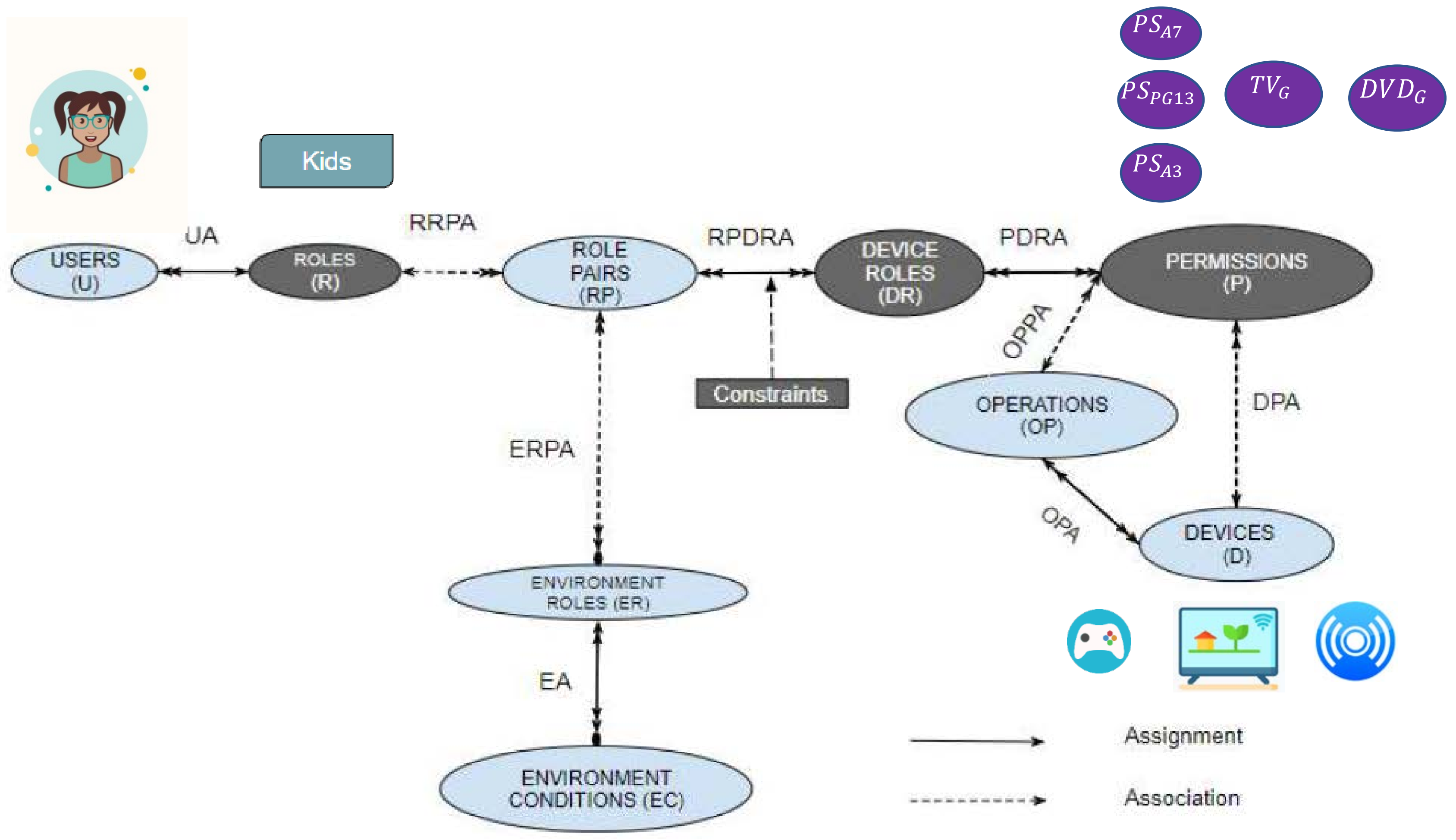
UTSA Computer Science

Use Case 1.B



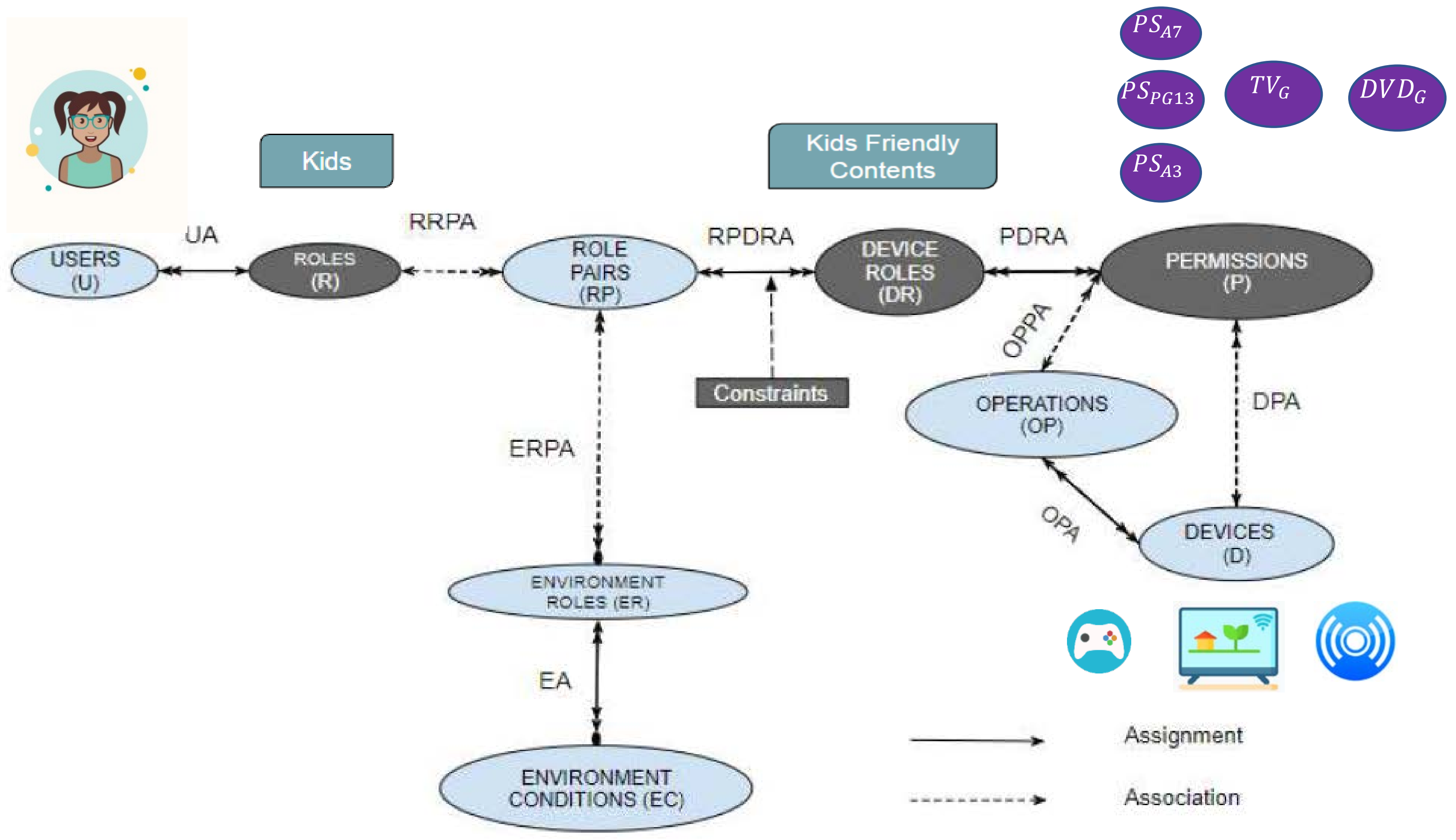
PS_{A7}
PS_{PG13}
TV_G
DVD_G
PS_{A3}
USERS (U)
UA
ROLES (R)
RRPA
ROLE PAIRS (RP)
RPDRA
DEVICE ROLES (DR)
PDRA
PERMISSIONS (P)
Constraints
OPPA
OPERATIONS (OP)
DPA
ERPA
OPA
DEVICES (D)
ENVIRONMENT ROLES (ER)
EA
ENVIRONMENT CONDITIONS (EC)
Assignment
Association

