M-PIN AUTHENTICATION IN VEHICLE TRACKING

WITH APACHE MILAGRO.

Giorgio Zoppi <giorgio@apache.org>

- APACHE MILAGRO IS COMPOSED OF :

1. A SELF CONTAINED CRYPTO LIBRARY FOR IOT.
2. A MULTIFACTOR AUTHENTICATION SERVER
3. A TLS SUBSYSTEM BASED ON DISTRIBUTED TRUSTED AUTHORITIES

https://milagro.incubator.apache.org/

APACHE MILAGRO: A DISTRIBUTED CRYPTOSYSTEM FOR IOT

PKI+Password authentication is just fine in normal environment.

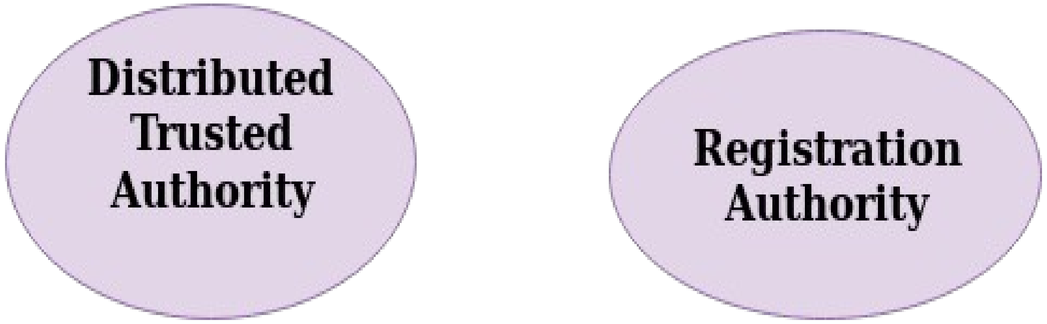
Our concerns with PKI in Internet of Things Environment:

1. Securing private keys in any device.
2. Updating Certificates before expirations in each client.
3. Dealing with certificate revocation.
4. CA are expensive. Our market has a lot of small companies.

AUTHENTICATION STATE OF ART : PKI + PASSWORD

HOW TO CHANGE:
THE ROAD FROM CA+PKI TO DTA+RA

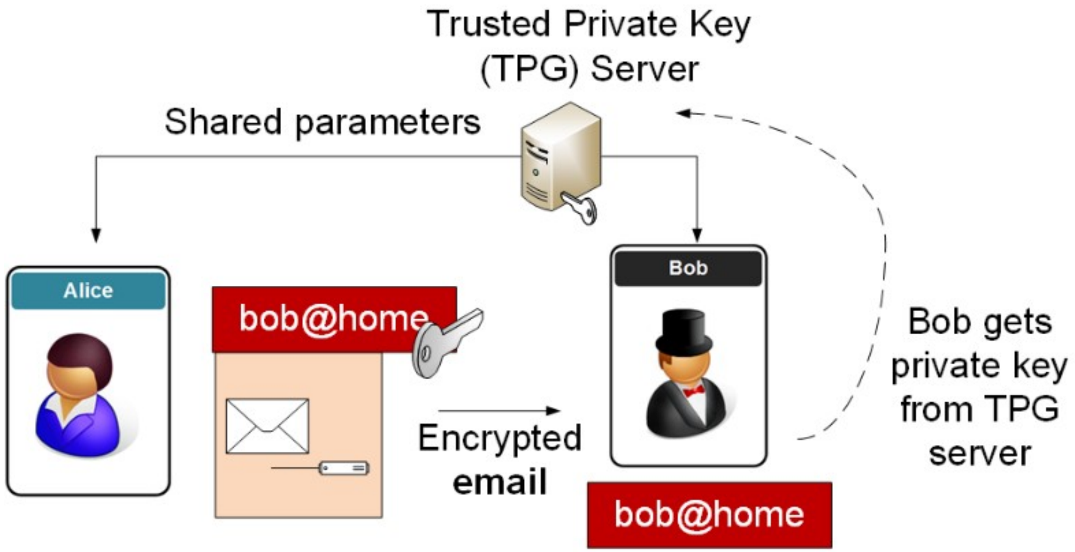
The idea is to decompose a CA in two distributed parts.



