Jordi Garcia Tremosa

Gestió d'esdeveniments de seguretat en un vehicle a temps real

Treball Fi de Grau
dirigit pel Dr. Enric Vidal

Grau d' Enginyeria en Electrònica Industrial i Automàtica

UNIVERSITAT ROVIRA I VIRGILI

Tarragona
2023

Índex

Índex de figures

1. Resum

Aquest treball és un projecte relacionat amb el món de l'automoció, realitzat durant el programa de pràctiques de l'empresa Lear Corporation. L'objectiu principal és crear un component de programació dins el conjunt dels controladors complexos en l'arquitectura AUTOSAR Clàssica, encarregat de la gestió dels esdeveniments de seguretat en el vehicle.

Per tant, s'ha elaborat un algorisme que utilitza una estructura de dades organitzades en matrius de cues circulars, en les que s'emmagatzema la informació i es protegeix la seva integritat com a servei als diferents elements del vehicle, tant informàtics com físics que són capaços d'enviar la informació en el format definit dins la xarxa de comunicació del vehicle.

Aquest projecte s'ha implementat com un conjunt d'elements integrats en un vehicle i permet la sincronització d'aquests. Els resultats obtinguts permeten observar el comportament del projecte davant de diferents entrades de dades, simulant possibles situacions reals en un entorn automobilístic real.

2. Resumen

Este trabajo es un proyecto relacionado con mundo de la automoción, realizado durante el programa de prácticas de la empresa *Lear Corporation*. El objetivo principal es crear un componente de programación dentro del conjunto de los controladores complejos en la arquitectura AUTOSAR Clásica, encargado de la gestión de los eventos de seguridad en un vehículo.

Para ello se ha elaborado un algoritmo que utiliza una estructura de datos organizados en matrices de colas circulares, en las que se almacena la información y protege su integridad como servicio a los diferentes elementos del vehículo, tanto informáticos como físicos que son capaces de enviar la información en el formato definido dentro de la red de comunicación del vehículo.

Este proyecto se ha implementado como herramienta a un conjunto de elementos integrados en un vehículo que permite la sincronización de datos de estos. Los resultados obtenidos permiten observar el comportamiento del proyecto delante diferentes entradas de datos, simulando posibles situaciones reales en un entorno automovilístico real.

3. Abstract

This work is a project related to the automotive world, carried out during the internship program of the Lear Corporation. The main objective is to create a software component within the set of complex drivers in the AUTOSAR Classic architecture, responsible for the management of security events in a vehicle.

For this, an algorithm has been developed that uses a data structure organized in arrays of circular tails, in which the information is stored and protects its integrity as a service to the different elements of the vehicle, both computer and physical, that can send the information in the format defined within the vehicle's communication network.

This project has been implemented as a tool to a set of elements integrated in a vehicle that allows the synchronization of data from them. The results obtained allow us to observe the behavior of the project in front of different data inputs, simulating possible real situations in a real automotive environment.

4. Introducció

Els éssers humans som curiosos per naturalesa, això ens fa qüestionar-nos el perquè de tot el que ens rodeja, pels enginyers en concret, el com funciona. Des de sempre els automòbils han despertat un interès innat en la meva consciència i em preguntava a mi mateix com funcionaven aquests vehicles incomprensibles.

Avui dia, em trobo realitzant una estada de pràctiques en una de les empreses més grans del sector automobilístic on tinc la oportunitat de respondre aquesta pregunta, com funciona? I no només això sinó ser part de la resposta creant un component de software que té la possibilitat de ser aplicat en un vehicle real.

El contingut d'aquest document explica el projecte elaborat durant la meva estada de pràctiques a Lear, Valls. On hi he creat un projecte assignat per l'empresa on he viscut l'experiència del món laboral en el camp de l'enginyeria.

Aquesta estada de pràctiques ha tingut una durada de deu mesos on he tingut la oportunitat d'aprendre de professionals de l'àmbit dels que he volgut absorbir tots els coneixements que he tingut oportunitat, per aplicar-los en el meu propi treball i millorar jo mateix com a futur enginyer.

El meu TFG proposa la gestió d'esdeveniments de seguretat d'un vehicle a temps real sobre l'arquitectura AUTOSAR *classic platform*. Un vehicle modern té una gran quantitat de sensors repartits pels diferents elements que el componen per monitorar el seu estat en cada moment. Aquests sensors en el moment d'enviar un senyal a la computadora del vehicle creen un esdeveniment de seguretat. Aquest projecte s'encarrega de gestionar els esdeveniments de seguretat que es puguin produir i crear un historial dinàmic, protegint la integritat dels últims esdeveniments més importants. De manera que en cas de fallida, es pugui inspeccionar el vehicle i recuperar la informació relativa a l'error del sistema físic o informàtic. Aquest és l'origen del Registre de Seguretat.

5. Aplicacions

Com s'ha mencionat anteriorment, el Registre de Seguretat té com a finalitat ser un producte per una empresa en el sector de l'automoció i com a tal té la possibilitat de ser implementat algun dia en un vehicle real. A continuació s'explicaran algunes de les aplicacions per les quals està pensat aquest projecte .

5.1 En l'arquitectura del vehicle

Al llarg de la història automobilística, l'arquitectura dels vehicles ha anat evolucionant, originalment l'arquitectura de l'electrònica d'un vehicle es denominava "arquitectura distribuïda" o "arquitectura modular" on els diferents components estaven directament connectats a la unitat de control, on cada component estava dissenyat únicament per una sola funció i el senyals que aquests retransmetien mitjançant busos de comunicació com LIN o CAN. Aquesta arquitectura implicava que el cablejat fos un dels elements que més pes aportava al vehicle, cada unitat de control es programava amb un programari específic i implementar funcions addicionals a posteriori del muntatge original requeria més material i cost final.

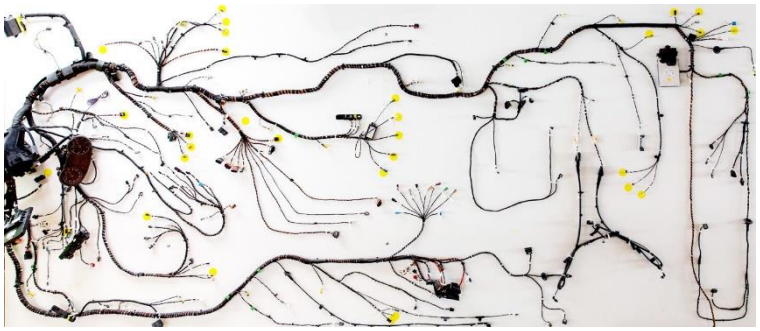

Figura 1: Arquitectura modular d'un vehicle

Actualment, els vehicles segueixen una "arquitectura de domini". Com diu la nomenclatura, consisteix a agrupar els diferents sensors i actuadors segons tasques o funcionalitats (dominis). Algunes de les categories actuals, són: accionament, confort i *infotainment .* Cada domini està governat per una connexió bus anomenada *gateway* que permet la implementació de funcions i dominis addicionals en un vehicle, però no soluciona el problema de la quantitat de cables, del pes, el consum associat i, per tant, pèrdua de rendiment.

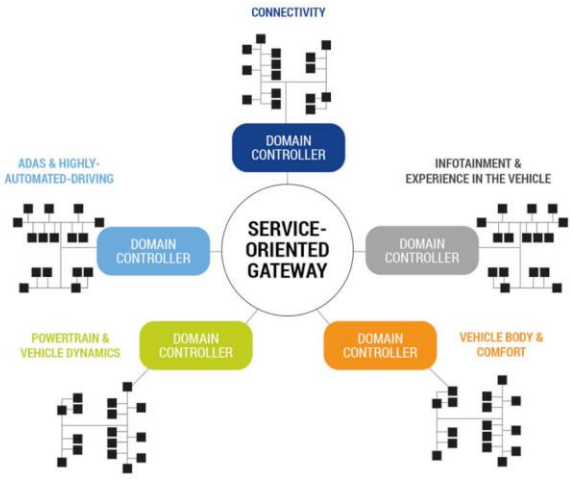

Figura 2: Diagrama d'arquitectura de domini d'un vehicle

Els vehicles de la pròxima generació, proposen un canvi d'arquitectura on el cablejat del vehicle passa d'estar controlat per una computadora central a estar format per diferents computadores per les diferents zones del vehicle "l'arquitectura zonal". En aquesta nova implementació, els components del vehicle es controlen segons la zona física on es troben i no segons la seva funcionalitat. En un vehicle estàndard es poden trobar fins a 4 zones controlades per una unitat de control zonal i connectades a una unitat de control central. La simplificació del cablejat, la modularitat i la flexibilitat permeten treure un màxim rendiment a les capacitats i possibilitats d'aquesta nova arquitectura basada en el tractament d'informació i comunicació entre els diferents components del vehicle.

El Registre de Seguretat podria implementar-se en les diferents unitats de control de l'arquitectura zonal permetent la gestió dels esdeveniments provocats pels diferents elements del sistema, creant una base de dades que pugui ser tractada per així tenir un millor coneixement i control sobre la informació del vehicle. Al ser un component modificable, és modular i adaptable en funció de les necessitats requerides, formant part de l'arquitectura de la següent generació.

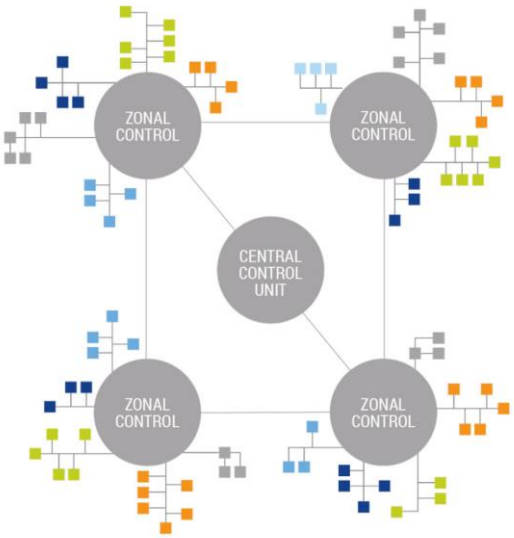

Figura 3: Diagrama d'arquitectura zonal d'un vehicle

5.2 En la seguretat del vehicle

Durant els últims vint anys, el sector automobilístic ha patit un gran nombre d'amenaces relacionades amb la ciberseguretat que han afectat diferents funcionalitats dels vehicles en totes les empreses del sector. La més destacable va ser el 2015, quan Chris Valasek i Charlie Miller, dos "*white hat hackers*" , *hackers* de bona fe, van demostrar com era possible controlar un vehicle a distància. Van ser capaços de controlar un gran nombre de funcionalitats com la ràdio, accelerador, anul·lar els frens, inclús girar el volant.