Use Case 1.B

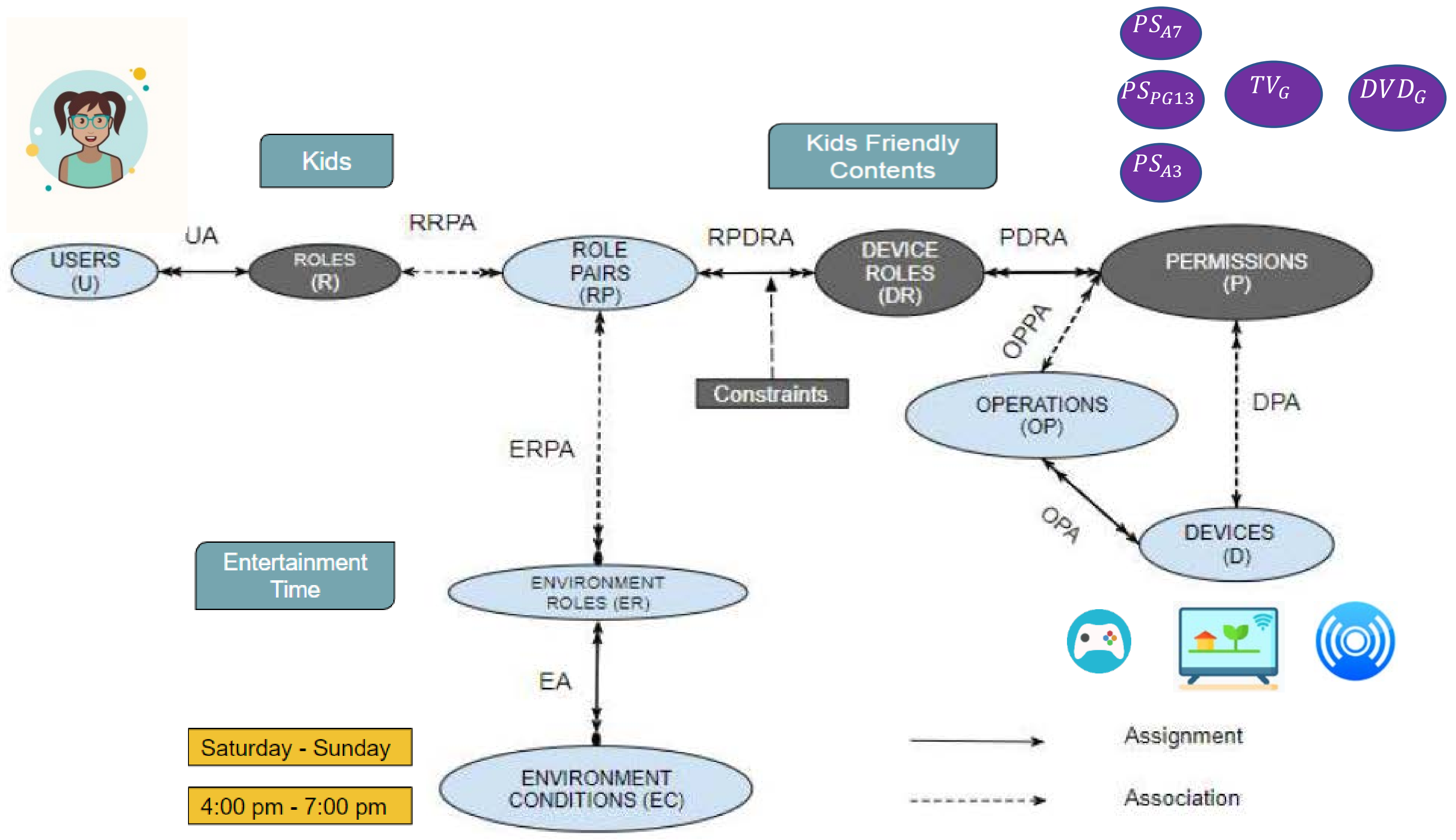


PS_{A7}
PS_{PG13}
TV_G
DVD_G
PS_{A3}
Kids
UA
RRPA
RPDRA
PDRA
USERS (U)
ROLES (R)
ROLE PAIRS (RP)
DEVICE ROLES (DR)
PERMISSIONS (P)
OPPA
Constraints
OPERATIONS (OP)
DPA
ERPA
OPA
DEVICES (D)
ENVIRONMENT ROLES (ER)
EA
ENVIRONMENT CONDITIONS (EC)
Assignment
Association

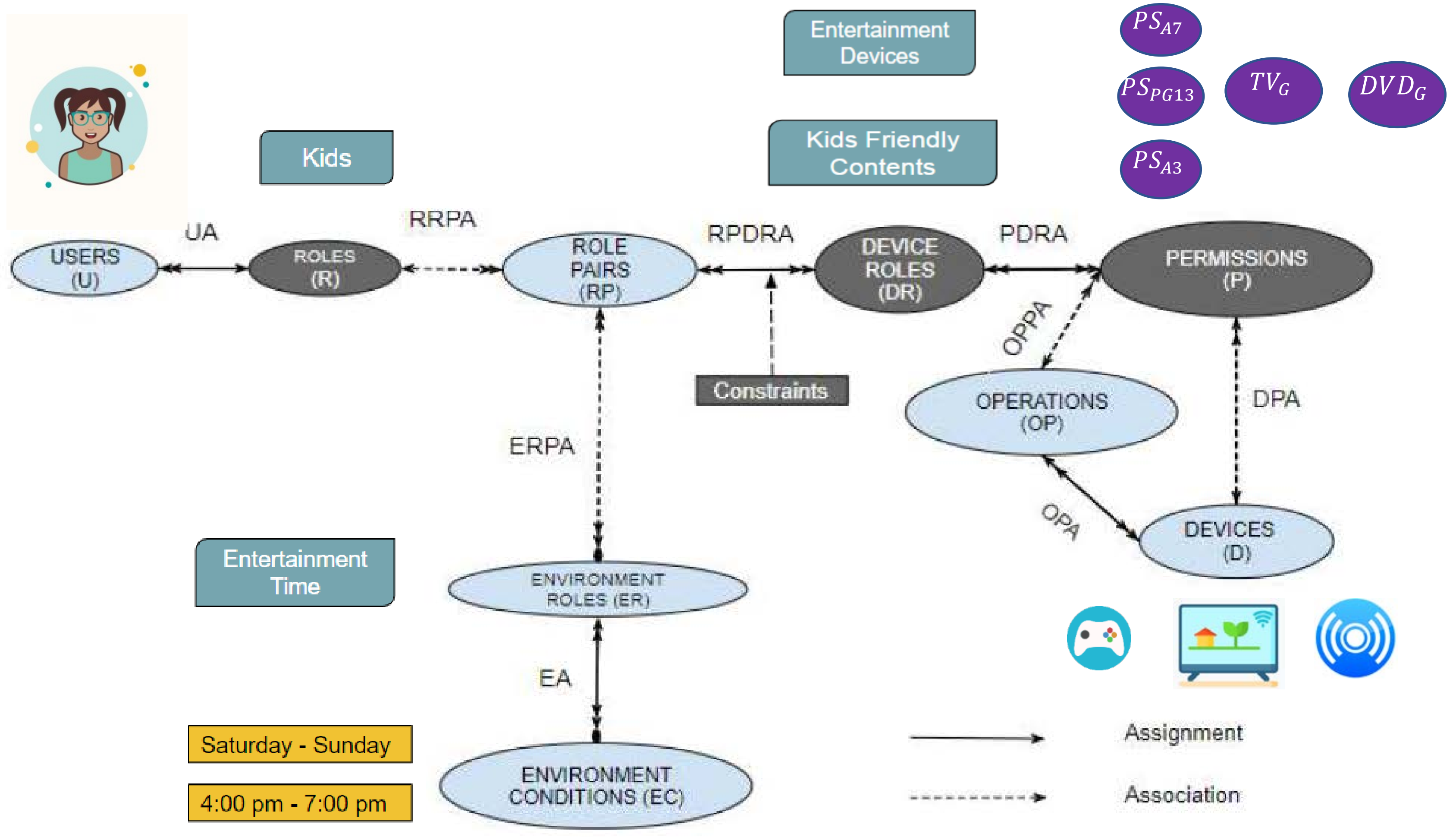
Use Case 1.B


PS_{A7}
PS_{PG13}
TV_G
DVD_G
PS_{A3}
Kids
Kids Friendly Contents
UA
RRPA
RPDRA
PDRA
USERS (U)
ROLES (R)
ROLE PAIRS (RP)
DEVICE ROLES (DR)
PERMISSIONS (P)
Constraints
OPPA
OPERATIONS (OP)
DPA
ERPA
OPA
DEVICES (D)
ENVIRONMENT ROLES (ER)
EA
ENVIRONMENT CONDITIONS (EC)
Assignment
Association