RA controls legitimate public key pairs in the System. RA is custom designed for each application.
Each **DTA** (Distributed Trusted Authority) assured the EC public keys with a master key.
Generally we have a set of DTA and the client use a part of the master key in order to **avoid a single part of compromise.**
Replication of DTA is enabled by a group of trusted professionals.

In order to succeed we need:

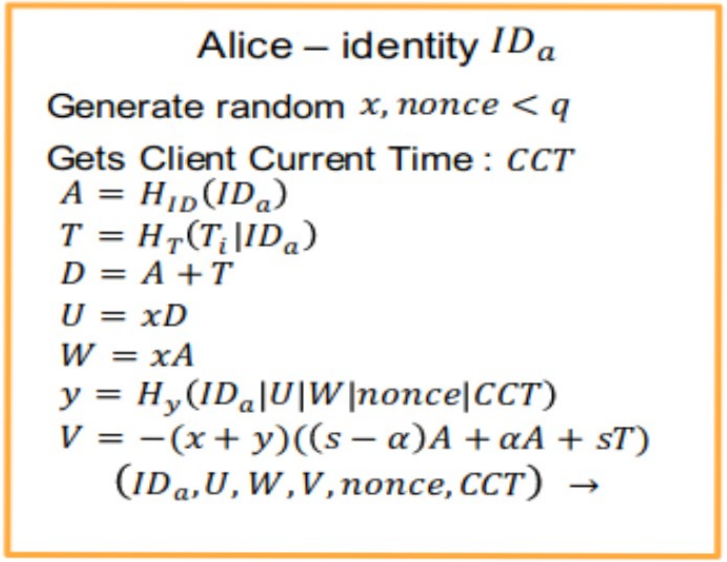
1. A mechanism for secure setup.
2. A mechanism for auth/revocation. Time Permits.
3. A mechanism to have a master distributed shared secret:
    1. Secrets are point of a elliptic curve.
    2. EC enable composition of keys.



Identity Based Encryption

HOW TO CHANGE. FROM CA+PKI TO DTA AND RA

MPIN PROTOCOL: HOW IT WORKS.