Per aquest motiu els vehicles moderns necessiten incorporar elements que s'oposin als atacs cibernètics. El Registre de Seguretat forma part d'aquesta solució. La seva funció és formar una base de dades dels esdeveniments del vehicle i incorpora una característica essencial, la comprovació de la integritat de la informació. El fet de verificar constantment que la informació no ha sigut modificada per un agent extern, permet detectar si al vehicle hi ha hagut una intrusió i així activar el sistema de seguretat del vehicle. Una altra funció és guardar la informació per poder permetre una anàlisi forense sobre el vehicle i així millorar la seguretat.

En la indústria de l'automoció s'ha arribat a una proposta de solució a la qual es dona llibertat d'interpretació. Fonamentalment, la detecció d'intrusions en els vehicles està formada per cinc elements principals: la detecció d'intrusió, l'avís, l'anàlisi de la intrusió, el desenvolupament d'una solució i finalment la implementació sobre el vehicle a fi de millorar la seguretat contra futures intrusions.

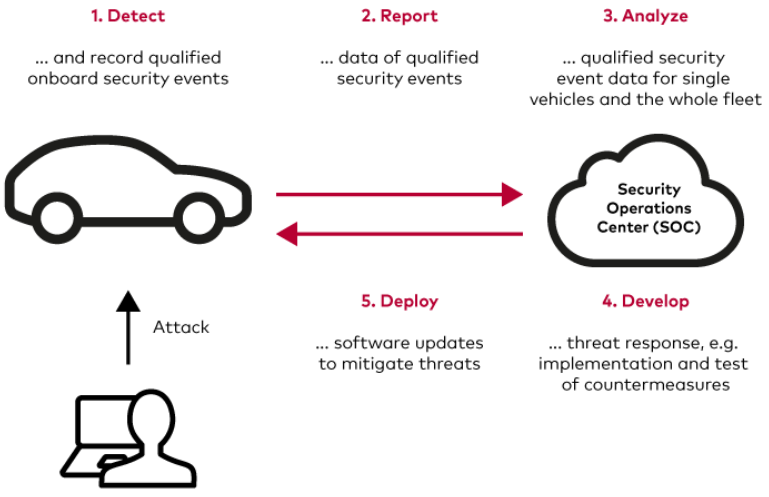

Figura 4: Diagrama solució de detecció contra intrusions AUTOSAR

Existeixen dues principals vies de resolució per aquest mateix problema i la implementació varia en funció del client i els requisits.

La primera via és la solució on s'incorpora un sistema de detecció d'intrusió IdsR, encarregat de verificar que les diferents unitats de control extern ECU, no han estat atacades i notificar a la unitat central *Security Operation Center* SOC en cas d'intrusió al vehicle.

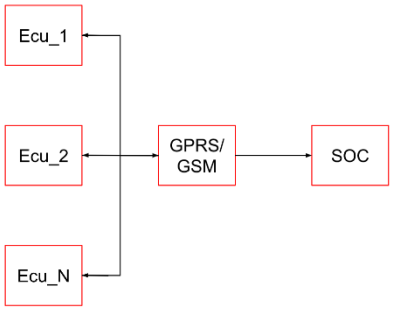

Figura 5: Solució mitjançant alerta remota

La segona solució i la que s'implementa en aquest projecte és la solució que guarda la informació en memòria per dur a terme les comprovacions de forma periòdica mentre el sistema actua a temps real de forma paral·lela.

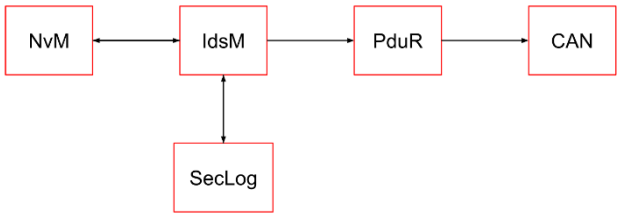

Figura 6: Solució mitjançant alerta local

Com en molts camps en el món de l'enginyeria, no existeix una única solució per a un mateix problema i la implementació d'un sistema o un altre depèn de la decisió del client, en aquest cas la casa automobilística que contracta el projecte amb Lear.

6. Abast del treball

Com que un projecte en el món de l'automoció abraça una gran quantitat de requisits, funcionalitats i exigències que s'han de complir seguint els estàndards, per aquest motiu des d'un principi ha estat plantejat i enfocat en la primera part de la solució, la detecció d'intrusions.

L'abast del treball inclou el disseny del codi, tenint en compte les característiques de l'entorn i les necessitats en les quals està pensat el seu funcionament. La introducció a l'arquitectura AUTOSAR *classic platform*. La metodologia de treball en el projecte, millores dutes a terme durant la creació del projecte. Diagrames de funcionament del codi i les proves realitzades per garantir el compliment dels objectius i correcte funcionament del Registre de Seguretat.

7. Objectius

L'Objectiu principal del projecte és implementar el Registre de seguretat en l'arquitectura AUTOSAR Clàssic.

El Registre de Seguretat, al ser un projecte que està destinat a una aplicació real, ha de complir un seguit de requisits que li permetin ser un producte funcional per a la finalitat a la qual està dissenyat. Els objectius específics que es pretenen complir en aquest projecte són els següents:

La capacitat de ser personalitzable mantenint totes les seves funcions per poder ser incorporat al projecte que sigui necessari.

Necessita ser independent del sistema físic al qual s'incorpora.

La captació d'esdeveniments de seguretat generats per diferents aplicacions del sistema que actuaran com a clients que estan fent una enquesta per poder registrar un nou esdeveniment.

Un cop enregistrats els esdeveniments de seguretat, aquests necessiten ser filtrats de diferents formes en funció de les necessitats del sistema on està integrat el projecte, la freqüència dels diferents esdeveniments o la importància d'aquestes.

Finalment, el Registre de Seguretat ha de ser capaç de gestionar els esdeveniments filtrats, guardar-los en memòria no volàtil i garantir la integritat de la informació emmagatzemada.

8. AUTOSAR

Quan parlem d'automoció, el primer concepte que és necessari definir és l'arquitectura AUTOSAR. Què és?

L'arquitectura AUTOSAR (*Automotive Open System Architecture)* és una aliança formada per fabricants, proveïdors, sector serveis, empreses d'electrònica, semiconductors i programació en la indústria de l'automoció que defineix un estàndard en les empreses per oferir solucions i productes dins el mercat automobilístic.

Aquest acord entre empreses proporciona uns grans avantatges a l'hora de desenvolupar noves tecnologies, estalviant temps en l'adaptació de nous sistemes, permet la col·laboració entre diferents entitats, augmenta la productivitat i incrementa el percentatge d'èxit dels productes que surten al mercat.

8.1 Plataformes

Els elements principals de l' arquitectura AUTOSAR són els següents:

Classic platform: una rama per sistemes encastats que treballen a temps real amb restriccions de temps i seguretat molt estrictes.

Adaptive platform: la branca nova, creada per les noves ECU d'alt rendiment que estan pensades per sistemes que garanteixin la seguretat i compliment del seu propòsit inclús en cas de fallida com per exemple en la implementació de la conducció autònoma.

Foundation: El propòsit d'aquest bloc és reforçar la sincronització entre les diferents plataformes de AUTOSAR mitjançant la sincronització de protocols de comunicació i metodologies en comú.

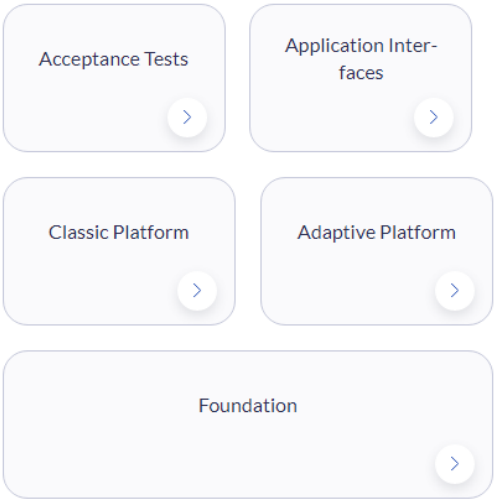

Figura 7: Diagrama Plataformes de l'arquitectura AUTOSAR

8.1.1 Plataforma clàssica

La plataforma AUTOSAR Clàssic és la utilitzada en els sistemes de temps real i la necessària per aquest projecte. Segueix una arquitectura per capes. Aquestes són: software bàsic BSW, la interfície a temps real (RTE) i la capa d'aplicacions.

La capa BSW és la més pròxima a la capa física i a la mateixa vegada està formada per tres capes. L'abstracció del microcontrolador, l'abstracció de la unitat de control (ECU) i la capa de serveis. L'objectiu d'aquesta arquitectura és la separació entre les possibles aplicacions i els elements orientats específicament pel hardware del sistema. D'aquesta manera és possible realitzar una estandardització entre els components de diferents fabricants per poder escalar i personalitzar els elements del vehicle en cada línia de producció, permetent l'ús de diferents unitats de control externes (ECU) i paquets mòduls programables.

Aquesta abstracció entre la capa física i la capa de programació és possible gràcies a l'intercanvi d'informació que es genera entre les diferents capes de l'arquitectura. A causa de de l'estandardització existent, es poden desenvolupar components físics i programables sense la necessitat de saber les característiques del seus components complementaris permetent així una gran flexibilitat en desenvolupament.