Use Case 1.B


PS_{A7}
PS_{PG13}
TV_G
DVD_G
PS_{A3}
Kids
Kids Friendly Contents
UA
RRPA
RPDRA
PDRA
USERS (U)
ROLES (R)
ROLE PAIRS (RP)
DEVICE ROLES (DR)
PERMISSIONS (P)
OPPA
Constraints
OPERATIONS (OP)
DPA
ERPA
OPA
DEVICES (D)
Entertainment Time
ENVIRONMENT ROLES (ER)
EA
Saturday - Sunday
4:00 pm - 7:00 pm
ENVIRONMENT CONDITIONS (EC)
Assignment
Association

Use Case 1.B



Entertainment Devices
PS_{A7}
PS_{PG13}
TV_G
DVD_G
PS_{A3}
Kids
Kids Friendly Contents
UA
RRPA
RPDRA
PDRA
USERS (U)
ROLES (R)
ROLE PAIRS (RP)
DEVICE ROLES (DR)
PERMISSIONS (P)
OPPA
Constraints
OPERATIONS (OP)
DPA
ERPA
OPA
DEVICES (D)
Entertainment Time
ENVIRONMENT ROLES (ER)
EA
Saturday - Sunday
4:00 pm - 7:00 pm
ENVIRONMENT CONDITIONS (EC)
Assignment
Association

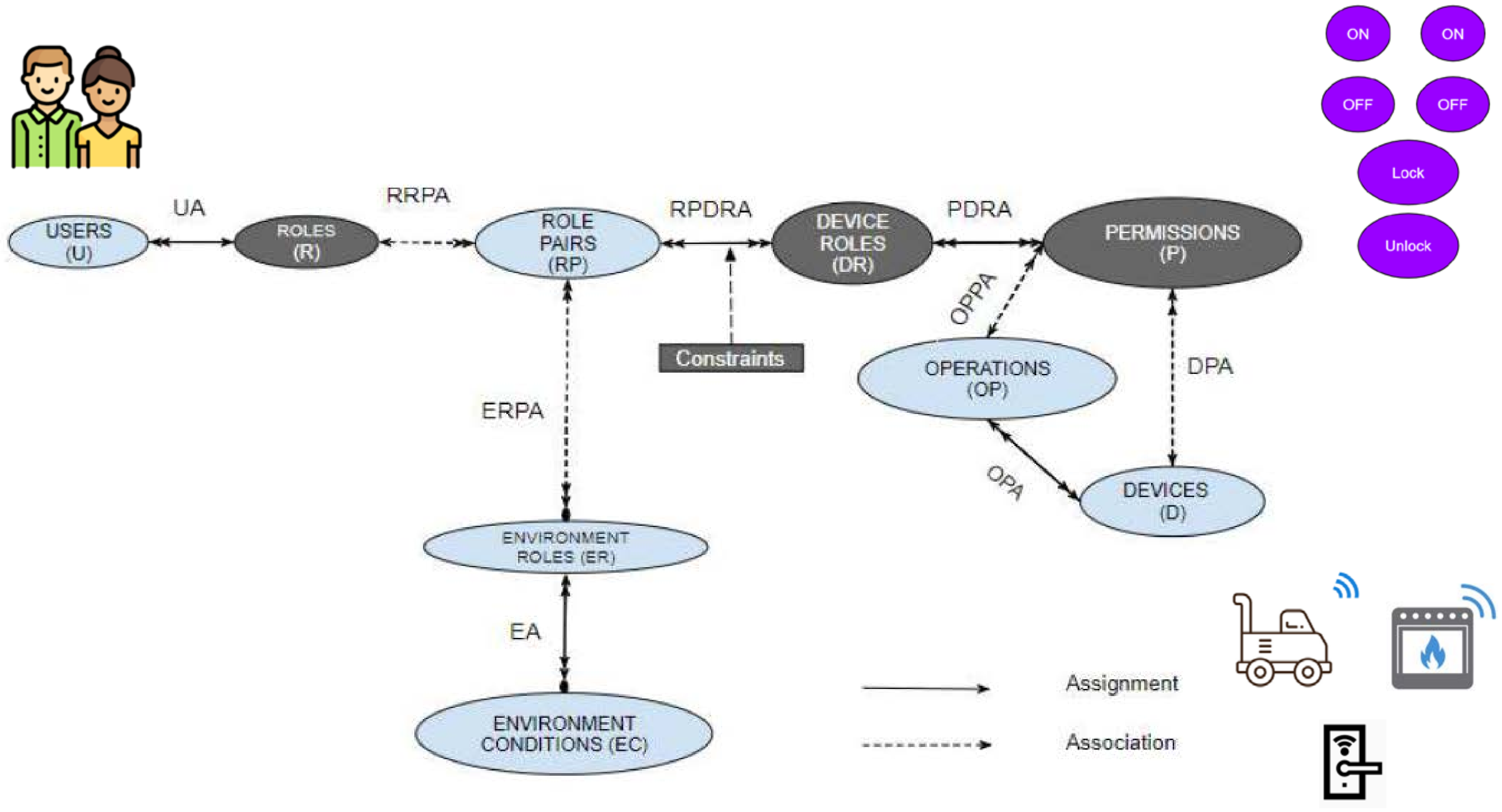
Use Case 2 Configuration in EGRBAC

$$
\begin{aligned}
&U = \{alex, bob, \ldots\},\ R = \{kids, parents, \ldots\}\\
&UA = \{(alex, kids), (bob, parents), \ldots\}\\
&D = \{DoorLock, Oven, LawnMawer, \ldots\}\\
&OP_{DoorLock} = \{Lock, Unlock, \ldots\}\\
&OP_{Oven} = \{On, Off, \ldots\}\\
&OP_{LawnMawer} = \{On, Off, \ldots\}\\
&OP = OP_{DoorLock} \cup OP_{Oven} \cup OP_{LawnMawer} \cup \ldots\\
&P_{DoorLock} = \{DoorLock_{Lock}, DoorLock_{Unlock}, \ldots\}\\
&P_{Oven} = \{Oven_{On}, Oven_{Off}, \ldots\}\\
&P_{LawnMawer} = \{LawnMawer_{On}, LawnMawer_{Off}, \ldots\}\\
&P = P_{DoorLock} \cup P_{Oven} \cup P_{LawnMawer} \cup \ldots\\
&DR = \{Dangerous\_Devices, \ldots\}\\
&PDRA = (\{DoorLock_{Lock}, DoorLock_{Unlock}\} \times\\
&\qquad\qquad \{Dangerous\_Devices\}) \cup\\
&\qquad (\{Oven_{On}, Oven_{Off}\} \times\\
&\qquad\qquad \{Dangerous\_Devices\}) \cup\\
&\qquad (\{LawnMawer_{On}, LawnMawer_{Off}\} \times\\
&\qquad\qquad \{Dangerous\_Devices\}) \cup \ldots
\end{aligned}
$$