Alice – identity ID_a

Generate random $x, nonce < q$

Gets Client Current Time : CCT

$A = H_{ID}(ID_a)$

$T = H_T(T_i|ID_a)$

$D = A + T$

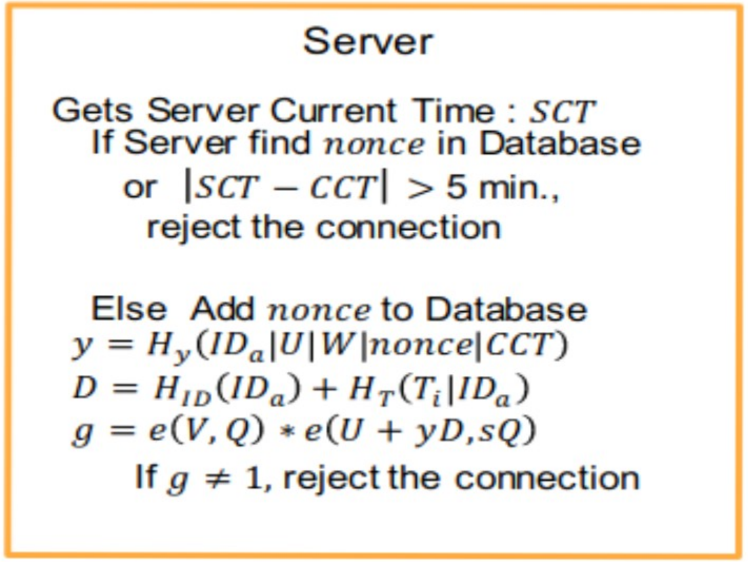
$U = xD$

$W = xA$

$y = H_y(ID_a|U|W|nonce|CCT)$

$V = -(x + y)((s - \alpha)A + \alpha A + sT)$

$(ID_a, U, W, V, nonce, CCT)$ →



Server

Gets Server Current Time : SCT

If Server find $nonce$ in Database

or $|SCT - CCT| > 5$ min.,

reject the connection

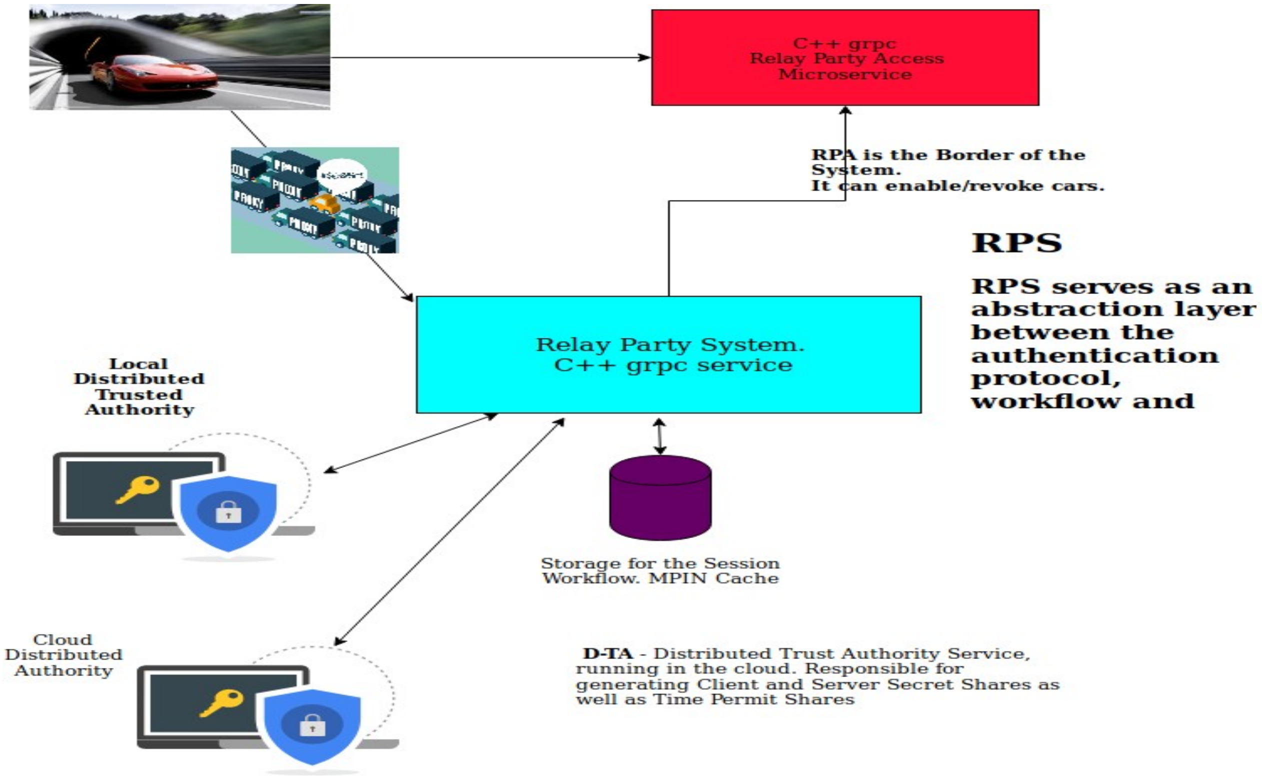
Else Add $nonce$ to Database

$y = H_y(ID_a|U|W|nonce|CCT)$

$D = H_{ID}(ID_a) + H_T(T_i|ID_a)$

$g = e(V, Q) * e(U + yD, sQ)$

If $g \neq 1$, reject the connection



C++ grpc
Relay Party Access
Microservice

RPA is the Border of the System.
It can enable/revoke cars.

RPS

RPS serves as an abstraction layer between the authentication protocol, workflow and