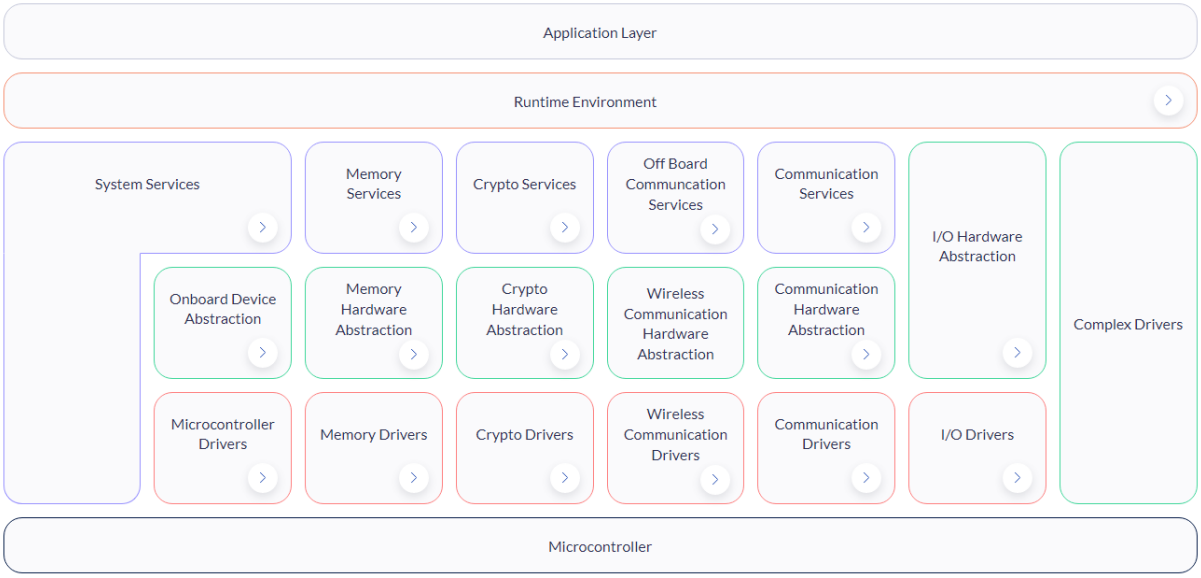

Figura 8: Diagrama capes de la plataforma AUTOSAR Classic

8.1.2 Plataforma Adaptive

A mesura que avança la tecnologia, l'evolució del processadors tant en capacitat de treball com en reducció de volum és un dels fronts oberts avui en dia. El món de la automoció és el primer interessat a fer ús de les tecnologies més punteres i involucrar-les en els diferents serveis que ofereix un vehicle.

El concepte de la plataforma Adaptive és ser un entorn flexible des del qual les aplicacions poden ser modificades o afegides segons les necessitats del client mitjançant els diferents sistemes ADAS (Sistema d'Assistència Avançada al Conductor) com per exemple l'assistència a canvi de carril, frenada d'emergència o llums de carretera intel·ligents. Això ens porta al concepte de l'actualització dels vehicles en un món que constantment canvia. El que li dona el nom a la plataforma: "adaptive" és la capacitat de canviar les seves característiques i qualitats fins i tot mentre el vehicle està sent utilitzat i les aplicacions del vehicle s'executen de manera independent.

L'entorn AUTOSAR Adaptive ha estat emergent a partir de les ECUs cada vegada més potents amb una major capacitat de comunicació i computació. Els vehicles cada vegada són capaços de captar més informació del seu entorn cosa que provoca la millora dels canals de comunicació, de protocol CAN a Ethernet, actualitzacions OTA (*Over The Air*), processament d'imatges i en un futur inclús la comunicació entre vehicles.

Figura 9: Diagrama de capes de la plataforma AUTOSAR Adaptive

8.2 Elements utilitzats d'AUTOSAR clàssic

Els elements de la plataforma clàssica amb els que el projecte estarà relacionat són: els serveis criptogràfics, els serveis d'accés a memòria, els serveis de comunicació i els serveis criptogràfics de les capes superiors del BSW. A continuació s'explicarà la funció dels elements principals relacionats amb el projecte que es troben vinculats de forma explícita.

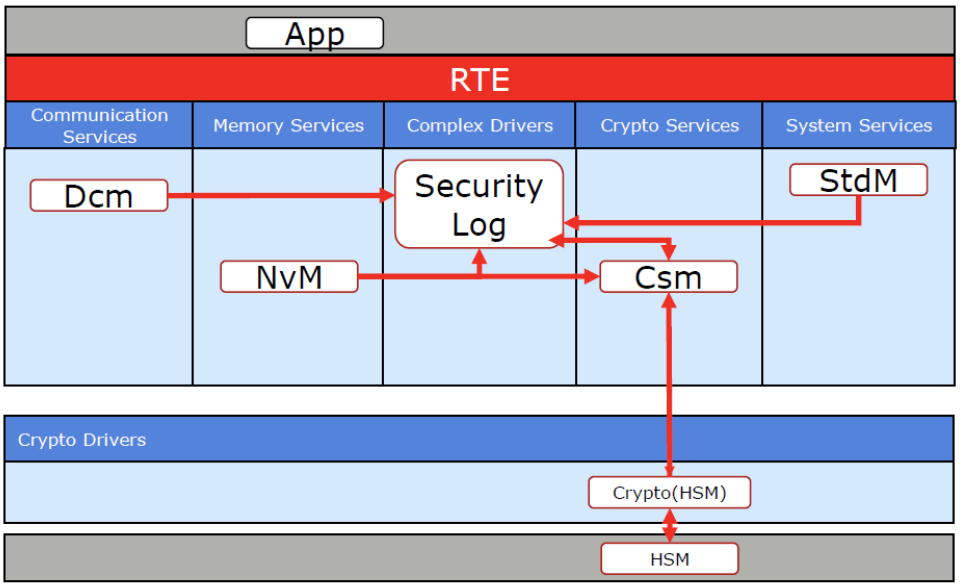

Figura 10: Diagrama d'integració del component Security Log en l'arquitectura AUTOSAR Classic

8.2.1 Crypto Service Manager

El conjunt d'elements encarregats de la criptografia són comunament anomenats *Cripto Stack*, permet la configuració del servei en funció de les necessitats de l'aplicació i client. El CSM proveeix serveis síncrons i asíncrons que habiliten l'accés a les funcionalitats criptogràfiques a la resta dels components del vehicle. Es troba en la capa de l'abstracció de software oferint una interfície estàndard per accedir a les capes inferiors del sistema.

Els components de capes inferiors relacionats amb el CSM són el *CryIf* (*Crypto* Service Interface) la capa intermitja entre la comunicació del software d'alt nivell i la capa física on es troben els *Crypto Drivers*, els microprocessadors encarregats de gestionar el càlcul criptogràfic.

Amb els nous perills que apareixen amb els atacs cibernètics, és necessària la protecció de les dades, comunicacions i accessos. Aquesta protecció es brinda mitjançant la criptografia i els diferents algoritmes implementats. Tota aquesta protecció requereix càlculs i aquests suposen una major càrrega al microprocessador. Aquest fet ha provocat que s'implementin microprocessadors exclusius per al càlcul criptogràfic i així mantenir el rendiment dels components electrònics del vehicle.

El processador dedicat a la criptografia s'anomena HSE (*Hardware Security Engine*). És un mòdul físic, protegit contra l'accés extern tant per software com per hardware. Aquest microprocessador no té la capacitat de comunicar-se amb l'exterior i l'única forma de transmetre-li informació és mitjançant una adreça de memòria des de la qual llegeix les dades per processar i retornar. I en cas d'intrusió per hardware, aquest sistema està implementat de forma que si s'intenta manipular, elements crucials del sistema es trenquen i no permeten la continuació del funcionament.

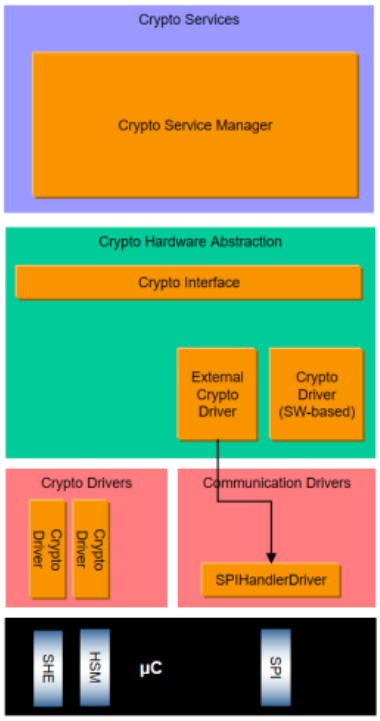

Figura 11: Diagrama de capes Crypto Stack

8.2.2 Non Volatile RAM Manager

El mòdul encarregat de gestionar la informació guardada en memòria no volàtil s'anomena NvM. Les seves tasques principals són oferir serveis que permetin i assegurin el guardat i manteniment de la informació en memòria en funció dels requisits per cada aplicació. Interactua amb la memòria no volàtil amb la finalitat de llegir i escriure les dades necessàries.

La següent capa, amb la que interactua l'NvM és el MemIf (Memory Interface), que és el responsable de l'abstracció de la capa física de les unitats de memòria que es troben a la ECU. El mòdul és independent del hardware existent en el sistema i s'encarrega d'administrar els blocs de memòria identificats individualment mitjançant un "block ID" i en cas de ser necessari poden tenir informació addicional per facilitar la gestió. Les aplicacions no són capaces d'accedir a memòria no volàtil de forma independent, solament a la memòria volàtil (RAM). Per aquest motiu, es produeixen comprovacions constants de sincronització de la memòria volàtil i la no volàtil a fi d'assegurar el correcte funcionament del conjunt.

El funcionament del NvM comença abans que els components encarregats d'administrar les ECUs actuïn, s'inicialitza en dos passos. El primer es tracta d'inicialitzar la màquina d'estats interna i diferents cues del component, a continuació es fa un bolcat d'informació de la memòria no volàtil a la memòria dinàmica. Com aquest procés pot implicar la saturació de recursos, se segueix un ordre específic.

