Secure Cloud Assisted Smart Cars

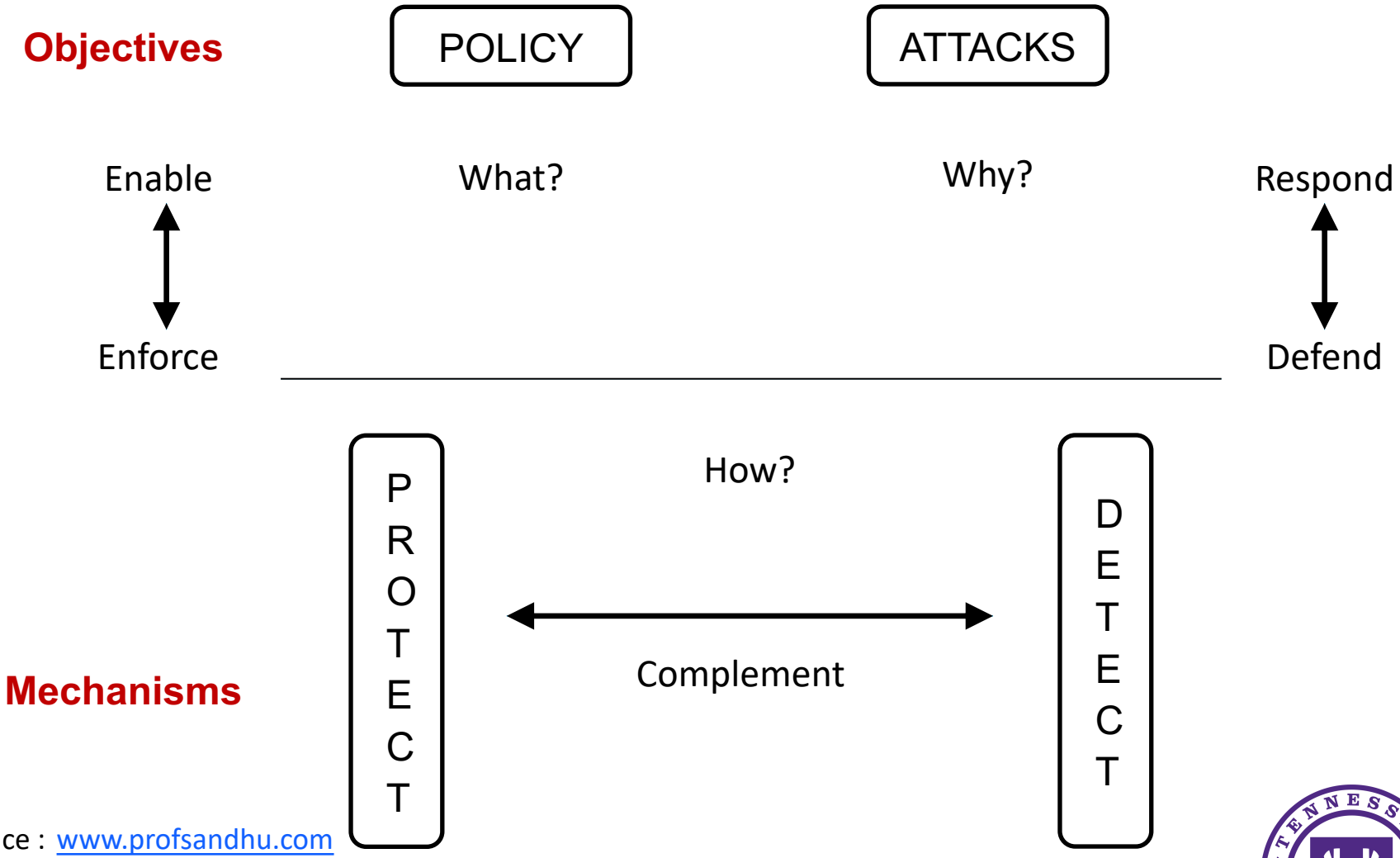
Maanak Gupta
Assistant Professor
Department of Computer Science
Tennessee Technological University