Relay Party System.
C++ grpc service

Local Distributed Trusted Authority

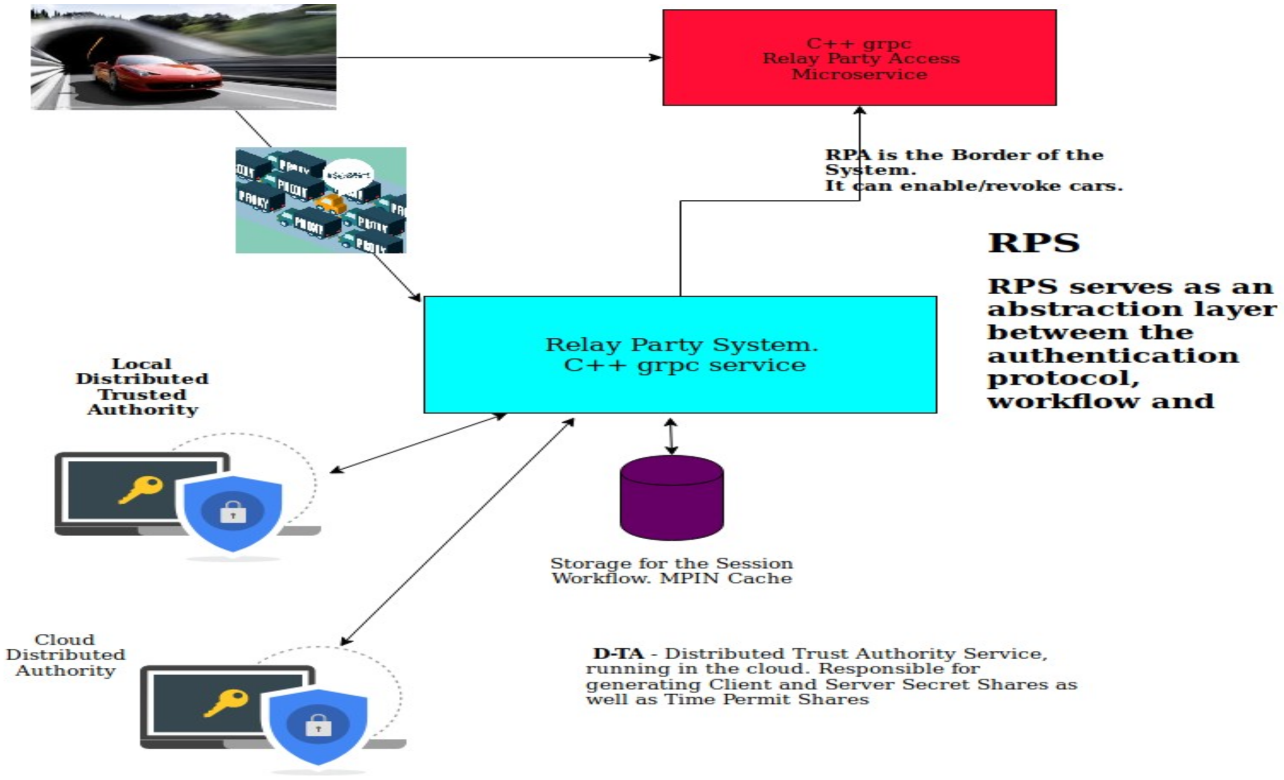
Storage for the Session Workflow. MPIN Cache

Cloud Distributed Authority

D-TA - Distributed Trust Authority Service, running in the cloud. Responsible for generating Client and Server Secret Shares as well as Time Permit Shares

CAR TRACKING AUTHENTICATION SUPPORT

MPIN VEHICLE SETUP PROCESS.



C++ grpc
Relay Party Access
Microservice

RPA is the Border of the System.
It can enable/revoke cars.

RPS

RPS serves as an abstraction layer between the authentication protocol, workflow and