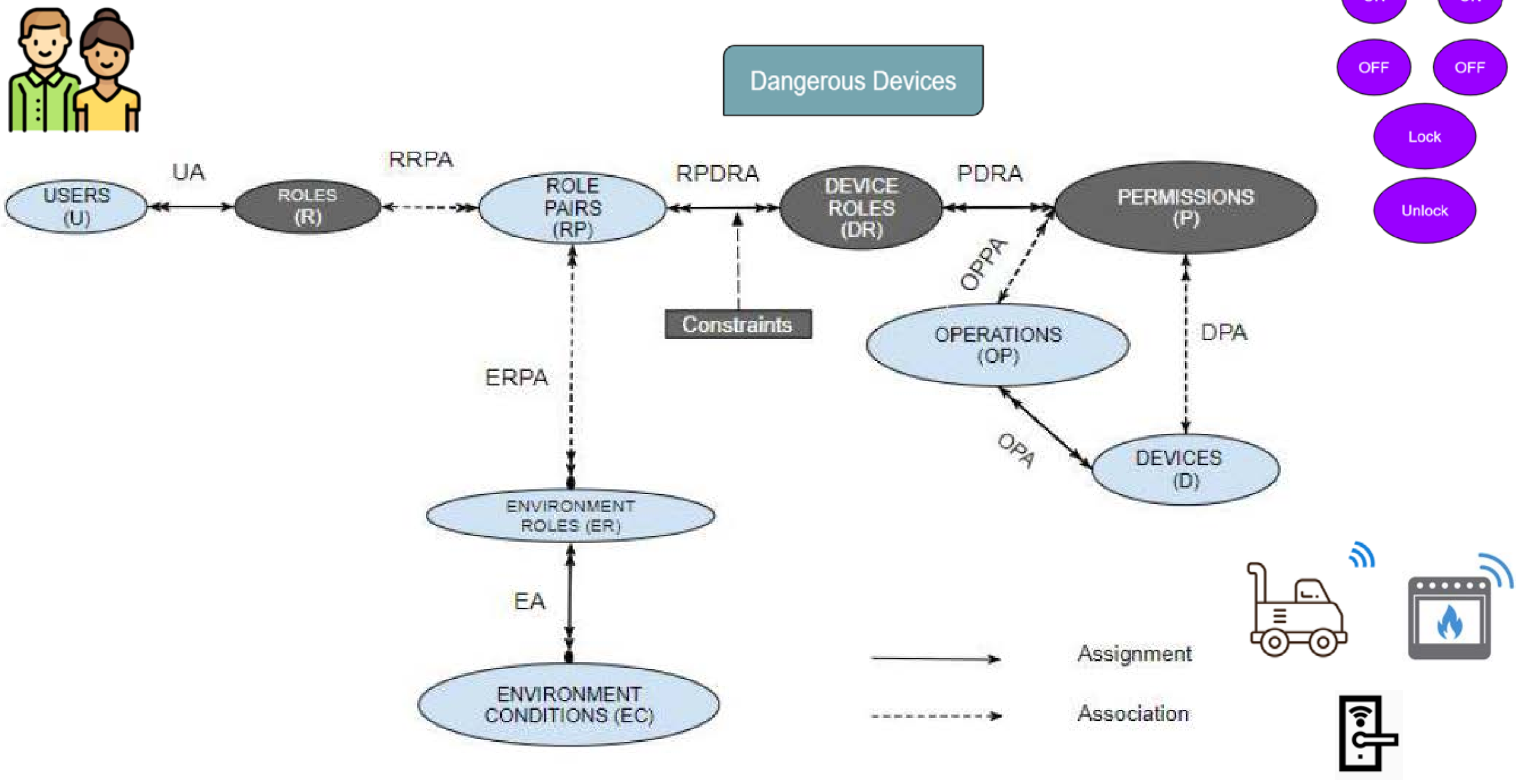
$$
\begin{aligned}
&EC = \{TRUE, \ldots\}\\
&ER = \{Any\_Time, \ldots\}\\
&EA = \{(TRUE, Any\_Time), \ldots\}
\end{aligned}
$$

$$
\begin{aligned}
&RP = \{(parents, \{Any\_Time\}), \ldots\}\\
&RPDRA = \{((parents, \{Any\_Time\}),\\
&\qquad\qquad Dangerous\_Devices), \ldots\}
\end{aligned}
$$

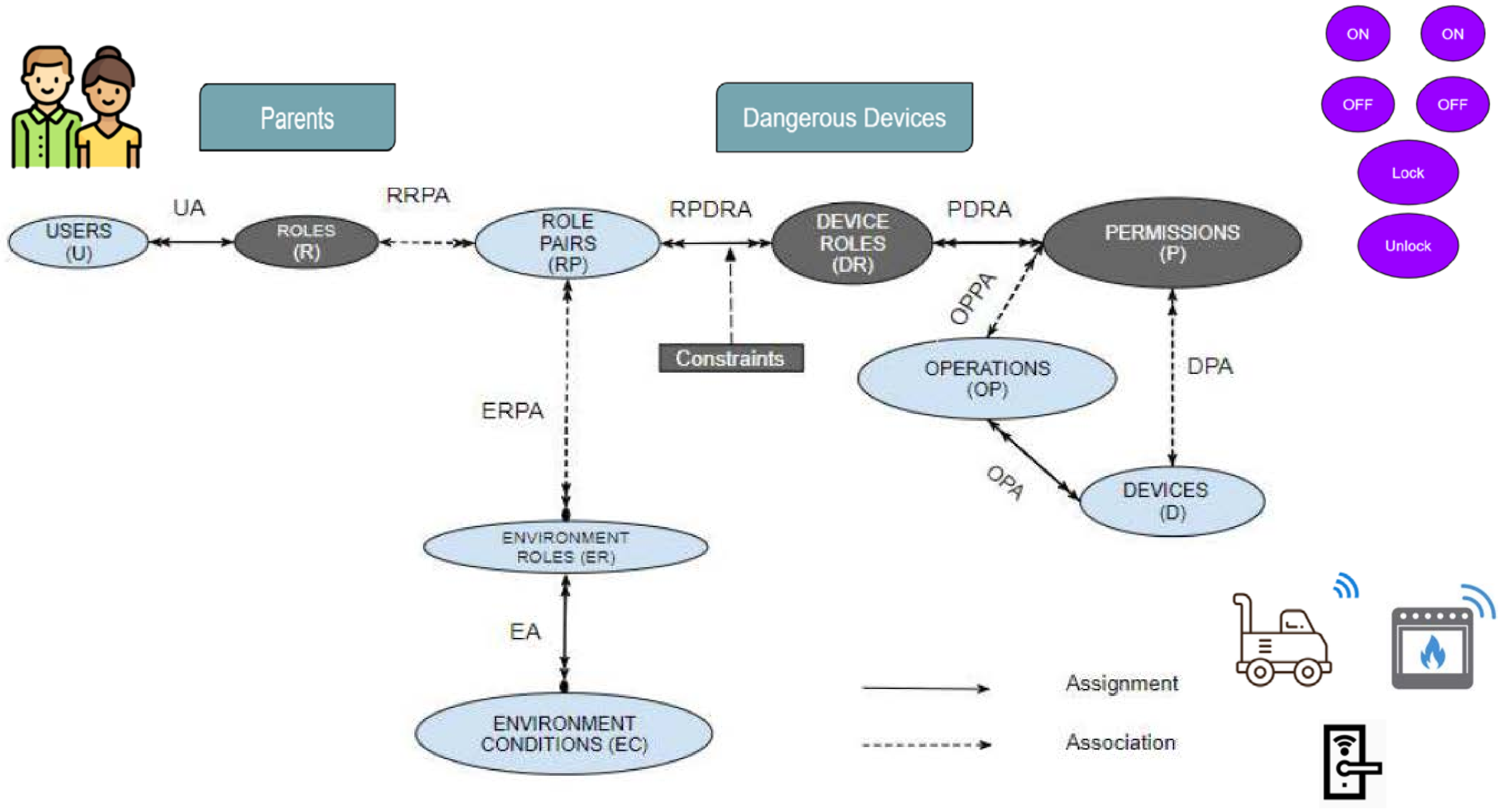
Use Case 2


ON
ON
OFF
OFF
Lock
Unlock
USERS (U)
UA
ROLES (R)
RRPA
ROLE PAIRS (RP)
RPDRA
DEVICE ROLES (DR)
PDRA
PERMISSIONS (P)
OPPA
Constraints
OPERATIONS (OP)
DPA
ERPA
OPA
DEVICES (D)
ENVIRONMENT ROLES (ER)
EA
ENVIRONMENT CONDITIONS (EC)
Assignment
Association