En primer lloc, el bloc inicial és el "NvM_Init()" que s'encarrega de definir els paràmetres necessaris per al correcte funcionament del component. En segon lloc tots els blocs de memòria configurats per ser transmesos en la inicialització del sistema, ja que interessa disposar d'informació anterior, es posen en una cua que anirà processant-los progressivament. Per determinar si la informació emmagatzemada és correcta, abans de ser transmesa passa per una avaluació realitzada per un *checksum*. Si la informació es valida, es transmetran els blocs d'informació a la seva imatge corresponent en memòria dinàmica.

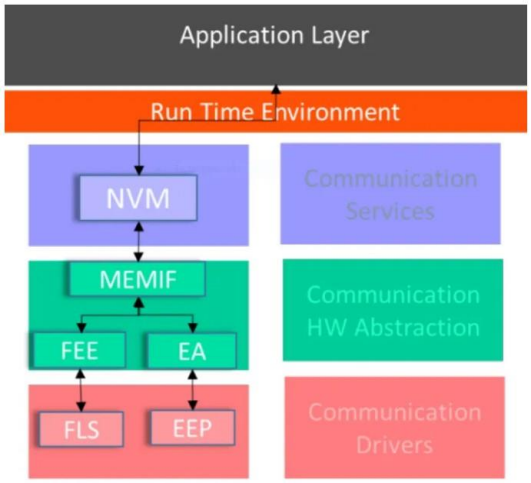

Figura 12: Diagrama dels serveis de memòria en la plataforma AUTOSAR Classic

Els blocs de memòria dinàmica poden trobar-se en diferents estats en funció de la seva validesa (vàlid o invàlid) o integritat de la informació (modificat / intacte). Els blocs vàlids es poden trobar intactes de forma que els seus continguts no han sigut modificats respecte al bloc de referència en memòria no volàtil, en cas contrari poden ser vàlids però modificats, això implica que els continguts del bloc de memòria han estat modificats, però la informació és reconeguda en els paràmetres definits.

Si no es compleix alguna d'aquestes condicions, es dona l'estat invàlid, intacte. Significant que la informació pot haver estat modificada fora dels paràmetres establerts i provocant una reescriptura de tots els blocs de memòria no volàtil a memòria volàtil i així garantir una transmissió correcta de totes les dades.

El mecanisme de sincronització entre la memòria volàtil i la no volàtil segueix un patró definit que evita la corrupció de la informació i minimitza el consum de recursos durant el traspàs d'informació. En quant l'aplicació necessita guardar informació en memòria no volàtil, aquesta actualitza la informació en memòria dinàmica que actua com a imatge de memòria no volàtil. És necessari que l'aplicació no utilitzi el bloc de memòria dinàmica mentre aquest està formant part de la comunicació d'informació. Per comprovar l'estat del procés i poder tornar a treballar sobre la comunicació cal esperar la marca de finalització de la comunicació de lectura o escriptura respectivament.

L'associació dels blocs de memòria pot ser fixa per mantindre un espai de memòria reservat i és el mètode que s'utilitza en aquest projecte. La configuració dels blocs de memòria a cada component ve determinada per la configuració prèvia amb el programari específic de la ECU.

L'avantatge principal d'aquest mètode on cada component controla el seu bloc associat de memòria dinàmica és que cada element pot realitzar el tractament de dades de forma eficient segons els requisits definits. Un altre aspecte rellevant és que al tenir un control limitat sobre aquest espai de memòria, es poden fer volcats de memoria de forma periòdica sense afectar els elements o informació de l'entorn. El bloc intermedi de memòria, serveix com a mesura de precaució contra la manipulació de la informació en el moment de ser extreta.

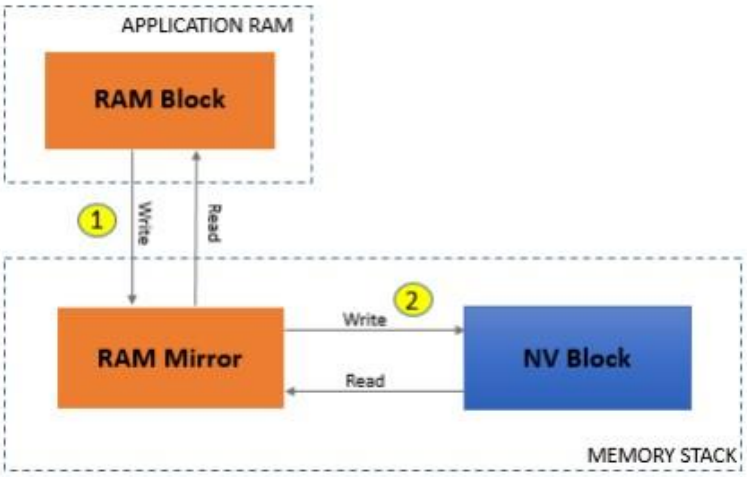

Figura 13: Diagrama de transmissió i sincronització de dades memòria Volàtil / No Volàtil

9. Registre d'esdeveniments de seguretat en el vehicle

Un cop s'ha introduït l'arquitectura i l'entorn del projecte, és possible descriure'l de forma més orgànica, desglossant-lo des dels conceptes més generals fins arribar als detalls del mateix.

9.1 Definicions i acrònims

En aquest apartat es trobaran els possibles elements que no es coneixen i necessiten de context per a seva comprensió.

9.1.1 Definicions

Software: Programari del sistema.

Hardware: Elements físics del sistema.

Void: Buit, absència d'informació o elemetns.

9.1.2 Acrònims

- SecurityLog / SecLog : Registre de seguretat
- ECU: Unitat de control externa
- RTE: entorn de temps real
- QSEv: esdeveniments qualificats com esdeveniments de seguretat
- SEv: esdeveniments pendents de qualificar

9.2 Descripció de funcionament general

El SecurityLog està dissenyat per ser un servei als diferents components de software de la ECU en la que està integrat.

Mitjançant funcions públiques per a la resta del sistema, s'encarrega de recaptar les peticions de registre de dades, classificar-les i emmagatzemar-les en una base de dades temporal. Aquesta, manté una quantitat definida dels esdeveniments més nous de cada tipus per ser tractats posteriorment.

De forma paral·lela, el sistema realitza crides periòdiques a un altre component del SecurityLog encarregat d'accedir a la base de dades temporals i obtenir l'esdeveniment més prioritari en cada cas. A continuació, processar-lo mitjançant uns algoritmes encarregats de filtrar la informació rellevant. Un cop ha sigut qualificat, és necessari emmagatzemar-lo garantint la integritat de la base de dades del vehicle a fi de mantenir el registre dels esdeveniments del vehicle. En cas que es detecti una possible intrusió el sistema actua protegint la informació existent per poder ser analitzada per un forense. Un altre servei que ofereix el projecte és el bolcat d'informació en cas de ser sol·licitat, aquesta funcionalitat solament és accessible pels components designats per poder mantenir la sincronització de dades entre les diferents ECU.

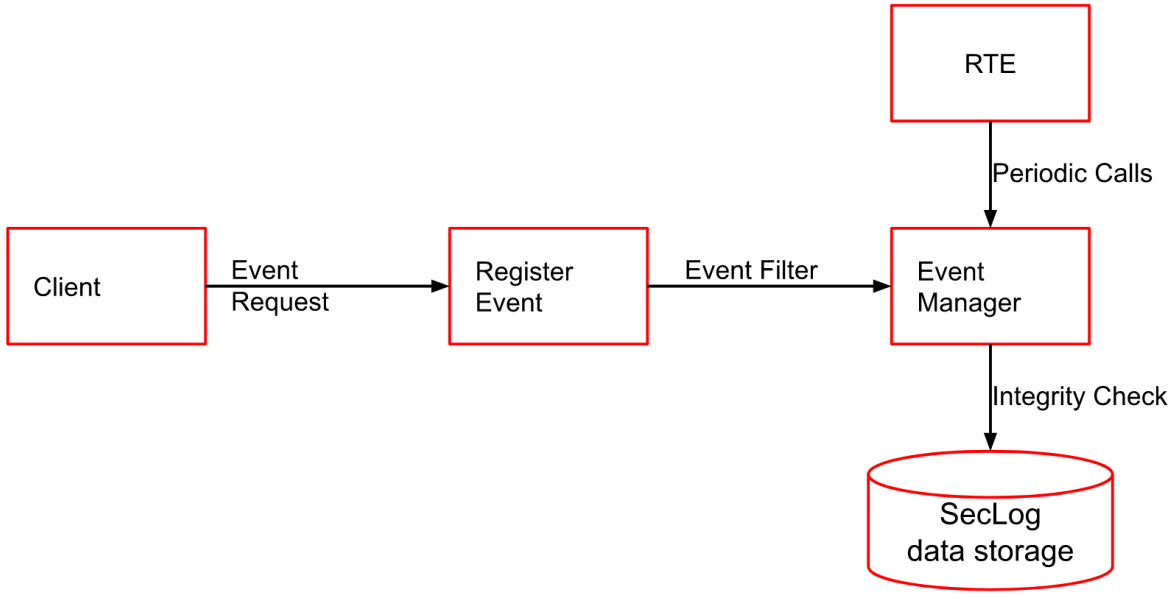

Figura 14: Diagrama del funcionament general del SecurityLog

9.3 Disseny del SecurityLog

9.3.1Sistema d'arxius

Els arxius del projecte es poden classificar en tres tipus principals, els arxius que contenen les funcions que proveeixen els diferents tipus de serveis del projecte, els arxius orientats a la gestió dels serveis del projecte i finalment els arxius orientats a la configuració del comportament i tipus de dades del projecte.