Relay Party System.
C++ grpc service

Local
Distributed
Trusted
Authority

Storage for the Session Workflow. MPIN Cache

Cloud
Distributed
Authority

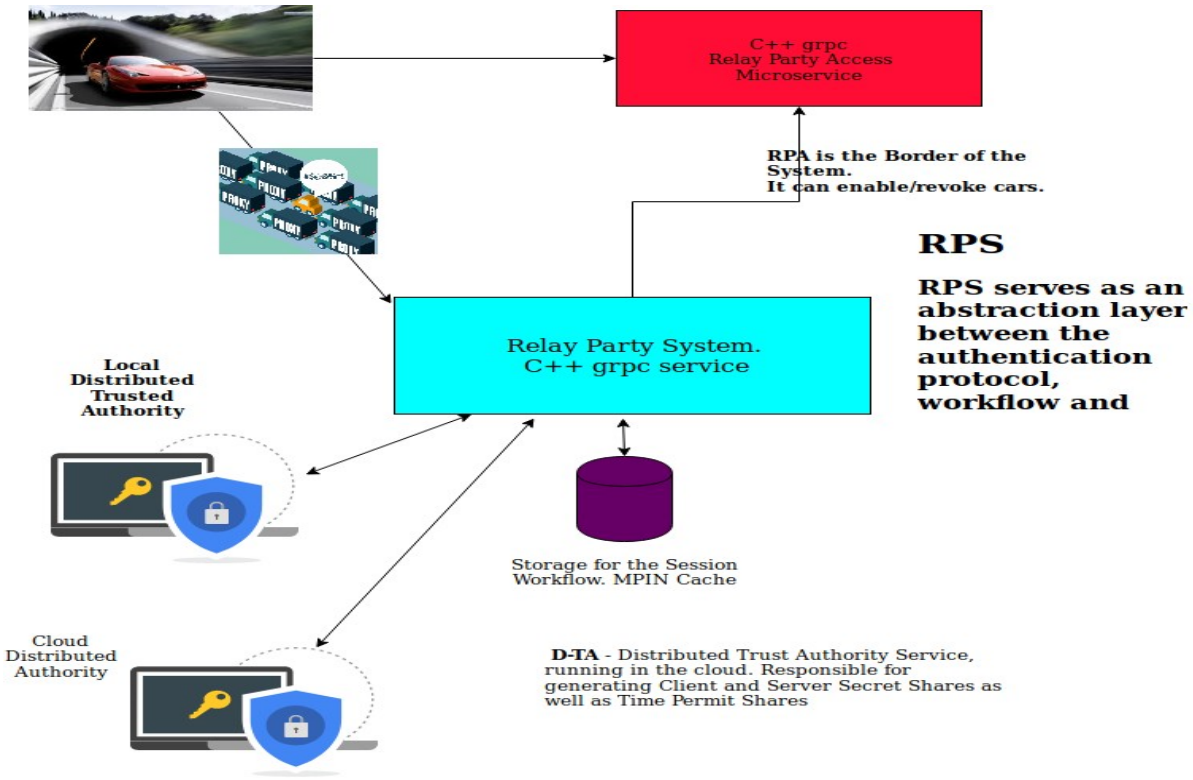
D-TA - Distributed Trust Authority Service, running in the cloud. Responsible for generating Client and Server Secret Shares as well as Time Permit Shares

The purpose of the Setup:

1. Verify user identity.
2. Get the two shares of the client secret and combine them.
3. Create a MPIN.
4. Store the MPIN inside the car.

After the setup we can proceed to the authentication.

CAR TRACKING AUTHENTICATION



RPA is the Border of the System.
It can enable/revoke cars.