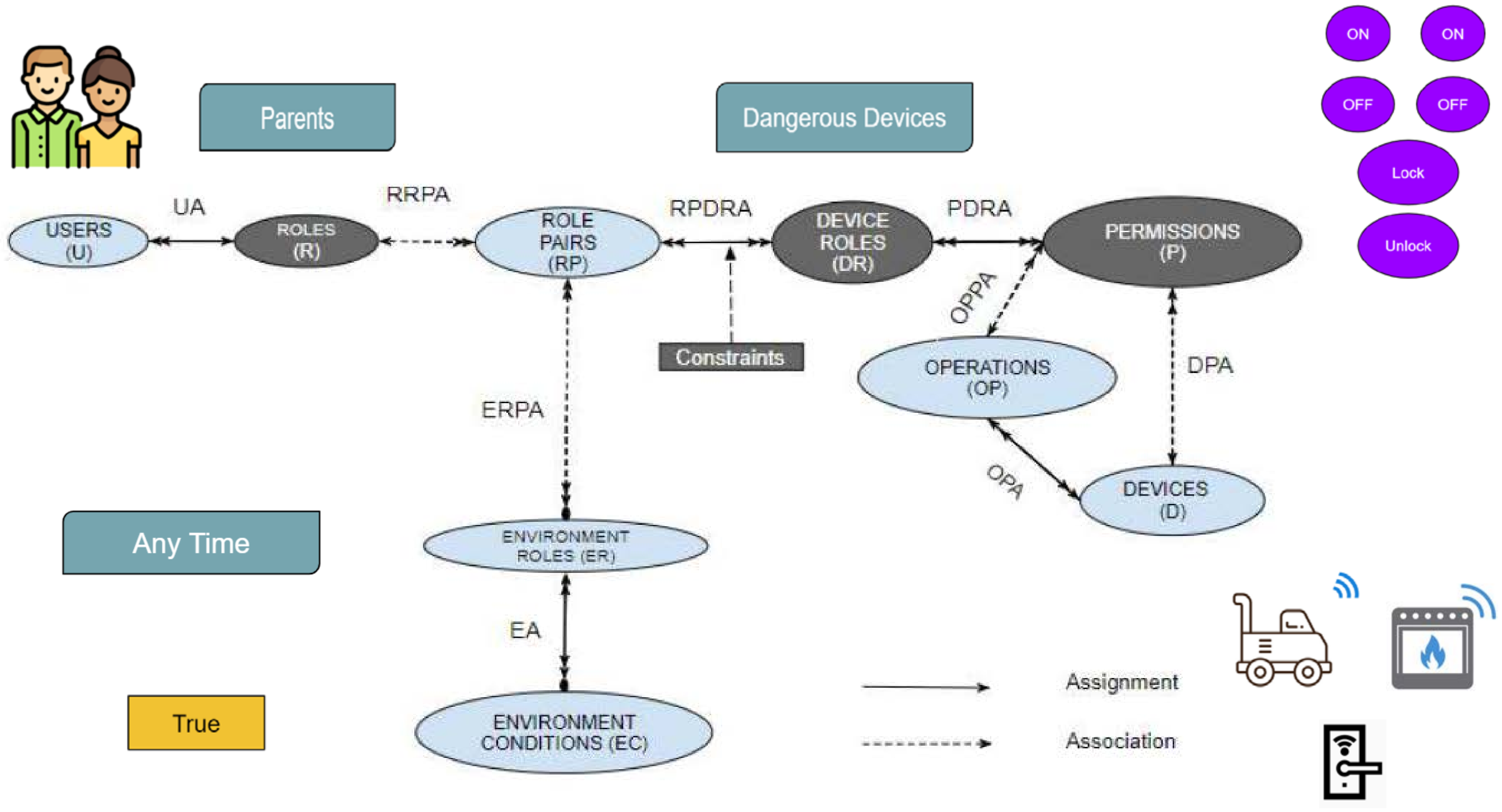
Use Case 2


ON
ON
OFF
OFF
Lock
Unlock
Dangerous Devices
USERS (U)
UA
ROLES (R)
RRPA
ROLE PAIRS (RP)
RPDRA
DEVICE ROLES (DR)
PDRA
PERMISSIONS (P)
Constraints
OPPA
OPERATIONS (OP)
DPA
ERPA
OPA
DEVICES (D)
ENVIRONMENT ROLES (ER)
EA
ENVIRONMENT CONDITIONS (EC)
Assignment
Association

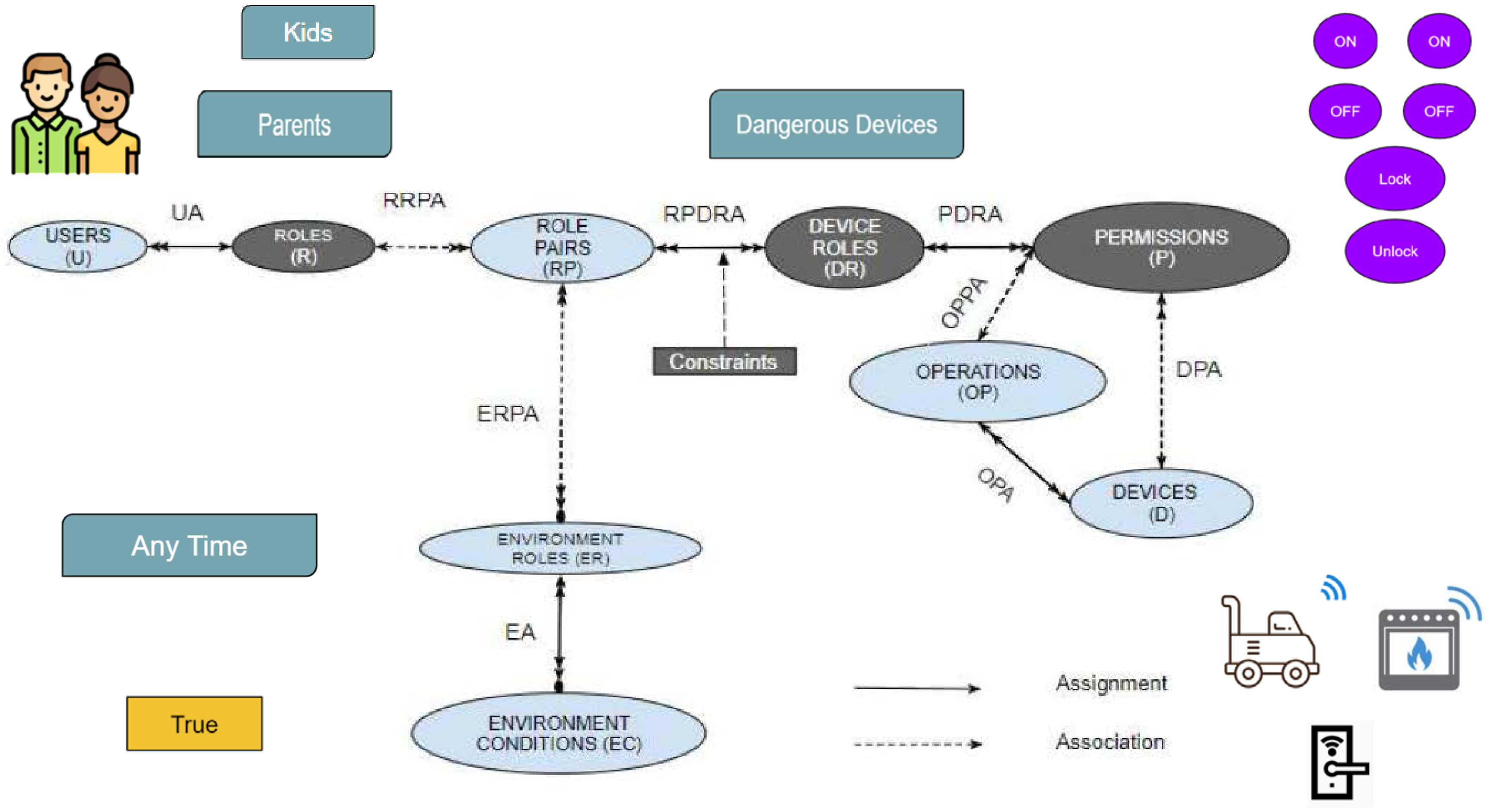
Use Case 2


ON
ON
OFF
OFF
Lock
Unlock
Parents
Dangerous Devices
USERS (U)
UA
ROLES (R)
RRPA
ROLE PAIRS (RP)
RPDRA
DEVICE ROLES (DR)
PDRA
PERMISSIONS (P)
OPPA
Constraints
OPERATIONS (OP)
DPA
ERPA
OPA
DEVICES (D)
ENVIRONMENT ROLES (ER)
EA
ENVIRONMENT CONDITIONS (EC)
Assignment
Association