CAE Forum

November 13, 2019

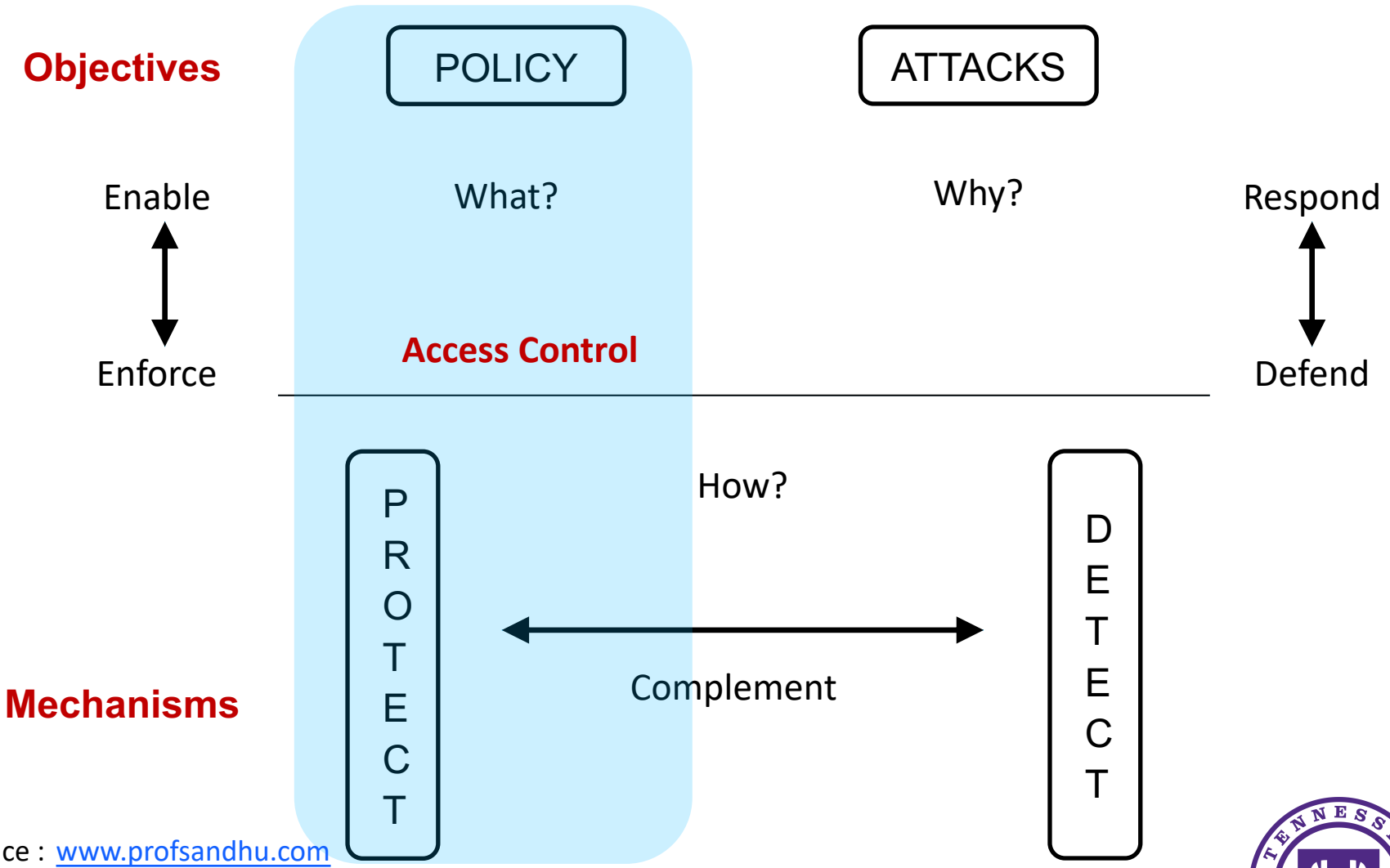
mgupta@tntech.edu
www.maanakgupta.com

Cyber Security Landscape



Reference : www.profsandhu.com

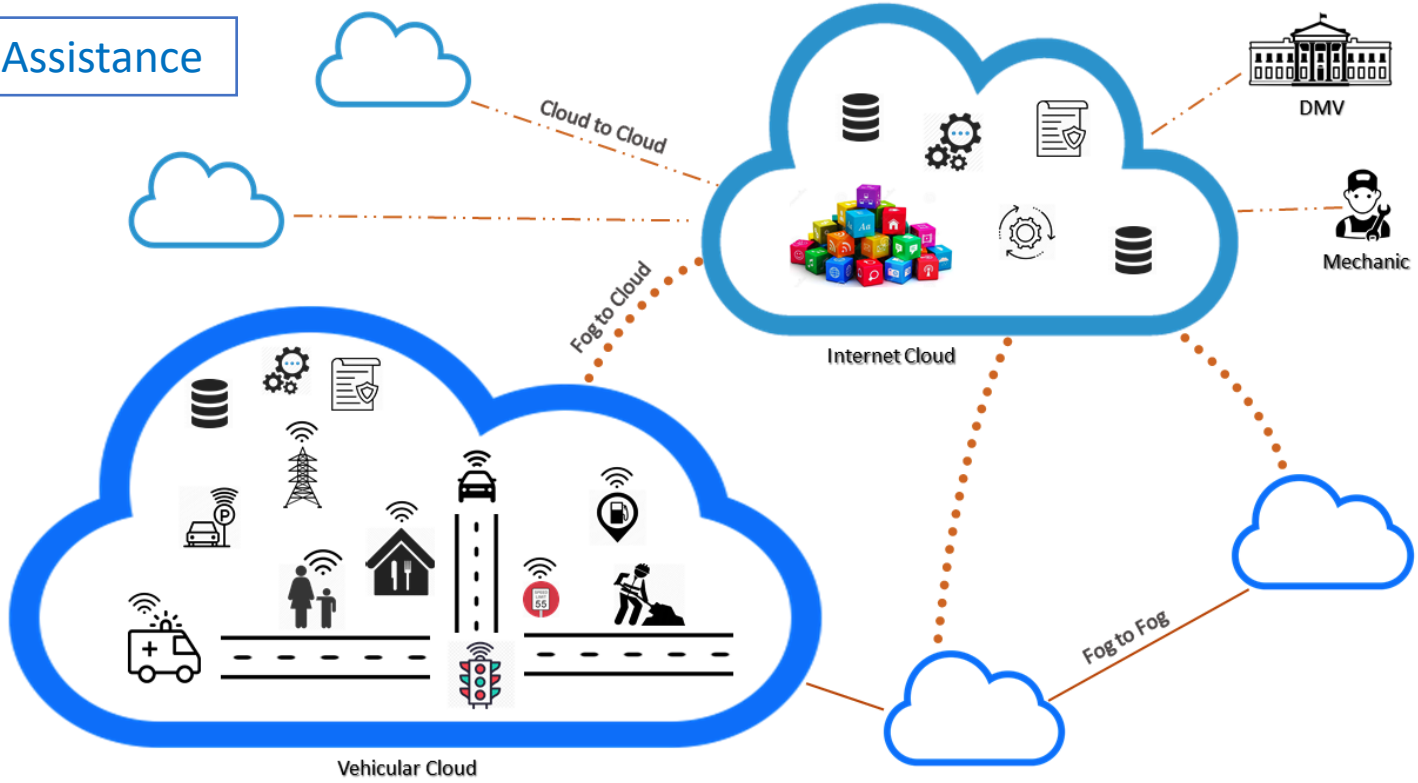
Cyber Security Landscape



Reference : www.profsandhu.com

Smart Cars Ecosystem

Safety and Assistance

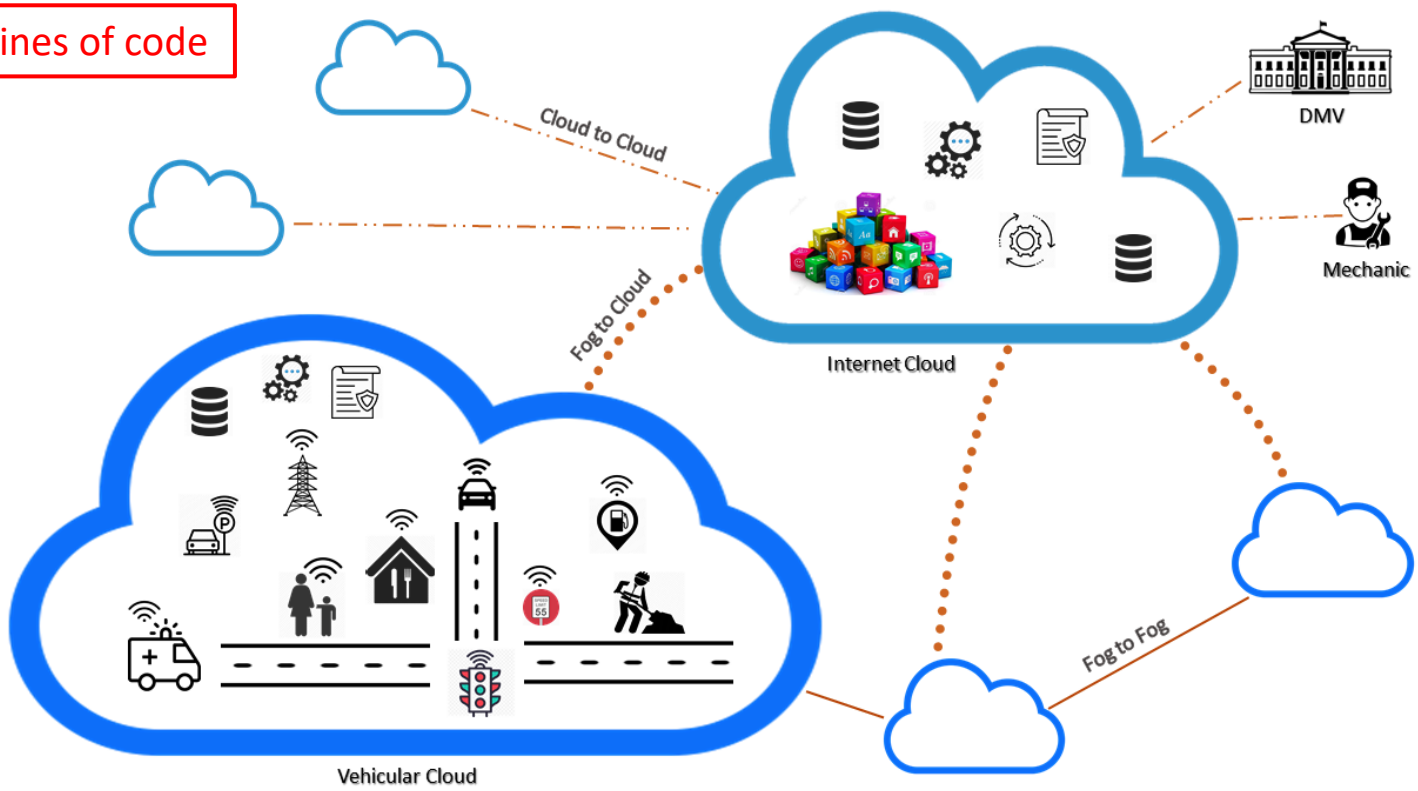


Information and Entertainment

High Mobility, Location Centric
Time Sensitive, Dynamic Pairing
Multiple Fog/Cloud Infrastructures

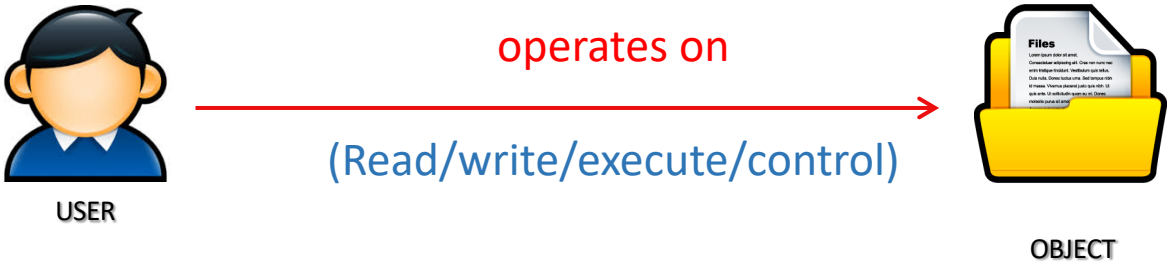
No More Isolated.!