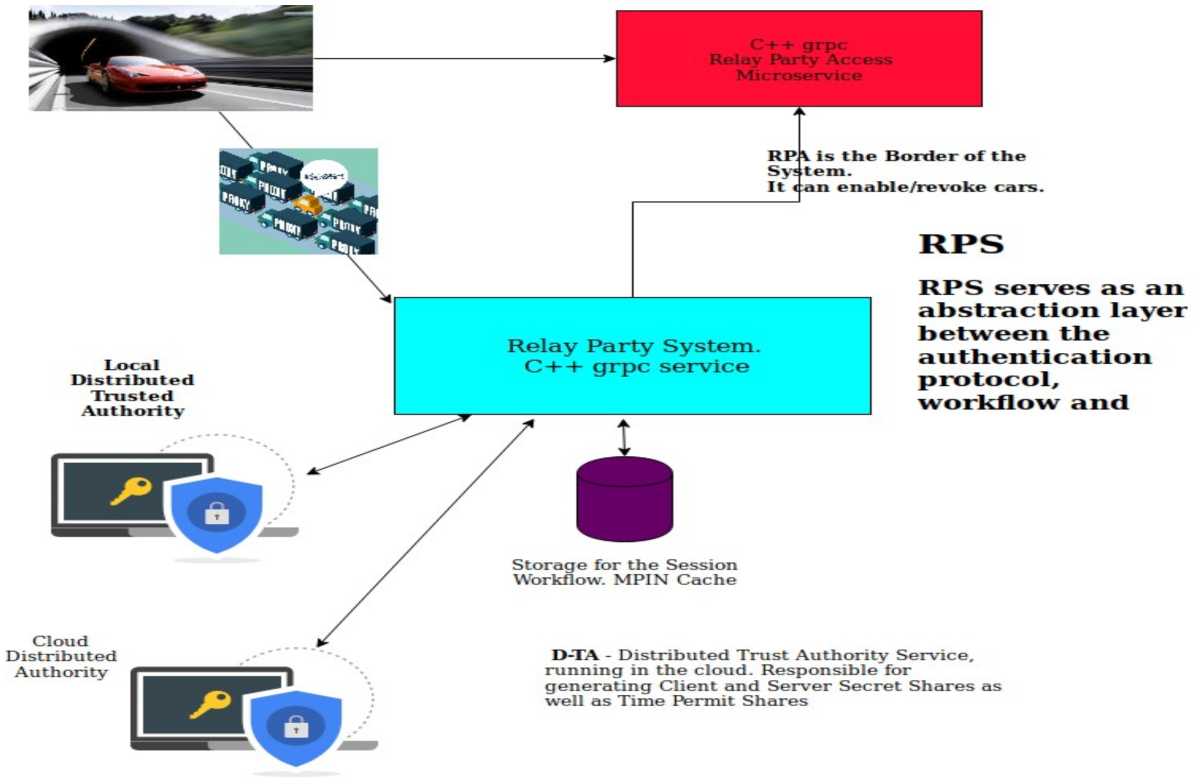
RPS

RPS serves as an abstraction layer between the authentication protocol, workflow and

D-TA - Distributed Trust Authority Service, running in the cloud. Responsible for generating Client and Server Secret Shares as well as Time Permit Shares

1. After the Car has been registered inside the DTA system, we can proceed. We assume that the Car has a valid token and a client secret used for signing messages.
2. The Car Initiate the authentication flow asking for a Time Permit to the for the tuple car identifier.

CAR TRACKING AUTHENTICATION



RPA is the Border of the System.
It can enable/revoke cars.

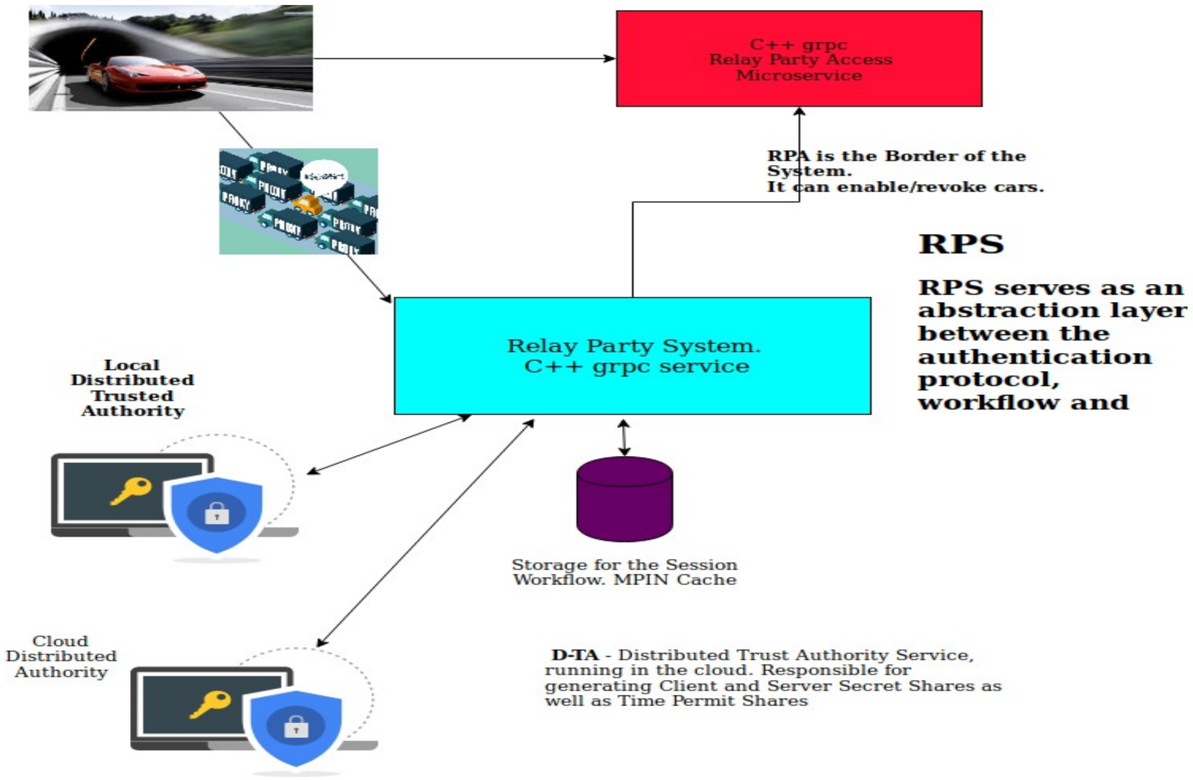
RPS

RPS serves as an abstraction layer between the authentication protocol, workflow and

D-TA - Distributed Trust Authority Service, running in the cloud. Responsible for generating Client and Server Secret Shares as well as Time Permit Shares

1. The Vehicle initiate the authentication flow asking for a Time Permit to the for the tuple userId = (vehicle model, variant,brand, driverName).
2, The RPS consults the RPA if the user has been banned.