Use Case 2


ON
ON
OFF
OFF
Lock
Unlock
Parents
Dangerous Devices
UA
RRPA
RPDRA
PDRA
USERS (U)
ROLES (R)
ROLE PAIRS (RP)
DEVICE ROLES (DR)
PERMISSIONS (P)
OPPA
Constraints
OPERATIONS (OP)
DPA
ERPA
OPA
DEVICES (D)
Any Time
ENVIRONMENT ROLES (ER)
EA
True
ENVIRONMENT CONDITIONS (EC)
Assignment
Association

Use Case 2


Kids
Parents
Dangerous Devices
ON
ON
OFF
OFF
Lock
Unlock
USERS (U)
UA
ROLES (R)
RRPA
ROLE PAIRS (RP)
RPDRA
DEVICE ROLES (DR)
PDRA
PERMISSIONS (P)
Constraints
OPPA
OPERATIONS (OP)
DPA
ERPA
OPA
DEVICES (D)
Any Time
ENVIRONMENT ROLES (ER)
EA
True
ENVIRONMENT CONDITIONS (EC)
Assignment
Association

Constraints

- A constraint is an invariant that must always be maintained.

- In use case B, we gave parents permission to access dangerous devices, what if we want prevent future assignment of dangerous devices to the role Kids (which could happen inadvertently by the homeowner)?

- To prevent such situations, EGRBAC incorporates constraints that forbid assigning specific permissions to specific roles. Formally:

  $Constraints \subseteq 2^P \times 2^R$ constitute a many to many subset of permissions to subset of roles relation. Each c $= (P_i, R_j) \in Constraints$ specifies the following invariant:

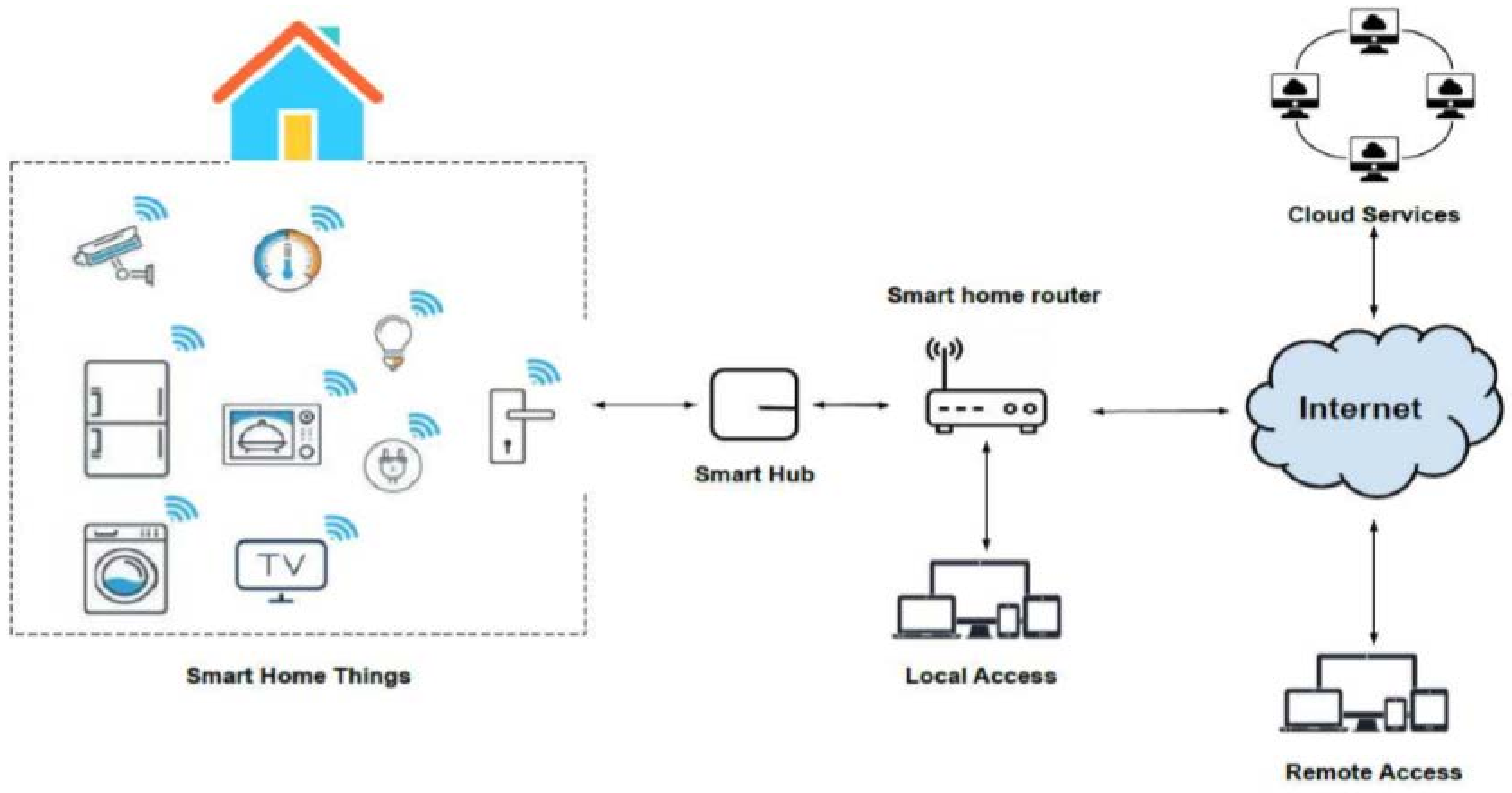
  For every $p_m \in P_i$ and every $r_n \in R_j$:

  $$\forall_{rp_p \in RP,\, dr_q \in DR} \ \left(rp_p, dr_q\right) \in RPDRA \quad \Rightarrow \quad \left(p_m, dr_q\right) \notin PDRA \quad \vee \quad (rp_p.r \neq r_n)$$

The Institute for Cyber Security

Enforcement Architecture

Center for Security and Privacy Enhanced Cloud Computing



- There are two types of requests: A- Local requests. B- Remote requests.

Proof of Concept Implementation

- We simulated our model using AWS IoT service.
- An AWS account is required to configure and deploy the AWS IoT service known as Greengrass.
- The Greengrass SDK extends cloud capabilities to the edge, which in our case is the smart home. This enables devices to collect and analyze data closer to the source of information, react autonomously to local events, and communicate securely with each other on local networks.
- In our system Greengrass serves as a smart hub and a policy engine. It runs on a dedicated virtual machine.

Proof of Concept Implementation

- Through AWS IoT management console, one virtual object (aka digital shadow) is created for each physical device and the two are cryptographically linked via digital certificates with attached authorization policies.

- Each simulated device is run on a separate virtual machine. These devices use MQTT protocol to communicate to the AWS IoT service with TLS security.

- To enforce EGRBAC, we utilized two Json files UserRoleAssignment.json and policy.json.

- UserRoleAssignment.json defines the assignments of users to their corresponding roles while policy.json defines all other EGRBAC components.

Proof of Concept Implementation

- We also utilized the lambda function service in AWS IoT platform to:
  1. Receive the operation requests of users to access the smart devices in the house.
  2. Analyze each request according to the content of the policy.json and UserRoleAssignment.json files.
  3. Trigger the desired actions on the corresponding simulated devices.