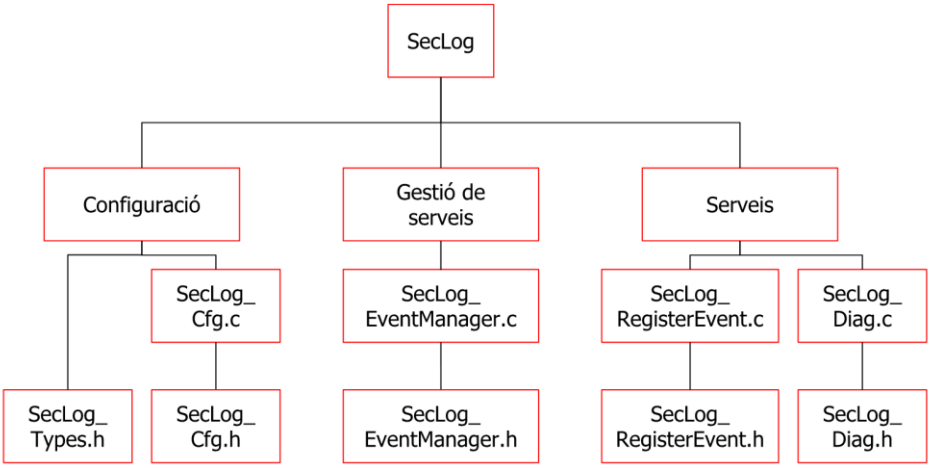

Figura 15: Diagrama classificació d'arxius SecurityLog

9.3.2 Anàlisi estàtic del SecurityLog

L'anàlisi estàtica del codi és un procediment utilitzat per trobar les diverses rutes que poden prendre les dades durant l'execució del codi. Aquest procés pot exposar problemes que porten a defectes crítics com problemes en l'emmagatzematge de dades. També s'utilitza per detectar problemes de seguretat que puguin alterar el comportament pel qual s'ha dissenyat el sistema.

Per poder analitzar les rutes de dades entre els diferents arxius del projecte cal entendre com es relacionen els components entre ells i quines funcions realitzen. En aquest projecte, els arxius es troben entrellaçats, creant una relació de dependència entre els diferents components. Els elements principals son els fitxers de configuració i definició, que especifiquen el comportament dels filtres d'esdeveniments per qualificar-los i paràmetres essencials d'aquests. L'altre element principal és l'encarregat de registrar la informació al component, ja que és on es troben les matrius de cues circulars encarregades de mantenir la informació rellevant durant el funcionament del SecurityLog.

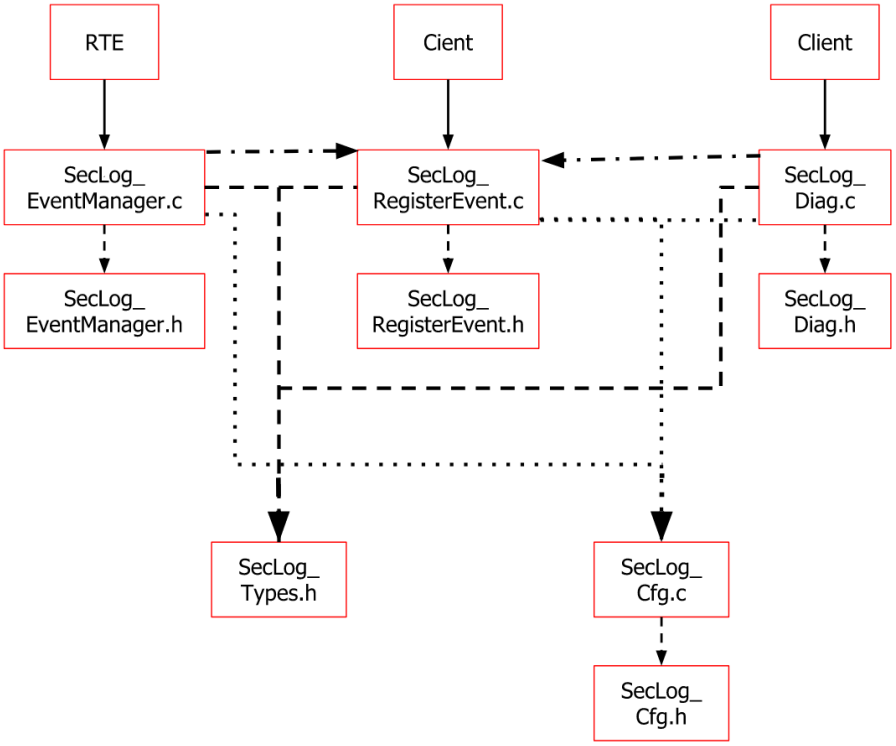

Figura 16: Diagrama de dependències del SecurityLog

9.3.2.1 Components

A continuació es troben les descripcions dels diferents components amb les explicacions detallades de la seva funció:

9.3.2.1.1 SecLog_Cfg

El fitxer SecLog_Cfg.h conté totes les constants que defineixen el comportament del projecte, els paràmetres de les dimensions de les matrius de cues circulars com la quantitat màxima d'esdeveniments es pot emmagatzemar, la quantitat de codis d'identificació (ID) inclús els valors que configuren els filtres i varien el seu funcionament.

L'arxiu SecLog_Cfg.c defineixen els tipus de filtres implementats en el sistema i s'associen a una matriu de configuració que conté totes les especificacions que han de complir cadascun dels esdeveniments definits. Per millorar el temps de registre, s'ha implementat la definició dels possibles esdeveniments que es trobarien en un entorn real, per no haver de realitzar un intercanvi de dades exhaustiu sinó que es pugui recol·lectar la informació de memòria utilitzant com a paràmetre la informació rebuda per l'exterior.

Els diferents filtres que es poden trobar en el SecurityLog segueixen les especificacions de l'arquitectura AUTOSAR i s'utilitzen per avaluar si cadascun dels esdeveniments es pot tractar com esdeveniments qualificats pel sistema o han de ser descartats. A continuació es mostra el diagrama de decisions que prenen els filtres i posteriorment s'explicaran de forma detallada.

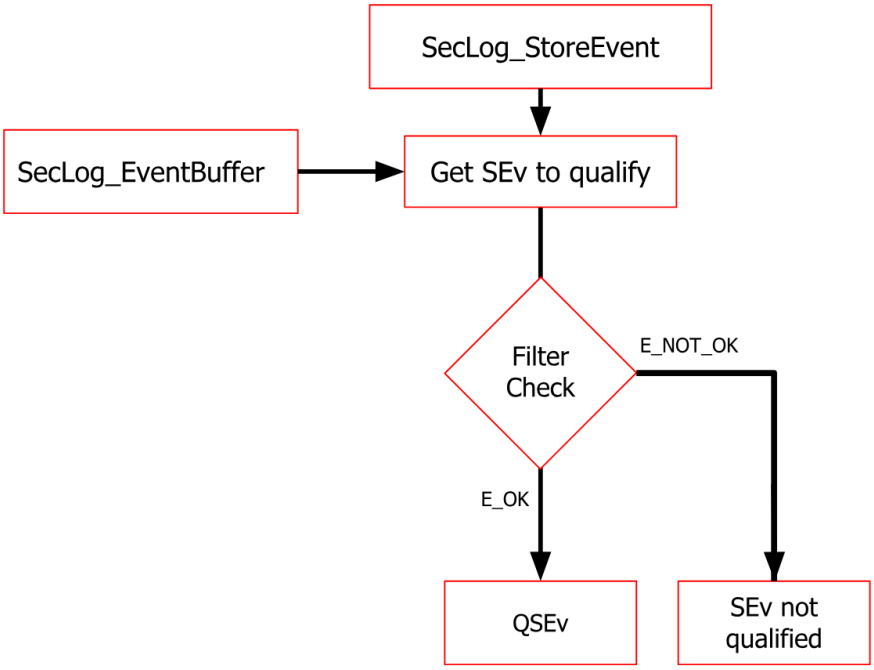

Figura 17:Diagrama de procés d'avaluació dels esdeveniments de seguretat

Els primers dos filtres són el SecLogBypassfilter i SecLogBlockFilter i com al seu nom indica, s'associen a esdeveniments que són d'alta prioritat o rarament succeeixen de forma que es poden qualificar directament com a esdeveniments de seguretat o, per altra banda, poden ser associats als esdeveniments que es vol evitar emmagatzemar, ja que com el registre de seguretat es pot implementar en diferents components al vehicle, cap a la possibilitat que no sigui necessari emmagatzemar tots els tipus d'esdeveniments en totes les ECUs del vehicle.

El filtre SecLogThresholdFilter s'utilitza per emmagatzemar un nou esdeveniment un cop ja n'han succeït una certa quantitat d'aquest. Aquesta característica permet no perdre informació rellevant quan es tracta d'esdeveniments que poden succeir de forma repetida o que són de baixa rellevància i únicament és necessari portar un control més flexible.

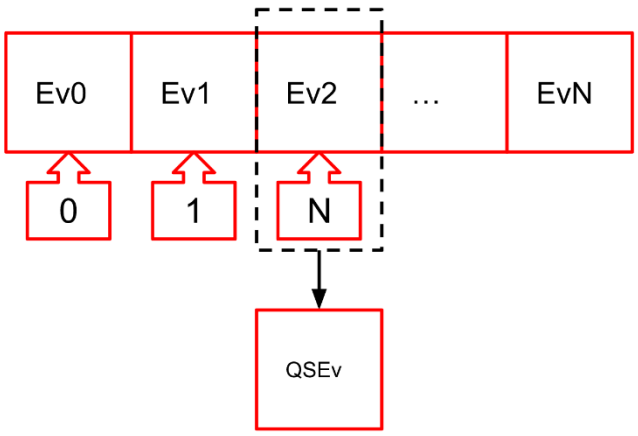

Figura 18: Representació del mètode d'avaluació del SecLogThesholdFilter

L'últim filtre implementat s'anomena *SecLogTimestampFilter*, s'encarrega de comprovar els esdeveniments tant per quantitat com per temps. Els esdeveniments s'avaluen individualment, però s'agafa una mostra de la quantitat definida i solament són qualificats són aquells que compleixen les següents condicions.