100 million lines of code



Software Reliance , Broad Attack Surface, Untrusted Entities

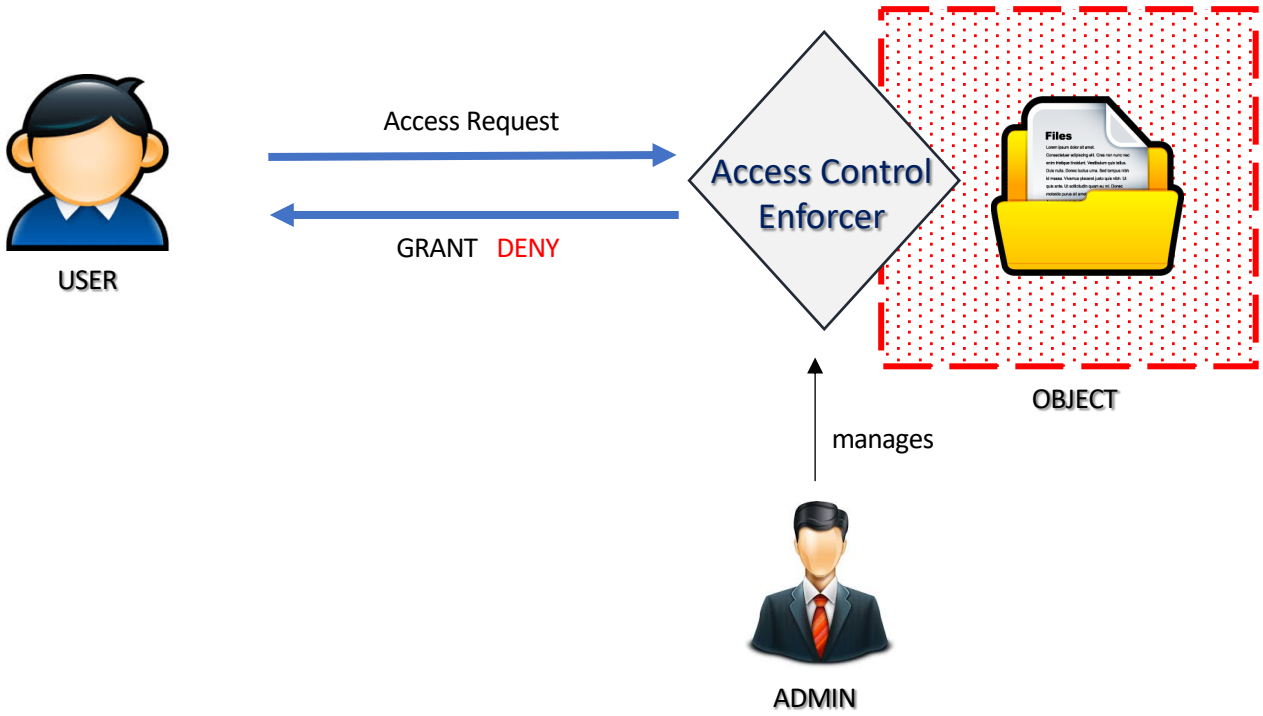
The Perfect World.!



I **TRUST** my users.
Everything is Secure. !!

Confidentiality
Integrity
Availability

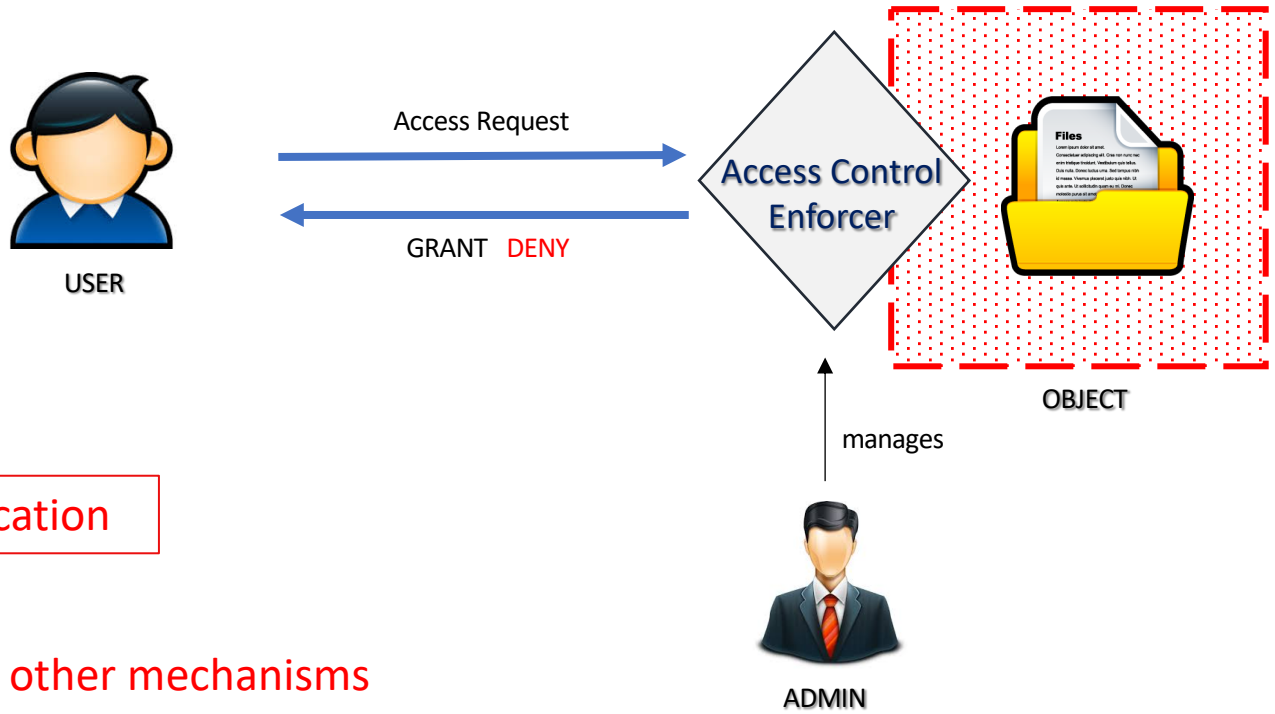
Access Control Mechanisms



A user [U] is allowed to perform an operation [OP] on an object [OB] if security policy [P] is satisfied.

Access Control Mechanisms

PROTECTION



Post Authentication

Complements other mechanisms

A user [U] is allowed to perform an operation [OP] on an object [OB] if security policy [P] is satisfied.

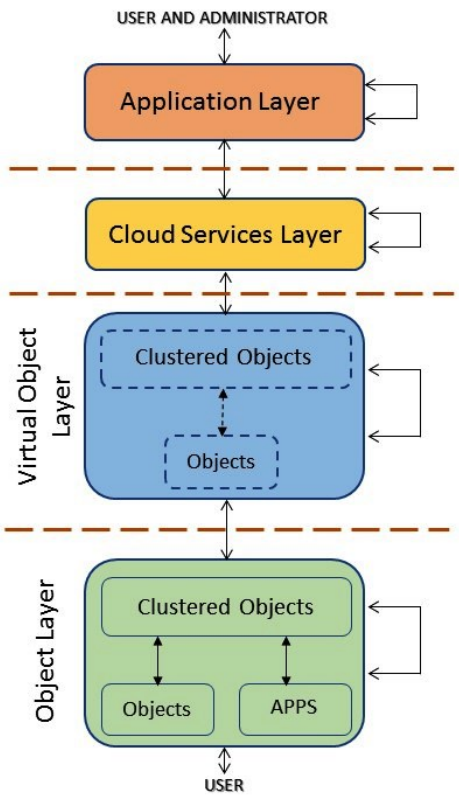
Attribute Based Access Control

- Three Dominant Models: DAC, MAC and RBAC.
- ABAC: Decision based on the attributes of entities
- Attributes are name value pair: age (Alice) → 29
- Core entities in ABAC include:
  - Users
  - Objects
  - Environment or Context
  - Operations

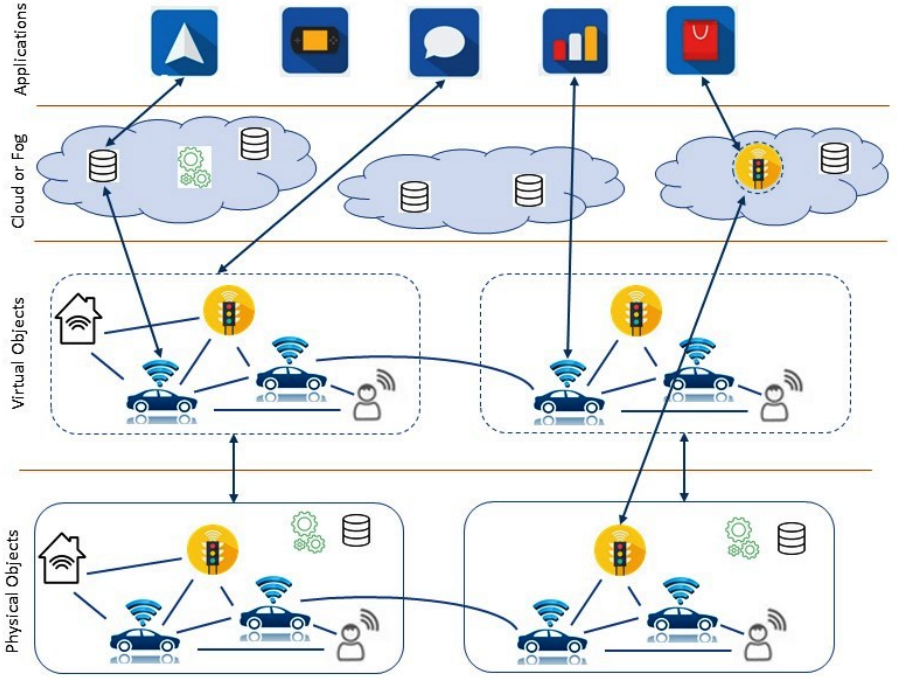
  } **Attributes**

- Authorization Policies: determine rights just in time
  - retrieve attributes of relevant entities in request
- Enhance flexibility and fine grained access control
- NIST Guidelines to ABAC