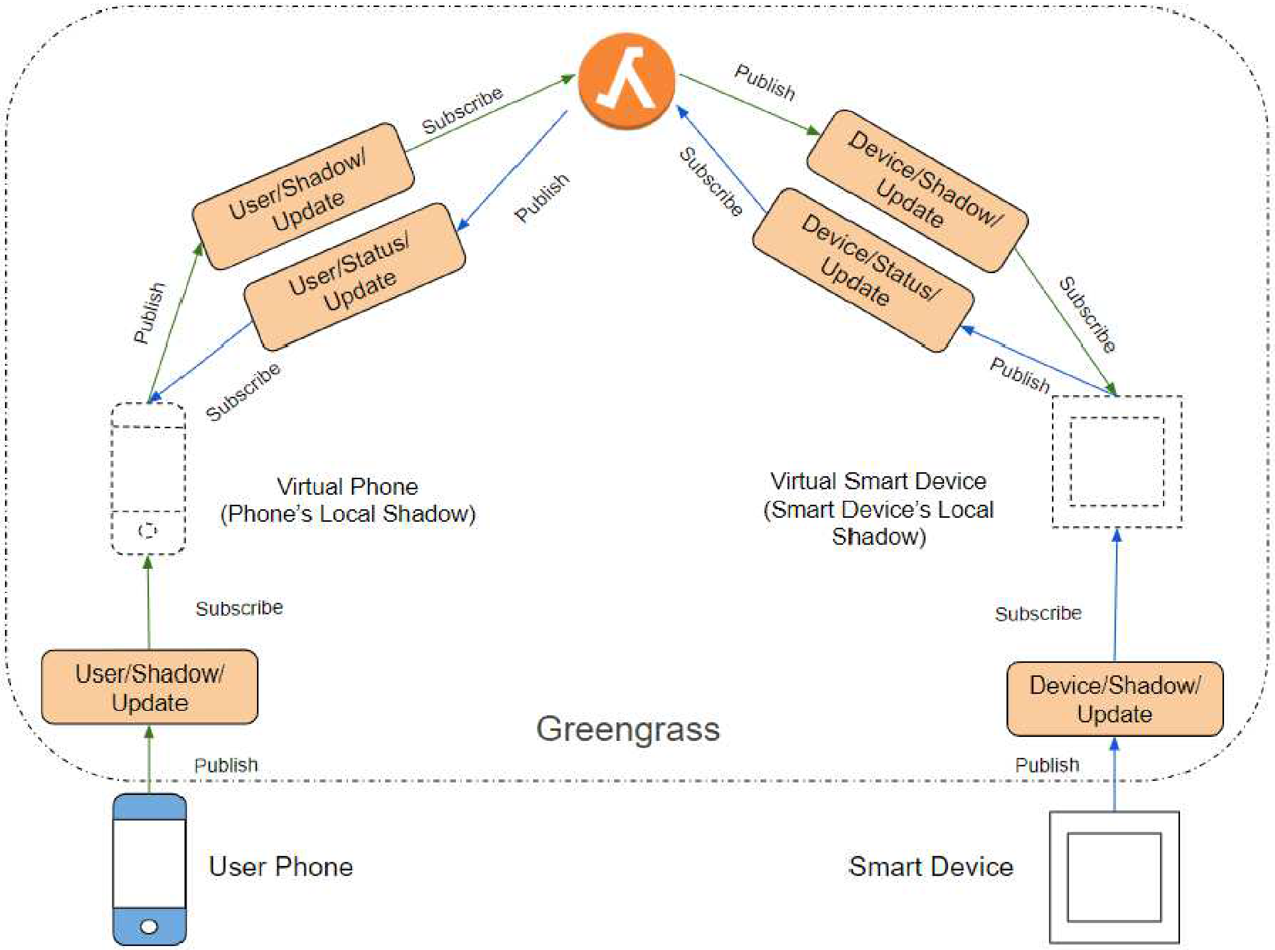
- We used a long-lived lambda function with 128 MB Memory Limit, 30 second timeout.

- The lambda function, the UserRoleAssignment.json file, and the policy.json file are all configured in the Greengrass