En primer lloc, els esdeveniments han de complir un temps mínim entre cadascun d'ells definit per les especificacions del client, en cas de no ser així, per l'empresa. En segon lloc, la quantitat d'esdeveniments de la mostra ha de complir un temps mínim des del primer fins a l'últim per entrar dins les especificacions que se li exigeix al component.

Aquest sistema de filtrar s'ha implementat amb la finalitat que si es produeix una intrusió al sistema de dades, els intrusos intentin sobreescriure la informació en la base de dades. Amb aquest filtre s'alliberaria la informació redundant que arribaria en intervals de temps molt curts per intentar aconseguir sobreeixir les dades emmagatzemades i ocultar la intrusió.

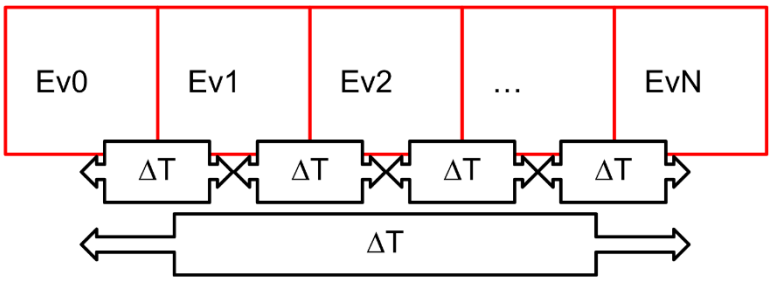

Figura 19: Representació del mètode d'avaluació SecLogTimestampFilter

9.3.2.1.2 SecLog_RegisterEvent

El *SecLog_RegisterEvent*, com indica el nom, es l'encarregat d'introduir els nous esdeveniments en un *buffer* de memòria on s'emmagatzemarà de forma temporal fins que sigui el seu torn de ser processa.

Es comprova, en primer lloc si el sistema es troba en l'estat d'admetre noves entrades de dades, si es dona el cas, a partir del codi identificador de l'esdeveniment, es classifica directament a partir del mètode definit dins de l'estàndard.

El *buffer* temporal, anomenat *SecLogEventBuffer* segueix una estructura en forma de matriu de dues dimensions. Aquesta matriu està dividida per files, a les que corresponen cadascun dels codis d'identificació (ID). Aquestes files es tracten com cues circulars individuals on es manté una quantitat especificada dels últims esdeveniments de cada tipus. Per tractar millor els elements de la matriu, cada codi identificador té associat un comptador extern en un vector de dades que és d'utilitat per aplicacions posteriors, també serveix per assignar el valor a la variable *SecLogEvCount.*

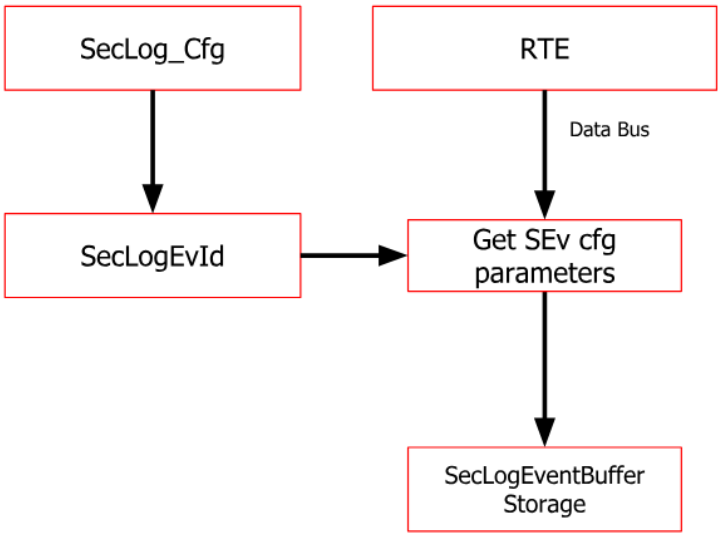

Figura 20: Diagrama procés de registre d'esdeveniments del SecLogRegisterEvent

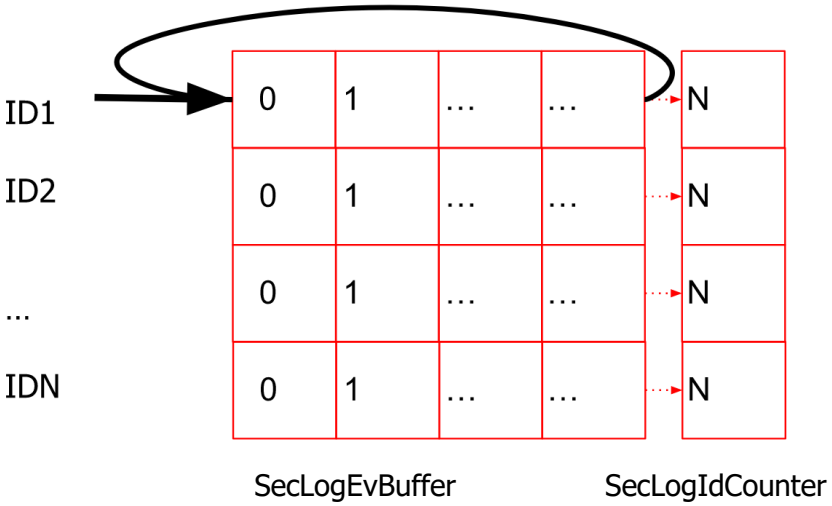

Figura 21: Diagrama de detall de l'estructura de dades del SecLogRegisterEvent

9.3.2.1.3 SecLog_EventManager

És component responsable de la gestió dels esdeveniments, processar-los i emmagatzemar-los preservant la seva integritat. Per millorar la gestió de la informació i poder processar les dades d'entrada i les de memòria de forma paral·lela, aquest component es troba dins d'una rutina periòdica dins del sistema RTE de l'empresa. Aquest entorn gestiona les crides al *SecLog_EventManager*.

Dintre d'aquest arxiu podem trobar la rutina d'inicialització que es crida una única vegada, cada vegada que el sistema s'inicia per configurar els paràmetres del codi segons especificacions. També es troba la referència al procés d'emmagatzematge de dades.

La rutina responsable d'emmagatzemar les dades és el *SecLog_StoreEvent*. Quan s'inicia, s'accedeix al valor més antic de cada codi identificador de la matriu de dades temporal per comprovar si compleix els requisits sol·licitats.

En el moment en què ha estat seleccionat l'esdeveniment per avaluar, s'accedeix al punter de la funció de filtrat. Els filtres agafaran la mostra dels esdeveniments segons en el codi d'identificació. Si es qualifica l'esdeveniment, es passarà a la protecció de la integritat.

Per garantir la integritat de la base de dades es sol·liciten els serveis de criptografia de l'arquitectura AUTOSAR. Actuen com funcions condicionals i envien la informació a processar al nucli dedicat a la computació criptogràfica. Una vegada retorna el resultat, si l'operació ha resultat fallida, el codi activa l'actuació contra intrusions i bloqueja l'entrada de nous esdeveniments per preservar la informació existent. En cas que l'operació hagi sigut exitosa, es procedeix a guardar l'esdeveniment sobre la matriu de dades. Finalment, es calcula de nou els paràmetres de seguretat per la pròxima iteració.

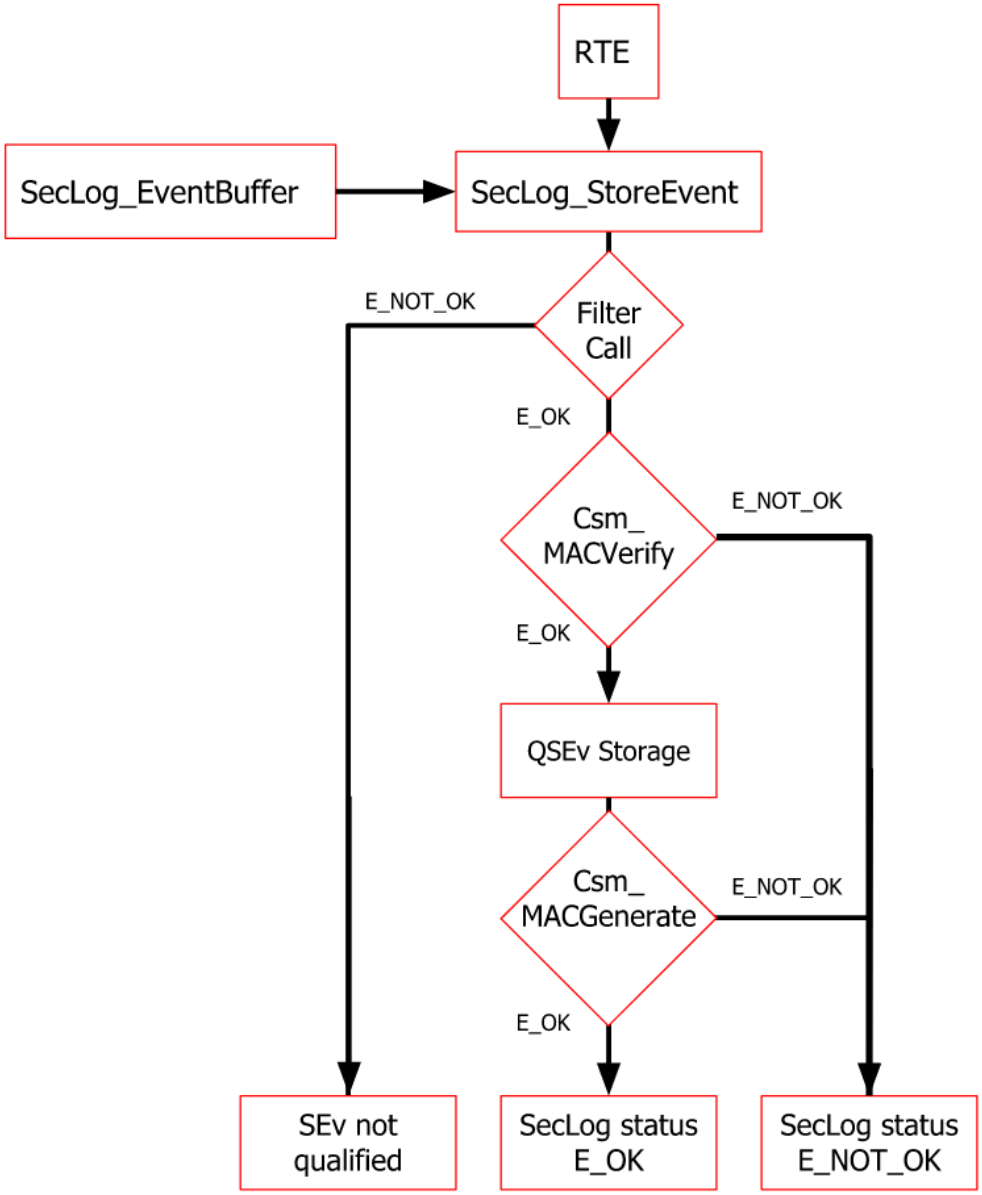

Figura 22: Diagrama de funcionament SecLog_StoreEvent

El sistema de seguretat criptogràfic es basa en un sistema de xifrat AES128 (*Advanced Encryption Standard*). Per minimitzar el temps de càlcul criptogràfic i disminuir la càrrega del processador dedicat, el qual solament és capaç de fer un procés, s'ha decidit utilitzar aquest mètode de xifrat per blocs. L'algoritme és de xifrat simètric, és a dir, s'usa la mateixa clau per xifrar i desxifrar el contingut, és el procés de xifrat òptim pels recursos i requisits que té el projecte a causa de la seva rapidesa, seguretat i eficiència.