Extended Access Control Oriented Architecture

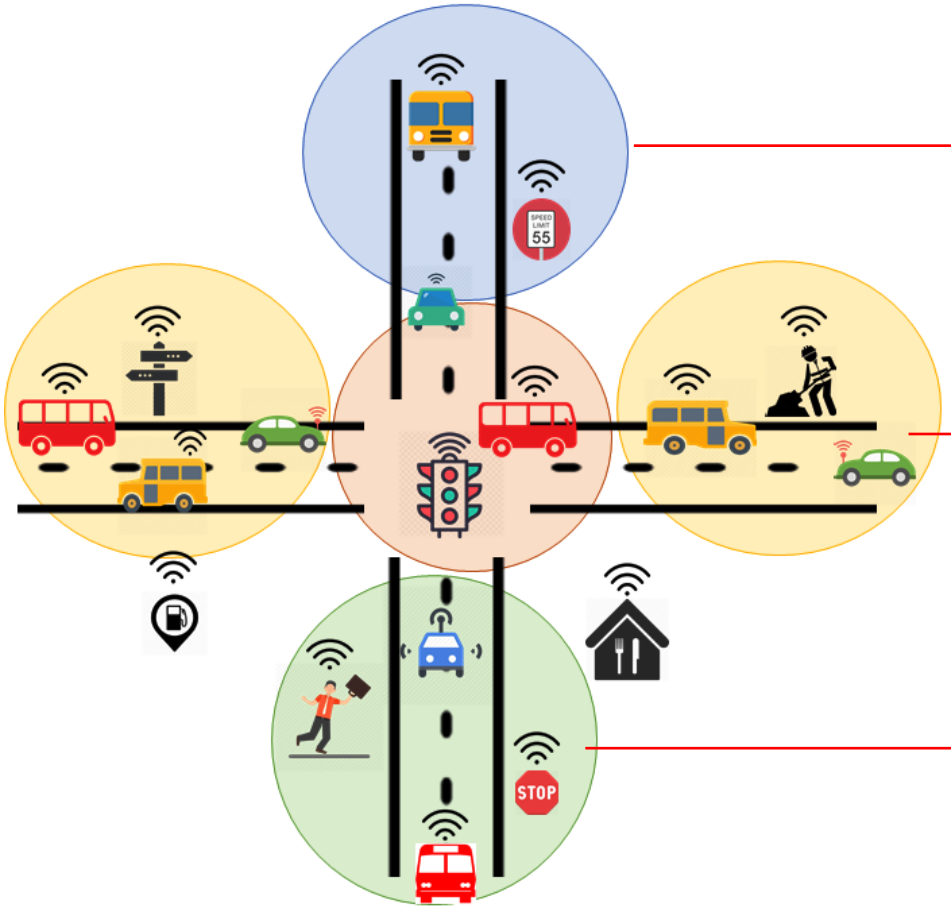


E-ACO architecture



Vehicular IoT components in architecture

Attributes and Alerts



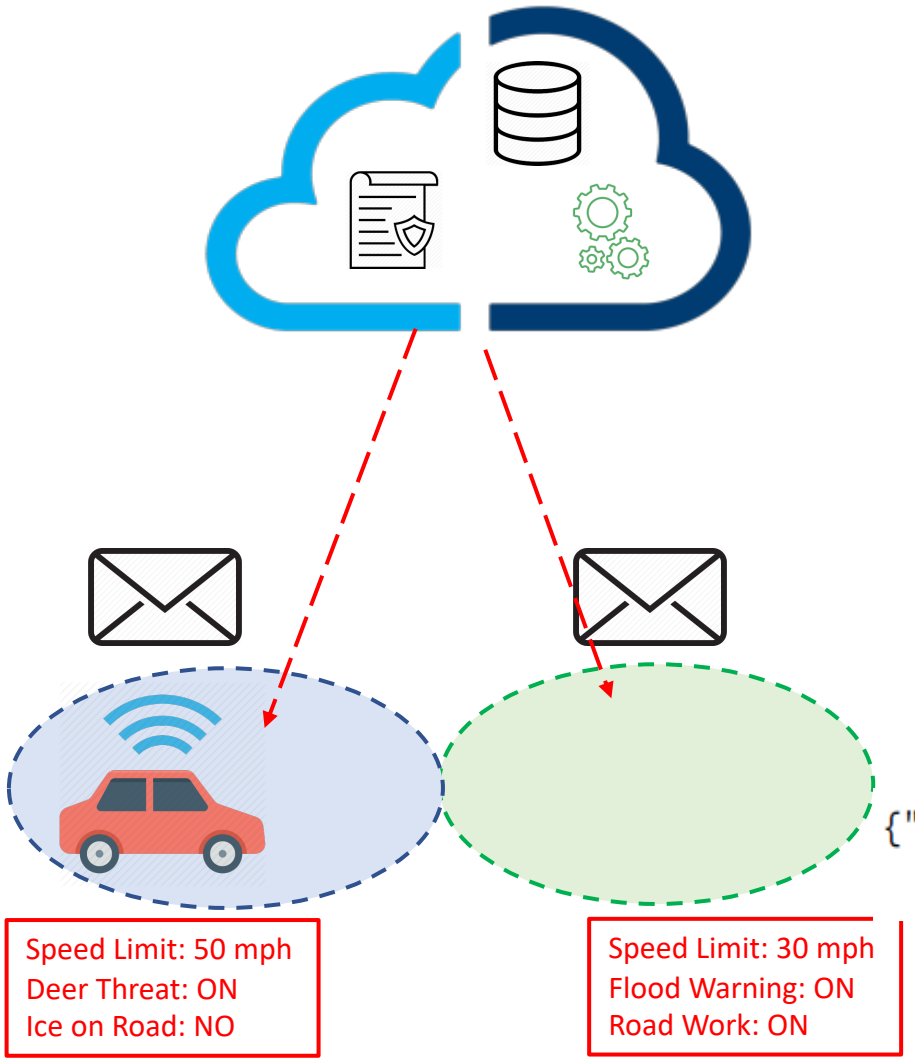
Speed Limit: 50 mph
Deer Threat: ON
Ice on Road: NO

Speed Limit: 30 mph
Flood Warning: ON
Road Work: ON

Speed Limit: 20 mph
School Zone: ON
Amber Alert: ABC123

Vehicle moves and are assigned to different groups and inherits their attributes/alerts.

Using Location Groups



Administrative Questions:

- How the attributes or alerts of groups are updated?
- How are moving entities assigned to groups?
- How groups hierarchy is created?

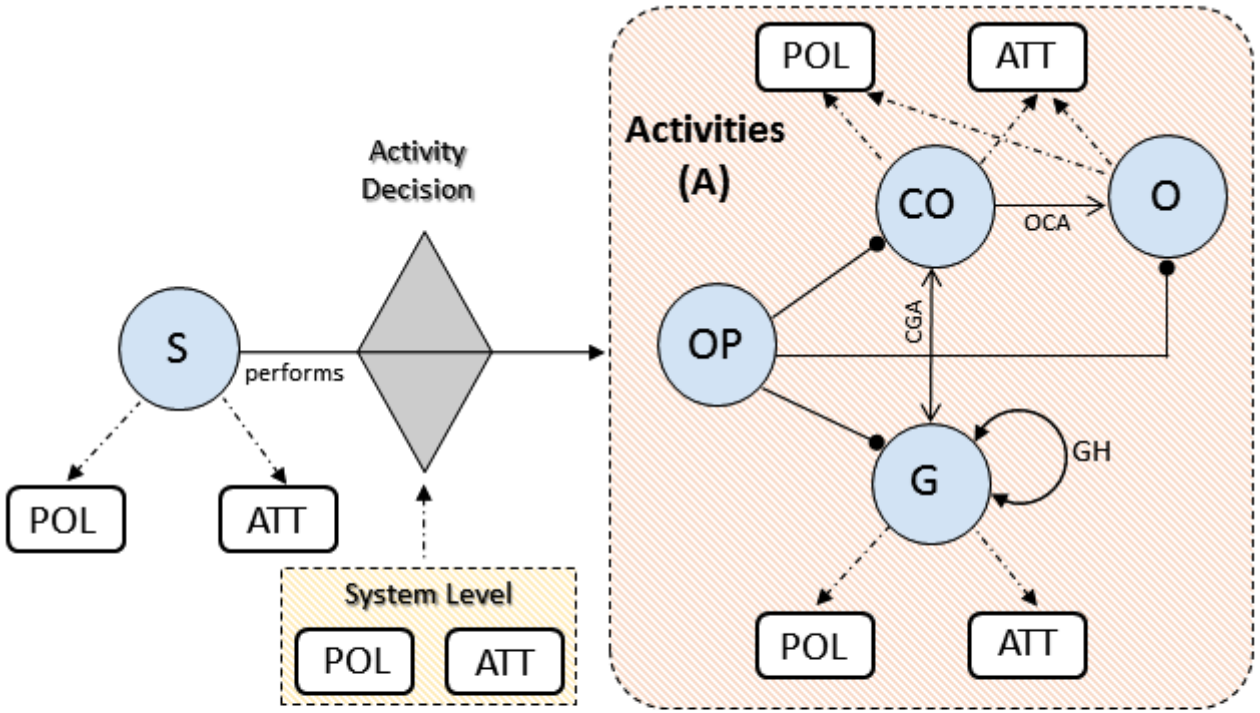
Operational Questions:

- How attributes and groups are used to provide security?
- How user privacy preferences are considered?