CAR TRACKING AUTHENTICATION



C++ grpc
Relay Party Access
Microservice

RPA is the Border of the System.
It can enable/revoke cars.

RPS

RPS serves as an abstraction layer between the authentication protocol, workflow and

Relay Party System.
C++ grpc service

Local
Distributed
Trusted
Authority

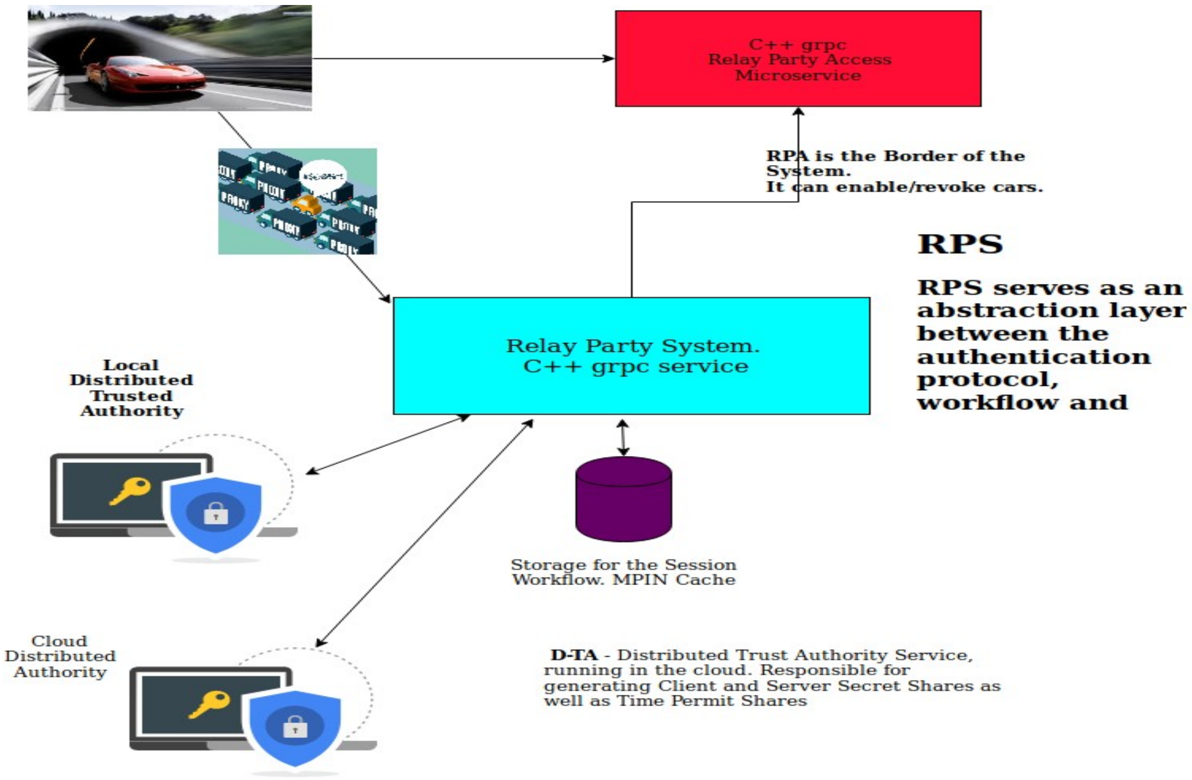
Storage for the Session
Workflow. MPIN Cache

Cloud
Distributed
Authority

D-TA - Distributed Trust Authority Service, running in the cloud. Responsible for generating Client and Server Secret Shares as well as Time Permit Shares

1. The Vehicle initiate the authentication flow asking for a Time Permit to the for the tuple userId = (vehicle model, variant,brand, driverName).
2, The RPS consults the RPA if the user has been banned.
3. The RPS consults DTAs for the composed time permit.

CAR TRACKING AUTHENTICATION



RPA is the Border of the System.
It can enable/revoke cars.