L'algoritme transforma un conjunt de dades a bits classificats dins una matriu. Un cop la informació s'ha descompost i barrejat amb la clau, se segueix una sèrie d'iteracions on es multipliquen les matrius d'informació i s'alternen l'ordre de les files i columnes elaborant un resultat final de xifratge de 128 bits de mida.

Per garantir la integritat, es calcula a cada crida de rutina la codificació AES128 a partir de la informació a verificar. Com aquest xifratge es calcula a partir de la informació vinculada, si el resultat no coincideix amb el paràmetre calculat anteriorment, implica que la informació ha estat modificada per un agent extern.

9.3.2.1.4 SecLog_Diag

El servei de diagnòstic està reservat en el sistema als components que requereixen de sincronització de dades com per exemple a altres ECUs. La funció d'aquest element és la de realitzar una còpia de la informació existent en el moment que es sol·licita. Per aconseguir aquest objectiu, per seguretat es fa una comprovació a la integritat de la informació abans de fer el bolcat de dades. Aquesta comprovació és redundant però entra dins els requisits de l'empresa ja que seria crític pel sistema dur a terme una sincronització entre les ECUs amb informació corrupta.

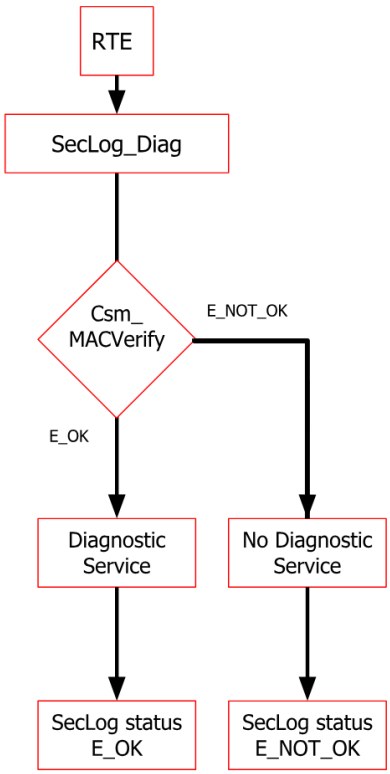

Figura 23: Diagrama de funcionament SecLog_Diag

9.3.2.2 Estructures de dades

Per organitzar totes les dades que es manipulen en aquest projecte, són necessàries una sèrie d'estructures que les emmagatzemin. Encara que les dimensions puguin variar en funció de l'entorn en que s'implementi, la prioritat ha estat generar el mínim ús de memòria possible.

Els components que formen la base de dades són tres matrius de dues dimensions, una principal (SecLog), que es copia en memòria, reté la informació del sistema. La secundària (SecLogEventBuffer) que es troba en memòria dinàmica i s'utilitza com a pas intermedi per classificar, organitzar i tractar la informació abans de passar a formar part de la matriu principal i la tercera (SecLogEvId), encarregada de contenir els paràmetres de configuració del sistema.

Per gestionar aquestes estructures de dades, s'implementen una sèrie de vectors que contenen informació auxiliar. L'encarregat d'emmagatzemar el conjunt d'esdeveniments que s'avaluen quan el sistema ho requereix (SecLogEventFilterScope), on s'emmagatzemen els punters a les funcions filtre (SecLogFilterAsign), els comptadors que gestionen la quantitat d'esdeveniments de cada tipus i els encarregats de contenir els paràmetres criptogràfics com la clau (SecLogCsmKey) i el valor generat pel procés de xifrat (SecLog_CMAC).

9.3.2.3 Descripció de funcionament global

Un cop han sigut descrits tots els elements del SecurityLog, s'han presentat les bases per descriure el funcionament del conjunt en l'entorn en el qual està dissenyat per operar. A continuació es troba una descripció breu que recull tots els conceptes mencionats fins al moment, per una comprensió adequada de la globalitat del projecte.

En quant un client crida al servei de registre d'esdeveniments de seguretat, el SecLog_RegisterEvent agafa la informació configurada per aquell SEv de la memòria de configuració i . El fet de predefinir els possibles esdeveniments en la configuració no només agilitza el procés d'entrada de noves dades, sinó que també garanteix una entrada de dades dintre dels paràmetres especificats. De forma paral·lela, el sistema crida de forma cíclica al SecLog_EventManager, que a la vegada gestiona les crides al SecLog_StoreEvent, encarregat de recollir l'últim esdeveniment de cada tipus emmagatzemat en la memòria dinàmica per qualificar-lo. Si els esdeveniments compleixen els requisits, seran aptes per emmagatzemar-se en memòria no volàtil i passar a formar part de la base de dades de la ECU. Per arribar a aquest últim nivell, és necessari comprovar que la informació no ha estat modificada per cap agent extern. Un cop es confirma, s'incorpora la nova informació i es generen nous paràmetres de seguretat per la informació actualitzada. De no complir els requisits de seguretat, el sistema es bloqueja amb el fi de preservar la informació i no permetre la possible entrada de dades corruptes.

Per finalitzar l'anàlisi estàtic del projecte, es troben dos diagrames globals del SecurityLog on es representen les crides entre els components en un cas real i una representació de l'organització en memòria dels elements que componen el conjunt.

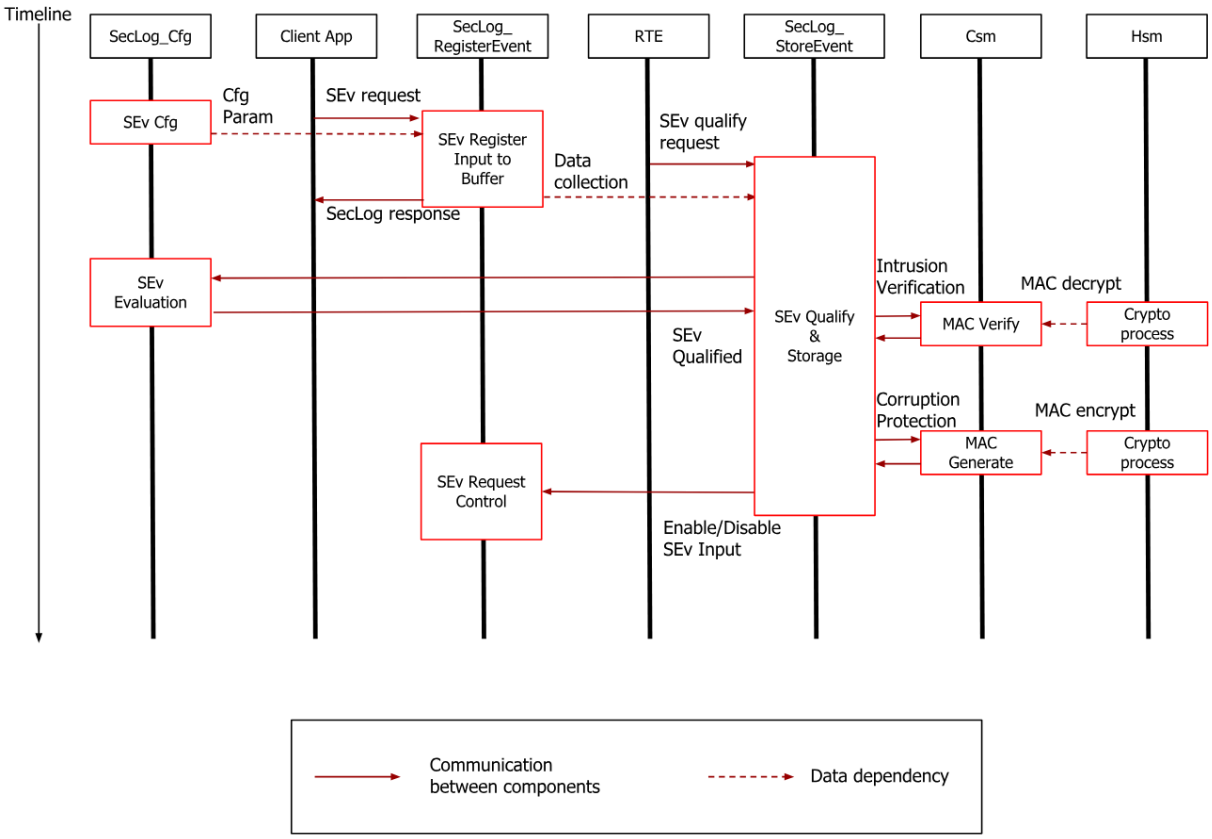

Figura 24: Diagrama de la interacció entre els components del SecurityLog simulant un cas real.