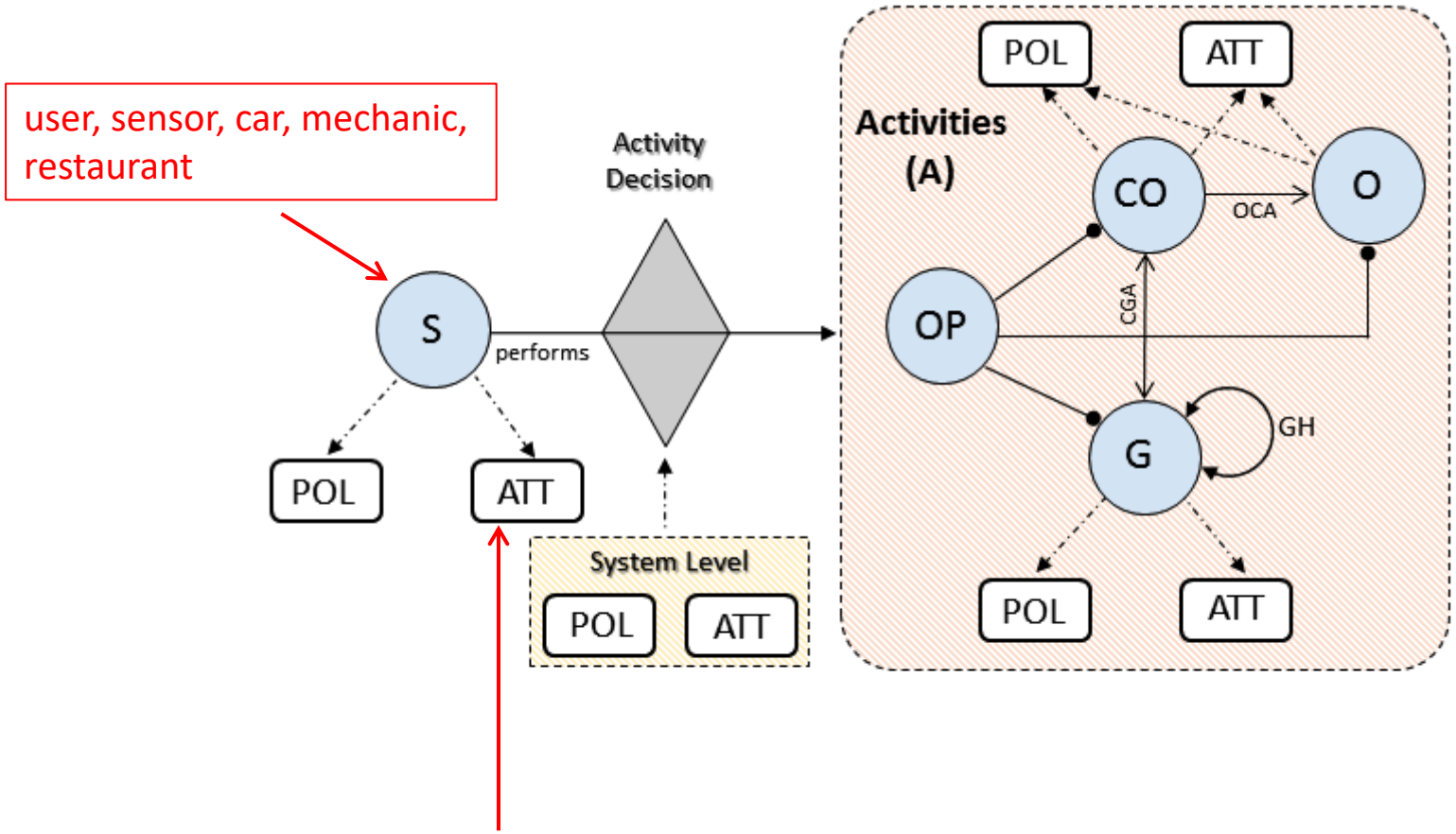
```
{"state": {"reported": {"Latitude": "29.4769353",
                        "Longitude":"-98.5018237"}}}
```