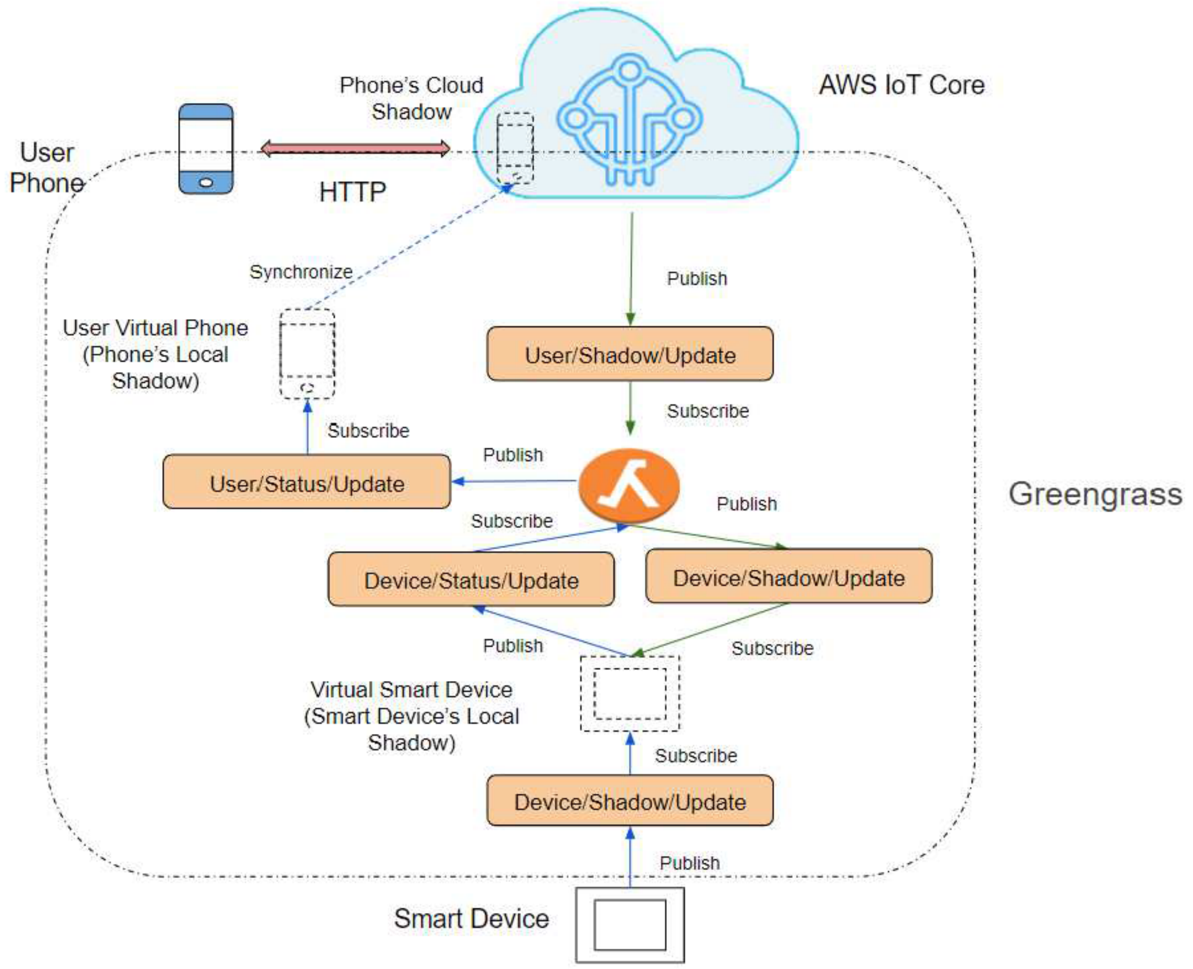
I·C·S
The Institute for Cyber Security

Local Request Processing

C·SPECC
Center for Security and Privacy Enhanced Cloud Computing



Remote Request Processing



Our Implemented Use Case

$$
\begin{aligned}
&U = \{alex, bob, susan, james, julia\}\\
&R = \{kids, parents, babySitters, guests, neighbors\}\\
&UA = \{(alex, kids), (bob, parents), (susan, babySitters),\\
&\qquad (james, guests), (julia, neighbors)\}\\
&D = \{DoorLock, Oven, TV, DVD, Playstation\}\\
&OP_{DoorLock} = \{Lock, Unlock\}\\
&OP_{Oven} = \{On, Off\}\\
&OP_{TV} = \{On, Off\}\\
&OP_{DVD} = \{On, Off\}\\
&OP_{Playstation} = \{On, Off\}\\
&OP = OP_{DoorLock} \cup OP_{Oven} \cup OP_{TV} \cup OP_{DVD} \cup OP_{Playstation}\\
&P_{DoorLock} = \{DoorLock_{Lock}, DoorLock_{Unlock}\}\\
&P_{Oven} = \{Oven_{On}, Oven_{Off}\}\\
&P_{TV} = \{TV_{On}, TV_{Off}\}\\
&P_{DVD} = \{DVD_{On}, DVD_{Off}\}\\
&P_{Playstation} = \{PS_{On}, PS_{Off}\}\\
&P = P_{DoorLock} \cup P_{Oven} \cup P_{TV} \cup P_{DVD} \cup P_{Playstation}\\
&DR = \{Dangerous\_Devices, Entertainment\_Devices\}\\
&PDRA = (\{DoorLock_{Lock}, DoorLock_{Unlock}\} \times\\
&\qquad\qquad \{Dangerous\_Devices\}) \cup\\
&\qquad (\{Oven_{On}, Oven_{Off}\} \times\\
&\qquad\qquad \{Dangerous\_Devices\}) \cup\\
&\qquad (P_{TV} \times \{Entertainment\_Devices\}) \cup\\
&\qquad (P_{DVD} \times \{Entertainment\_Devices\}) \cup\\
&\qquad (P_{Playstation} \times \{Entertainment\_Devices\})
\end{aligned}
$$

$$
\begin{aligned}
&EC = \{weekends, evenings, TRUE\}\\
&ER = \{Entertainment\_Time, Any\_Time\}\\
&EA = \{(\{weekends, evenings\}, Entertainment\_Time),\\
&\qquad (TRUE, Any\_Time)\}
\end{aligned}
$$

$$
\begin{aligned}
&RP = \{(kids, \{Entertainment\_Time\}), (parents, \{Any\_Time\}),\\
&\qquad (babySitters, \{Any\_Time\}), (guests, \{Any\_Time\}),\\
&\qquad (neighbors, \{Any\_Time\})\}
\end{aligned}
$$

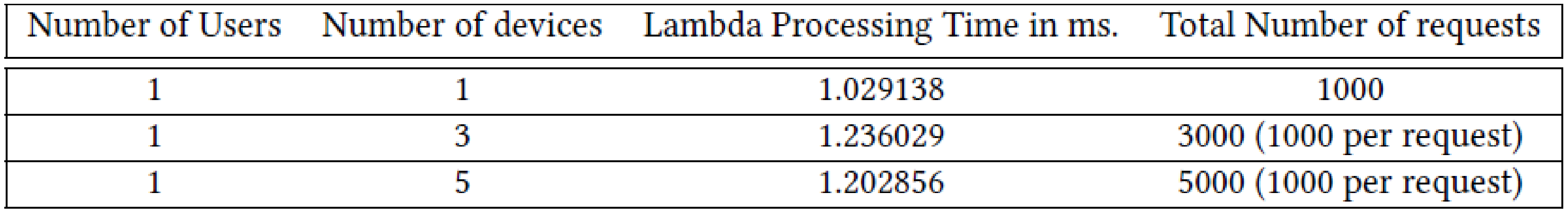
$$
\begin{aligned}
&RPDRA = \{((parents, \{Any\_Time\}), Dangerous\_Devices),\\
&\qquad ((kids, \{Entertainment\_Time\}), Entertainment\_Devices),\\
&\qquad ((parents, \{Any\_Time\}), Entertainment\_Devices),\\
&\qquad ((babySitters, \{Any\_Time\}), Entertainment\_Devices),\\
&\qquad ((guests, \{Any\_Time\}), Entertainment\_Devices),\\
&\qquad ((neighbors, \{Any\_Time\}), Entertainment\_Devices)\}
\end{aligned}
$$

Performance Results

- We executed multiple test cases to measure the processing time in different scenarios.
- we measured the average lambda function execution time under different conditions.
- Overall, our model is functional, and can easily applied. Moreover, we can notice that the execution time is generally low.

Performance Results

One User Sending Requests to Multiple Devices



Number of Users	Number of devices	Lambda Processing Time in ms.	Total Number of requests
1	1	1.029138	1000
1	3	1.236029	3000 (1000 per request)
1	5	1.202856	5000 (1000 per request)

- The first row shows the average time when the parent Bob requests to unlock the door lock. The second row shows the average time when Bob requests to turn on the oven, the TV, and the DVD at the same time. The third row shows the average time when Bob requests to unlock the door lock, turn on the oven, the TV, the DVD, and the playStation at the same time.

- All the requests were approved as they were supposed to according to our configured policies.

Performance Results

Multiple Concurrent Instances of One User Sending Requests to One Device

Number of Users	Number of devices	Lambda Processing Time in ms.	Total Number of requests
1	1	1.029138	1000
3	3	1.796938	3000 (1000 per request)
5	5	2.833097	5000 (1000 per request)

- The first row shows the average time when the parent Bob requests to unlock the door lock. The second row shows the average time when Bob requests to unlock the door lock, the kid Alex requests to turn on the oven, and the babysitter Susan requests to turn on the TV at the same time. The third row shows the average time when the three access requests tested in the second row are carried again in addition to, the guest James requests to turn on the DVD, and the neighbor Julia requests to turn on the PlayStation.

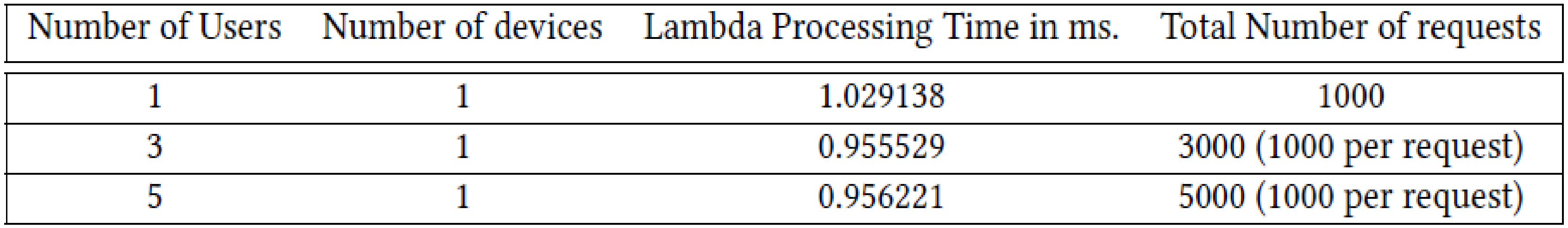
Performance Results

- The system responded correctly where all the requests were approved except for when the kid Alex requests to turn on the oven.

- We can conclude that when the number of requests for different users and different devices (one user per device) increases, the lambda processing time also increases.

Performance Results

Multiple Users Sending Requests to One Device



Number of Users	Number of devices	Lambda Processing Time in ms.	Total Number of requests
1	1	1.029138	1000
3	1	0.955529	3000 (1000 per request)
5	1	0.956221	5000 (1000 per request)

- This table, shows the average lambda function execution time when we send multiple requests from multiple users to one device at the same time. The first, second, and third rows show the average time when the parent (1 user), the parent the kid and the babysitter (3 users), or the parent the kid the babysitter the guest and the neighbor (5 users) respectively all request to unlock the door at the same time.

Performance Results

- The system responded correctly where all the requests were denied except for when the parent Bob requests to unlock the door lock.

- Here, we can see that the average lambda processing time decrease when we have more denies.

- This result is expected since In order to approve a request the lambda function need to verify each condition in the authorization predicate. On the other hand, if only one of the authorization predicate conditions is violated the lambda function will deny the request without the need to check the rest of the authorization predicate.

- To conclude, our system takes more time when approving a request than when denying it.

Conclusion

- We propose the EGRBAC access control model for smart home IoT.

- Our model's main goal is to ensure that legitimate users are only permitted to use the devices which they are allowed to access under the appropriate conditions.

- It is a dynamic, fine-grained model that grants access based on the specific permission required rather than at device granularity.

- In the future we are planning, to develop a family (or series) of models ranging from relatively simple and complete to incorporating increasingly sophisticated and comprehensive features.