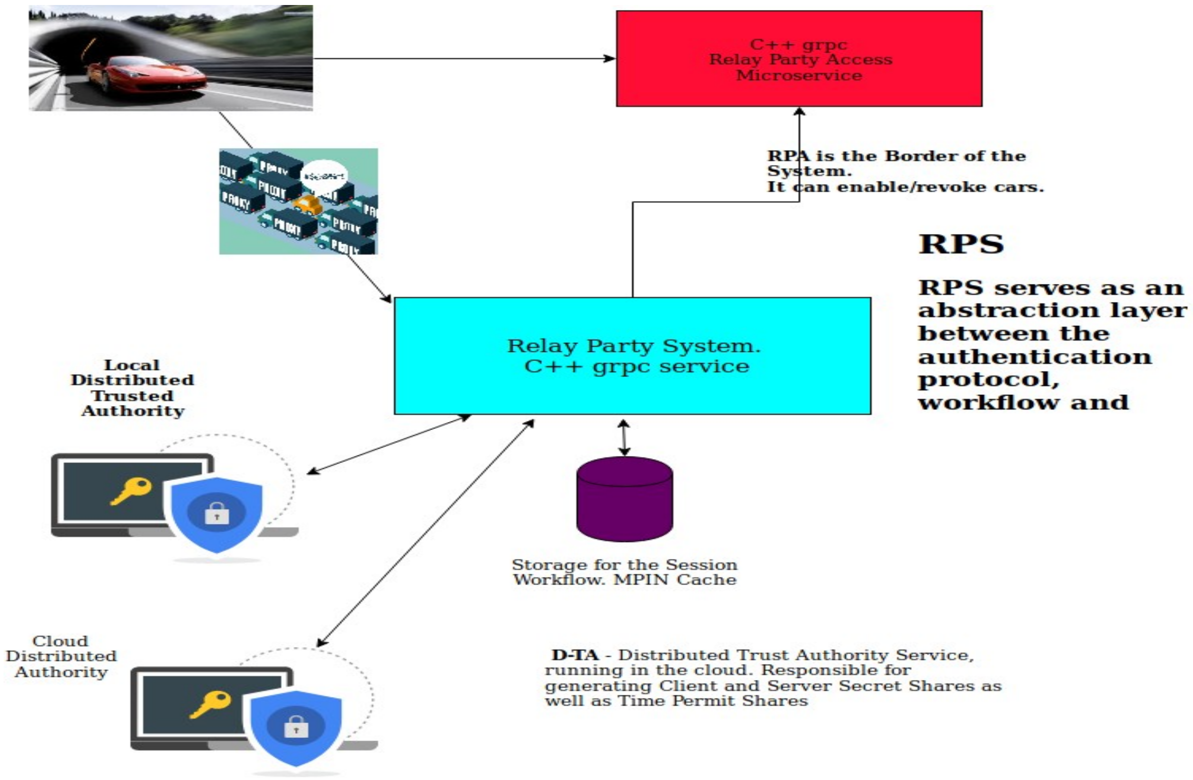
RPS

RPS serves as an abstraction layer between the authentication protocol, workflow and

D-TA - Distributed Trust Authority Service, running in the cloud. Responsible for generating Client and Server Secret Shares as well as Time Permit Shares

1. The Vehicle initiate the authentication flow asking for a Time Permit to the for the tuple userId = (vehicle model, variant,brand, driverName).
2. The RPS consults the RPA if the user has been banned.
3. The RPS consults DTAs for the composed time permit.
4. RPS send back the time permit to the Vehicle.

CAR TRACKING AUTHENTICATION



C++ grpc
Relay Party Access
Microservice

RPA is the Border of the System.
It can enable/revoke cars.

RPS

RPS serves as an abstraction layer between the authentication protocol, workflow and

Relay Party System.
C++ grpc service

Local
Distributed
Trusted
Authority

Storage for the Session
Workflow. MPIN Cache

Cloud
Distributed
Authority

D-TA - Distributed Trust Authority Service, running in the cloud. Responsible for generating Client and Server Secret Shares as well as Time Permit Shares

1. The Vehicle initiate the authentication flow asking for a Time Permit to the for the tuple userId = (vehicle model, variant,brand, driverName).

2, The RPS consults the RPA if the user has been banned.

3. The RPS consults DTAs for the composed time permit.

4. The RPS send back to the client.

5. The client starts the MPIN protocol.

THANK YOU

GIORGIO.ZOPPI@GMAIL.COM