Reported MQTT message

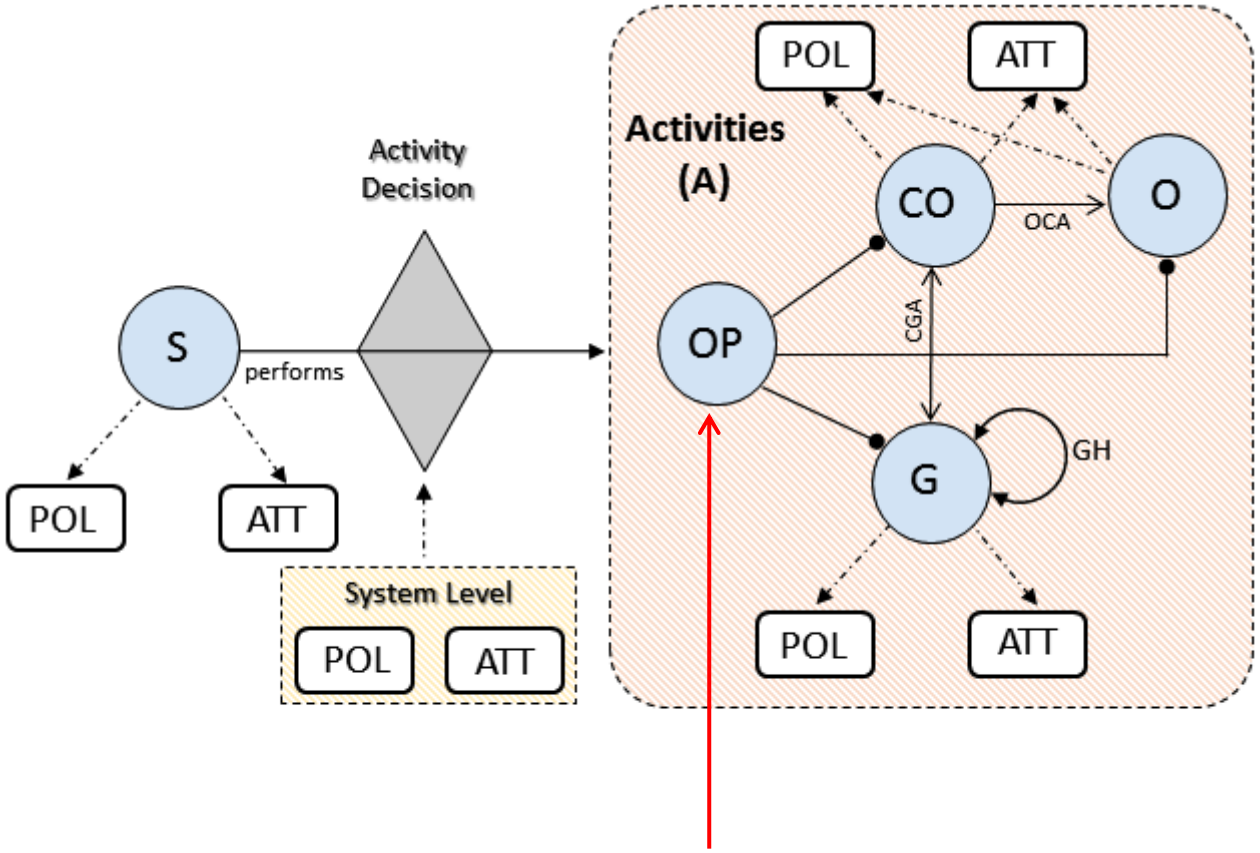
CV-ABAC$_G$ Model



Model Components

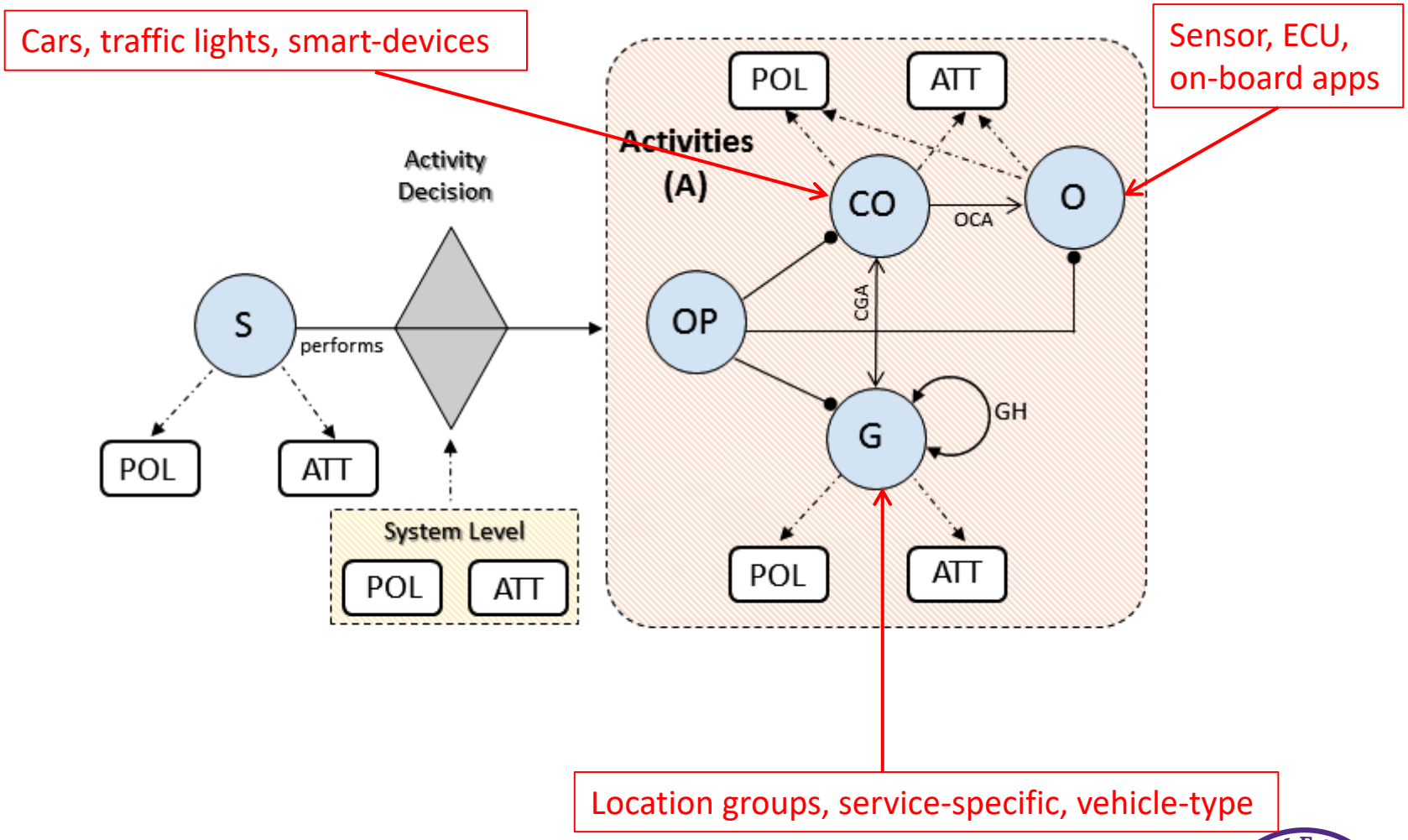


Model Components



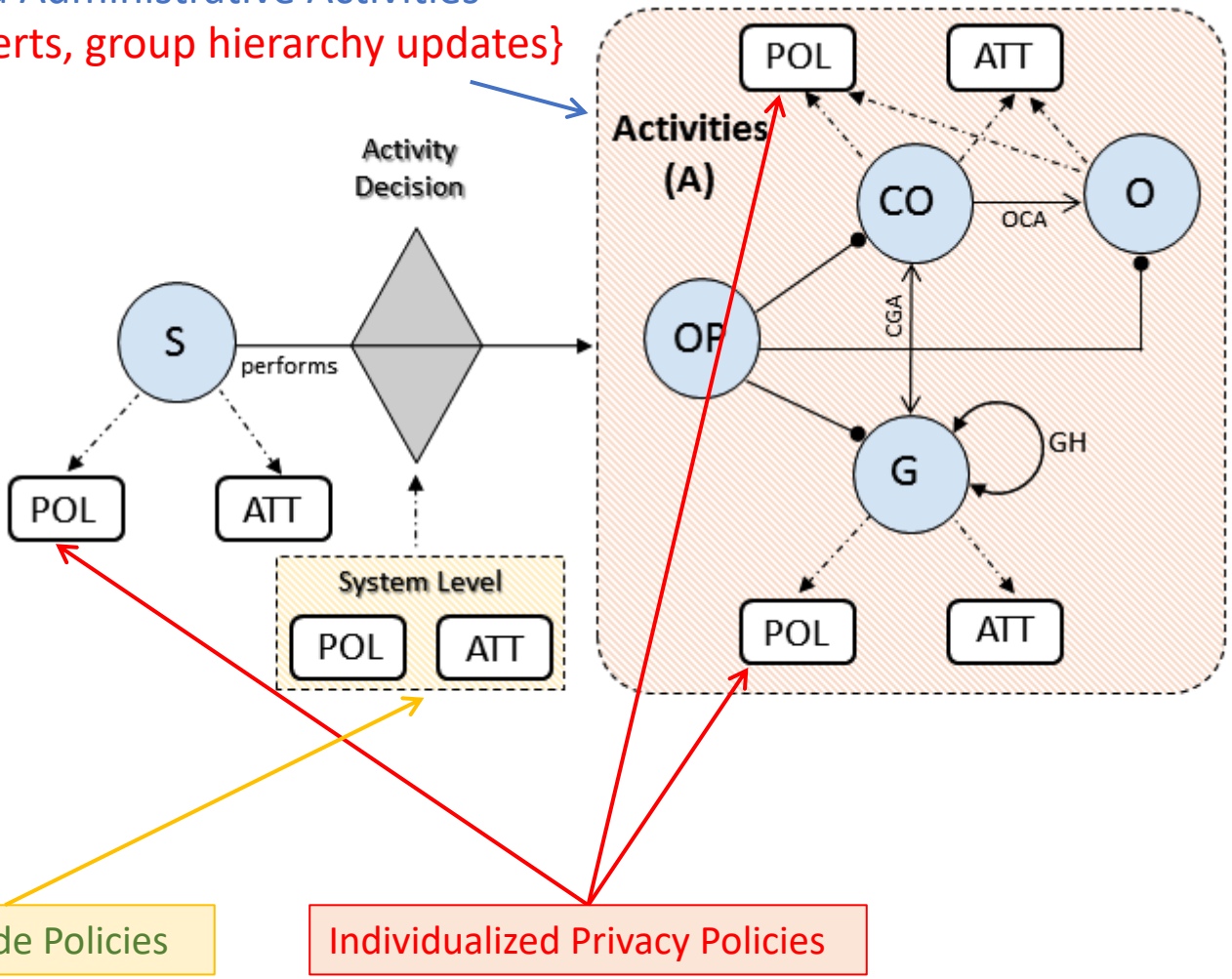
{ read, write, control, notify, administrative actions }

Model Components



Model Components

Operational and Administrative Activities
{notification, alerts, group hierarchy updates}



System Wide Policies

Individualized Privacy Policies

Formal Specification

Basic Sets and Functions

- S, CO, O, G, OP are finite sets of sources, clustered objects, objects, groups and operations respectively [blue circles in Figure 4].
- A is a finite set of activities which can be performed in system.
- ATT is a finite set of attributes associated with S, CO, O, G and system-wide.
- For each attribute att in ATT, Range(att) is a finite set of atomic values.
- attType: ATT = {set, atomic}, defines attributes to be set or atomic valued.
- Each attribute att in ATT maps entities in S, CO, O, G to attribute values. Formally,

$$\text{att} : S \cup CO \cup O \cup G \cup \{\text{system-wide}\} \rightarrow \begin{cases} \text{Range(att)} \cup \{\perp\} & \text{if attType(att)} = \text{atomic} \\ 2^{\text{Range(att)}} & \text{if attType(att)} = \text{set} \end{cases}$$

- POL is a finite set of authorization policies associated with individual S, CO, O, G.
- directG : CO $\rightarrow$ G, mapping each clustered object to a system group, equivalently CGA $\subseteq$ CO $\times$ G.
- parentCO : O $\rightarrow$ CO, mapping each object to a clustered object, equivalently OCA $\subseteq$ O $\times$ CO.
- GH $\subseteq$ G $\times$ G, a partial order relation $\succeq_g$ on G. Equivalently, parentG : G $\rightarrow 2^G$, mapping group to a set of parent groups in hierarchy.