9.3.2.4 Diagrama complet SecrityLog

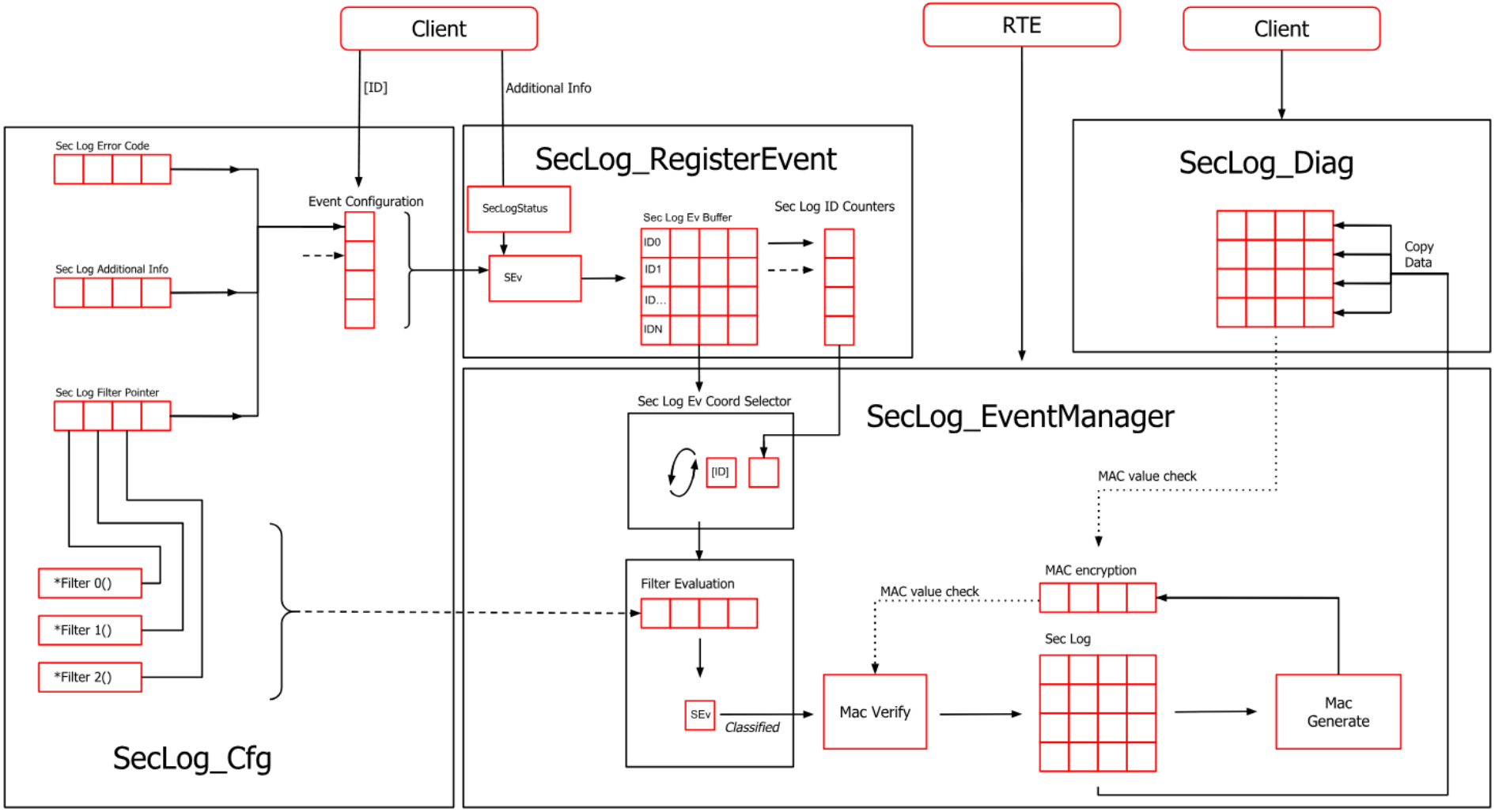

Figura 25: Diagrama complet SecurityLog, il·lustra la com es relacionen els components i com està organitzada la informació de la base de dades.

9.3.3 Anàlisi dinàmic del SecurityLog

Un cop s'han descrit els components del registre de seguretat i com es comporten per separat, es passarà a elaborar un anàlisi dinàmic del projecte. El mètode seguit per verificar el correcte funcionament del projecte segueix els estàndards de l'empresa amb un joc de proves anomenat *Unit Test* on s'intenta presentar totes les situacions possibles en les quals es pot trobar el codi en un entorn real.

9.3.3.1 Entorn de proves

Les proves del projecte s'han realitzat sobre un microprocessador real subministrat per l'empresa. A causa a factors externs al projecte, l'empesa ha necessitat redistribuir els recursos com el material designat als projectes, juntament amb el termini de l'estada a l'empresa, han sorgit una sèrie de complicacions.

En primer lloc, no s'ha pogut realitzar l'anàlisi dinàmic del xifrat d'informació ja que va ser l'últim element a incorporar-se al projecte. Les proves realitzades no utilitzen els elements condicionals de ciberseguretat encara que estiguin implementats. Per poder completar la resta de proves s'ha realitzat un bypass donant per aprovades les comprovacions realitzades pel servei extern al projecte.

En segon lloc, en el moment en què ha sigut possible dur a terme l'anàlisi dinàmica del xifrat de dades, la configuració en la programació del microcontrolador encastat no era compatible amb el servei criptogràfic. Les dues opcions disponibles eren reconfigurar la programació del microcontrolador o canviar de hardware. Cap de les opcions ha sigut possible dins del marge de temps restant a l'empresa i, per tant, implica l'aparició d'una última complicació.

Les proves dinàmiques del servei d'emmagatzematge, on es realitza una còpia del bloc de memòria dinàmica a memòria no volàtil, han estat descartades per falta de temps.

Ja que el sistema no està connectat a un entorn de temps real, però el sistema està associat a una crida periòdica del programa de l'empresa dins del microcontrolador. Per les proves en funció del temps, s'utilitza una variable que s'incrementa a cada crida del sistema de forma que escalant-la amb el valor del període de crides, podem tractar els valors en funció del temps de forma discreta, per aquestes proves són intervals de deu mil·lisegons.

10. Conclusions del projecte

El Registre d'esdeveniments de seguretat, és un projecte que no només té el propòsit de ser, sinó que està dissenyat per formar part d'un element real en el món de l'automoció, un component que estarà implementat en les ECUs dels vehicles de nova generació.

Els objectius del projecte són la capacitat de registrar els esdeveniments de seguretat generats en un vehicle, classificar-los, qualificar-los mitjançant una sèrie d'algorismes sincronitzats entre ells i emmagatzemar-los garantint la integritat dels mateixos usant els serveis de l'arquitectura en la qual es troba integrat.

Al llarg d'aquests mesos en l'estada a Lear, s'han desenvolupat els algorismes i estructures de dades necessàries pel compliment d'aquests objectius. Mitjançant proves i reestructuracions del projecte, eventualment integrals, s'ha arribat a la solució actual que compleix amb els requisits claus de l'empresa i el client en un entorn totalment desconegut com és el món de l'automoció i l'arquitectura AUTOSAR.

Finalment, després d'haver finalitzat el període en l'empresa i observant els resultats obtinguts amb el projecte, no només es poden extreure una evolució en el mètode de treball i creixement personal sinó un component amb uns serveis com són la classificació i gestió d'esdeveniments de seguretat. Encara que no ha sigut possible implementar els últims objectius que vinculen el projecte amb els serveis de criptografia i memòria no volàtil, s'ha creat una base ferma per la creació d'un component de programari que pot continuar desenvolupant.

11. Referències

1.Adaptive AUTOSAR vs Classic AUTOSAR: Which way is the automotive industry leaning? (2020, mayo 15). Embitel; Embitel Technologies. https://www.embitel.com/blog/embedded-blog/adaptive-autosar-vs-classic-autosar

2.Aurum. (2019, marzo 27). *Chris Valasek & Charlie Miller entrevista*. Aurum Speakers Bureau. https://www.aurumbureau.com/es/entrevista-con-chris-valasek-charlie-miller/

3.*Automotive intrusion detection systems*. (s/f). Vector Informatik GmbH. Recuperado el 9 de junio de 2023, de https://www.vector.com/int/en/know-how/security/automotive-intrusion-detection-systems/

4.Classic platform AUTOSAR. (s/f). Autosar.org. Recuperado el 9 de junio de 2023, de https://www.autosar.org/standards/classic-platform

5.*Document Status*. (s/f-a). *Document title NV data handling guideline*. Autosar.org. Recuperado el 9 de junio de 2023, de https://www.autosar.org/fileadmin/standards/R22-11/CP/AUTOSAR_EXP_NVDataHandling.pdf

6.*Document Status*. (s/f-b). *Document title specification of crypto interface*. Autosar.org. Recuperado el 9 de junio de 2023, de https://www.autosar.org/fileadmin/standards/R22-11/CP/AUTOSAR_SWS_CryptoInterface.pdf

7.*Foundation AUTOSAR*. (s/f). Autosar.org. Recuperado el 9 de junio de 2023, de https://www.autosar.org/standards/foundation

8.Reichart, G. (2023, abril 18). *Home AUTOSAR*. Autosar.org. https://www.autosar.org/

9.Ruiz, S. (2020, julio 9). *Los 25 casos más importantes de ciberataques a vehículos*. HackerCar; Cybentia. https://hackercar.com/los-25-casos-mas-importantes-de-ciberataques-a-vehiculos/

10.Standards. (s/f). Autosar.org. Recuperado el 9 de junio de 2023, de https://www.autosar.org/standards

11.Zauner, C. (2023, marzo 30). *Software-defined Vehicle: repercusiones en la red de a bordo*. MD ELEKTRONIK. https://www.md-elektronik.com/es/software-defined-vehicle-swdv-y-sus-repercusiones-en-la-red-de-a-bordo/

12.(S/f). Nxp.com. Recuperado el 9 de junio de 2023, de https://www.nxp.com/docs/en/product-brief/HSEPB.pdf