Attribute Function

Attribute Type

Group Hierarchy

Attribute Mapping

Formal Specification

Effective Attributes of Groups, Clustered Objects and Objects (Derived Functions)

– For each attribute att in ATT such that attType(att) = set:

- $\text{effG}_{att} : G \rightarrow 2^{\text{Range(att)}}$, defined as $\text{effG}_{att}(g_i) = \text{att}(g_i) \cup \left(\bigcup_{g \in \{g_j | g_i \succeq_g g_j\}} \text{effG}_{att}(g) \right)$.
- $\text{effCO}_{att} : CO \rightarrow 2^{\text{Range(att)}}$, defined as $\text{effCO}_{att}(co) = \text{att}(co) \cup \text{effG}_{att}(\text{directG}(co))$.
- $\text{effO}_{att} : O \rightarrow 2^{\text{Range(att)}}$, defined as $\text{effO}_{att}(o) = \text{att}(o) \cup \text{effCO}_{att}(\text{parentCO}(o))$.

– For each attribute att in ATT such that attType(att) = atomic:

- $\text{effG}_{att} : G \rightarrow \text{Range(att)} \cup \{\perp\}$, defined as $\text{effG}_{att}(g_i) = \begin{cases} \text{att}(g_i) & \text{if } \forall g' \in \text{parentG}(g_i).\ \text{effG}_{att}(g') = \perp \\ \text{effG}_{att}(g') & \text{if } \exists\ \text{parentG}(g_i).\ \text{effG}_{att}(\text{parentG}(g_i)) \neq \perp \text{ then select parent } g' \text{ with } \text{effG}_{att}(g') \neq \perp \text{ updated most recently.} \end{cases}$
- $\text{effCO}_{att} : CO \rightarrow \text{Range(att)} \cup \{\perp\}$, defined as $\text{effCO}_{att}(co) = \begin{cases} \text{att}(co) & \text{if } \text{effG}_{att}(\text{directG}(co)) = \perp \\ \text{effG}_{att}(\text{directG}(co)) & \text{otherwise} \end{cases}$
- $\text{effO}_{att} : O \rightarrow \text{Range(att)} \cup \{\perp\}$, defined as $\text{effO}_{att}(o) = \begin{cases} \text{att}(o) & \text{if } \text{effCO}_{att}(\text{parentCO}(o)) = \perp \\ \text{effCO}_{att}(\text{parentCO}(o)) & \text{otherwise} \end{cases}$

Attributes more Dynamic

Attributes Inheritance

Policy Language

Authorization Functions (Policies)

– Authorization Function: For each op ∈ OP, $Auth_{op}$(s : S, ob : CO ∪ O ∪ G) is a propositional logic formula returning true or false, which is defined using the following policy language:

- α ::= $\alpha \wedge \alpha \mid \alpha \vee \alpha \mid (\alpha) \mid \neg\alpha \mid \exists x \in set.\alpha \mid \forall x \in set.\alpha \mid set\ \Delta\ set \mid atomic \in set \mid atomic \notin set$
- Δ ::= $\subset \mid \subseteq \mid \not\subseteq \mid \cap \mid \cup$
- set ::= $eff_{att}(i) \mid att(i)$ for att ∈ ATT, i ∈ S ∪ CO ∪ O ∪ G ∪ {system-wide}, attType(att) = set
- atomic ::= $eff_{att}(i) \mid att(i) \mid value$ for att ∈ ATT, i ∈ S ∪ CO ∪ O ∪ G ∪ {system-wide}, attType(att) = atomic

❖ Administrators in the police department can send alert to location-groups in city limits.

$Auth_{alert}(u{:}U, g{:}G) :: dept(u) \ Police \wedge parent\text{-}city(g) = Austin \wedge Austin \in jursidiction(u).$

❖ Only mechanic in the technician department from Toyota-X dealership must be able to read sensor in Camry LE. Further, this operation must be done between time 9 am to 6 pm.

$Auth_{read}(u{:}U, co{:}CO) :: role(u) \ Technician \wedge employer(u) = Toyota\text{-}X \wedge make(co) = Toyota \wedge model(co) = Camry\ LE \wedge operation\_time(u) \in \{9am, 10, 11 \ldots 6pm\}$

Activity Authorization Decision

Authorization Decision

– A source $s \in S$ is allowed to perform an activity $a \in A$, stated as Authorization($a : A, s : S$), if the required policies needed to allow the activity are included and evaluated to make final decision. These multi-layer policies must be evaluated for individual operations ($op_i \in OP$) to be performed by source $s \in S$ on relevant objects ($x_i \in CO \cup O \cup G$)

Formally, $\text{Authorization}(a : A, s : S) \Rightarrow \text{Auth}_{op_1}(s : S, x_1), \text{Auth}_{op_2}(s : S, x_2), \ldots\ldots\ldots\ldots, \text{Auth}_{op_n}(s : S, x_3)$

Evaluate all relevant policies to make a decision

A restaurant in group A must be allowed to send notifications to all vehicles in location group A and group B.

I only want notifications from Cheesecake factory.

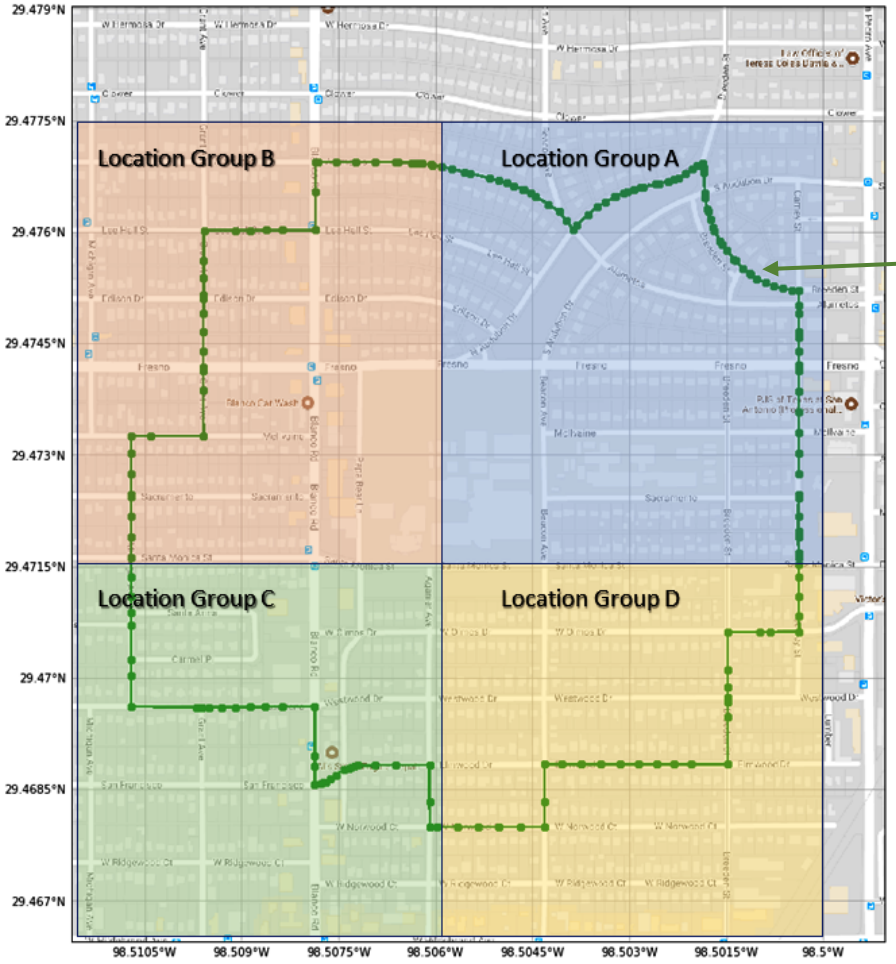
System defined

DECISION

User Preference

Implementation in Amazon Web Services (AWS)

Vehicles and Groups



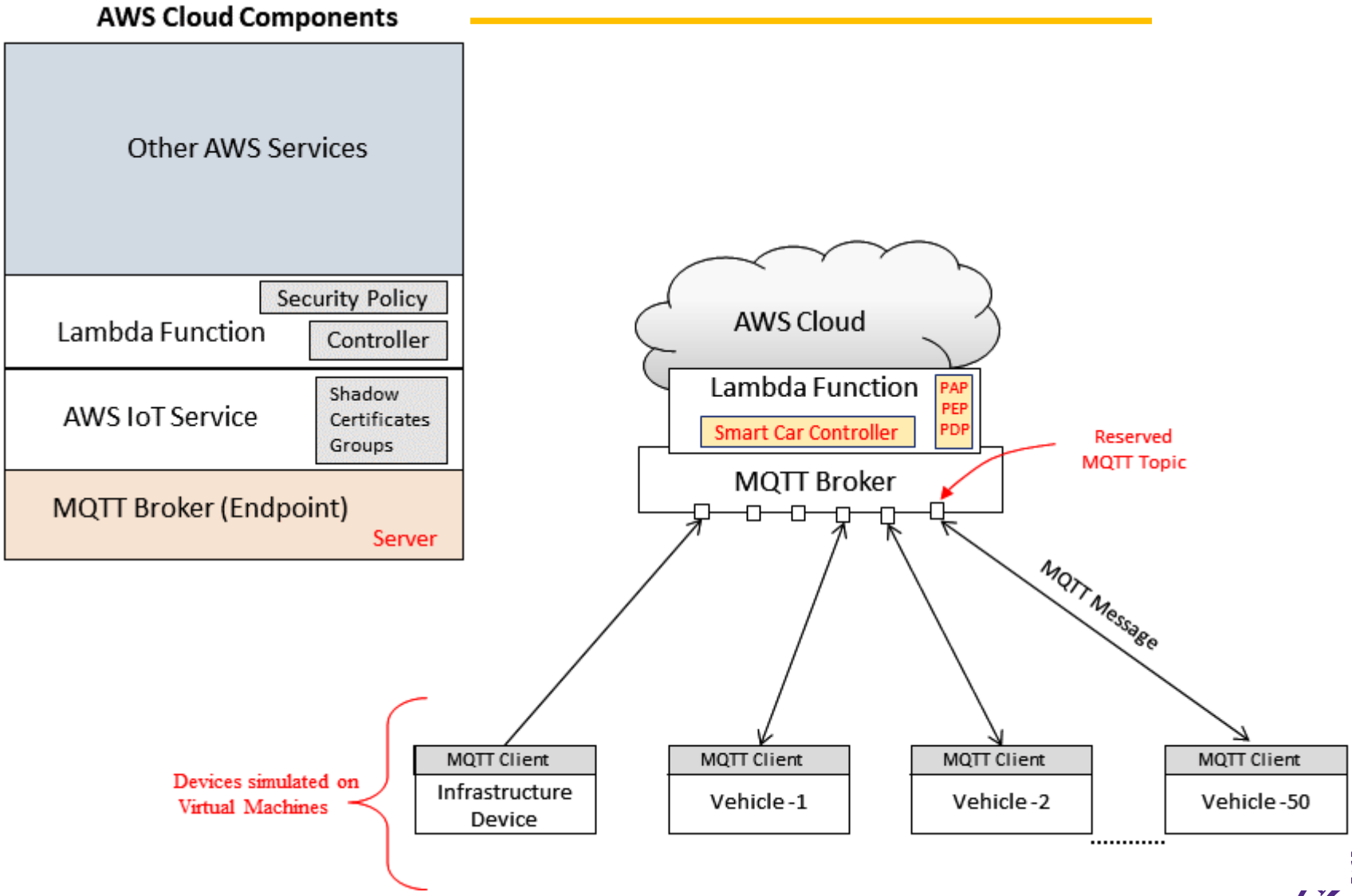
4 Location Groups
(static demarcation)

Vehicles movement
(coordinates generated
using Google API)

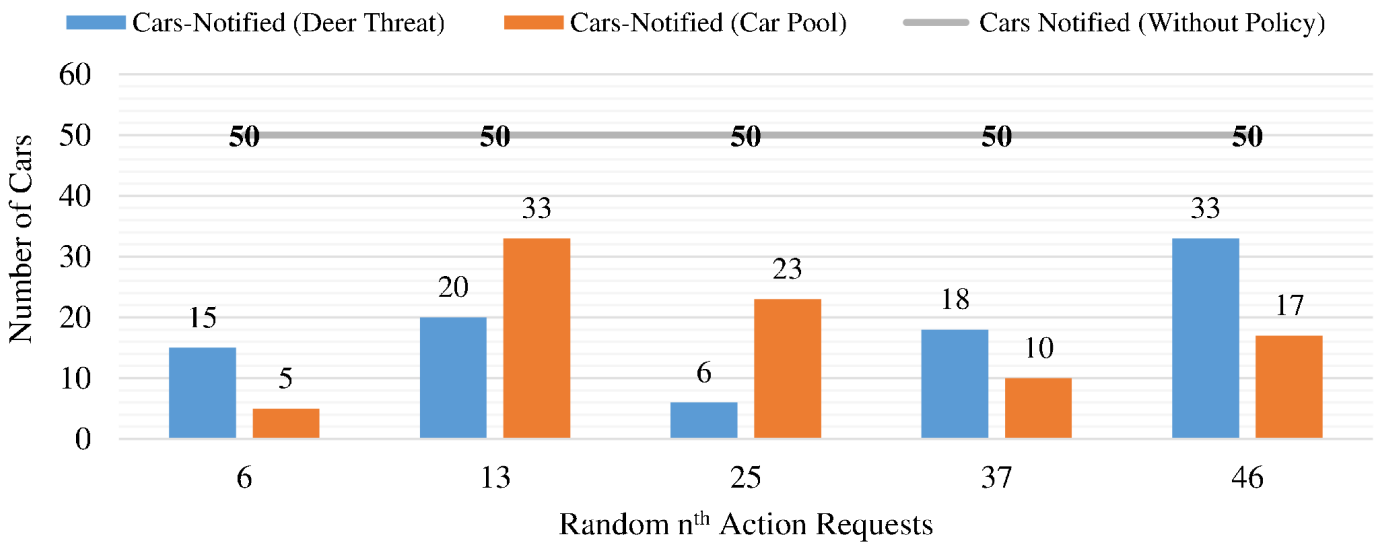
```
('Received new coordinates from:', 'Vehicle-1')
Sun May 27 02:56:30 2018
  Location A
     Car-A : [u'Vehicle-1', u'Vehicle-2']
     Bus-A : []
  Location B
     Car-B : []
     Bus-B : [u'Vehicle-6']
  Location C
     Car-C : [u'Vehicle-3', u'Vehicle-4']
     Bus-C : []
  Location D
     Car-D : []
     Bus-D : [u'Vehicle-5']
```

Snapshot (table keeps changing)

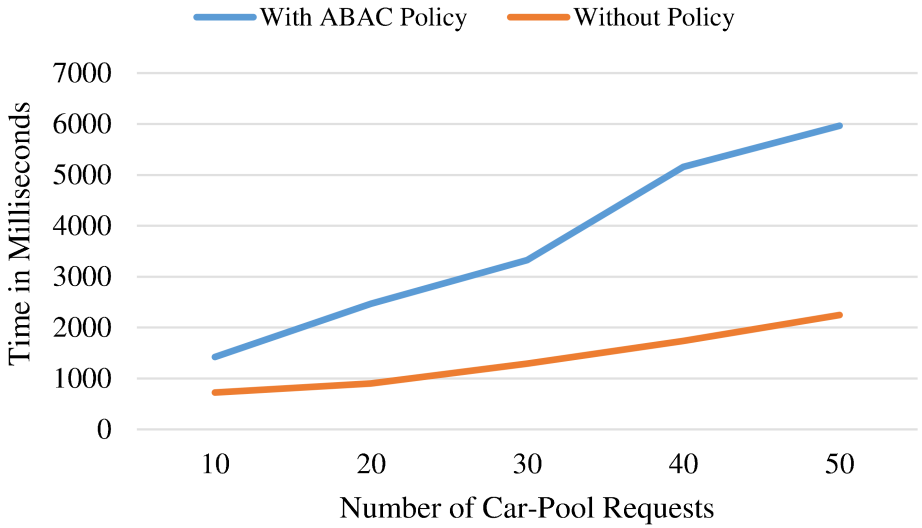
AWS Architecture

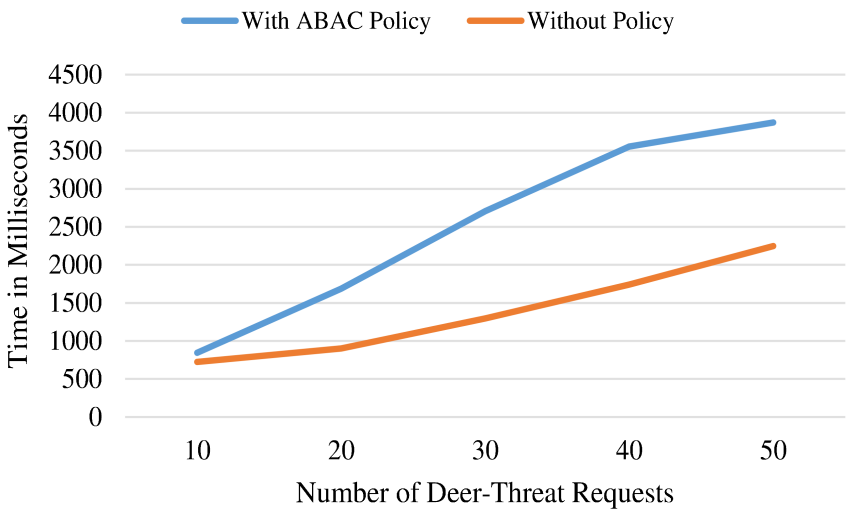


Performance Metrics



Performance Metrics





Let’s Talk ..!!

Questions, Comments or Concerns

mgupta@tntech.edu

www.maanakgupta.com