
This is the **published version** of the bachelor thesis:

Sardà Martín, Òscar. *Desenvolupament del Mòdul Software de Gestió i Autenticació SAS per a un Receptor GNSS Galileo*. Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026 (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/334449>

under the terms of the  license.

Desenvolupament del Mòdul de Gestió i Interfície d'Usuari per a un Receptor GNSS amb Autenticació de Senyal Galileo (SAS)

Òscar Sardà Martín

29 de juny de 2026

Resum– Històricament, l'evolució de les tecnologies de navegació global per satèl·lit (GNSS) s'ha centrat sobretot en la precisió i la qualitat del servei, però no s'havia plantejat gaire la ciberseguretat. En els darrers anys, i empès pel context geopolític actual, el focus s'ha ampliat cap a l'autenticació dels senyals com a defensa davant atacs com el *spoofing*. En aquesta línia, Galileo va desplegar primer l'autenticació a nivell de dades amb l'*Open Service Navigation Message Authentication* (OSNMA), operatiu des de 2025, i posteriorment està introduint l'autenticació a nivell de senyal amb el *Signal Authentication Service* (SAS), encara en fase de proves. Aquest treball presenta el desenvolupament del mòdul de gestió, la interfície d'usuari i el processament criptogràfic i de dades auxiliars d'un receptor SAS. El sistema s'integra dins la biblioteca de codi obert SASlib, desenvolupada dins del grup de recerca SPCOMNAV. El sistema implementa una arquitectura modular de tres capes que orquestra els tretze blocs de la cadena de processament SAS. El mòdul s'ha validat amb senyal E6-C xifrat real capturat durant la campanya de juny de 2025 i també en la campanya actual, en què s'ha xifrat el senyal dels satèl·lits 14 i 18 de Galileo. A més, s'ha desenvolupat una web de monitoratge de l'estat del servei SAS dels satèl·lits Galileo, que executa el *pipeline* de SASlib de manera periòdica en una plataforma dins de l'Escola d'Enginyeria i en permet comprovar el funcionament continu.

Paraules clau– Galileo, GNSS, autenticació de senyal, SAS, OSNMA, anti-spoofing, receptor SDR, *pipeline* de processament.

Abstract– Historically, the evolution of Global Navigation Satellite System (GNSS) technologies has focused mainly on accuracy and quality of service, with little attention paid to cybersecurity. In recent years, however, driven by the current geopolitical context, the focus has broadened towards signal authentication as a defence against attacks such as *spoofing*. Along these lines, Galileo first deployed data-level authentication with the *Open Service Navigation Message Authentication* (OSNMA), operational since 2025, and is now introducing signal-level authentication with the *Signal Authentication Service* (SAS), still in its testing phase. This work presents the development of the management, auxiliary-data processing, cryptographic processing and user-interface module of a SAS receiver, integrated into the open-source library SASlib, developed within the SPCOMNAV research group. The system implements a modular three-layer architecture that orchestrates the thirteen blocks of the SAS processing chain. The module has been validated with real encrypted E6-C signal captured during the June 2025 campaign and also in the current campaign, in which the signal of Galileo satellites 14 and 18 has been encrypted. In addition, a monitoring website has been developed that runs the SASlib pipeline on a platform within the School of Engineering and allows verifying that the service is continuously operational.

Keywords– Galileo, GNSS, signal authentication, SAS, OSNMA, anti-spoofing, SDR receiver, processing pipeline.



1 INTRODUCCIÓ I CONTEXT

EL sistema global de navegació per satèl·lit (GNSS) s'ha convertit en un servei transversal del qual depenen sectors com l'energia, les telecomunicacions, el transport o les finances, tant per al posicionament com per a la sincronització temporal. Aquesta dependència, però, conviu amb una debilitat estructural del sistema: el senyal obert té un format públic i arriba a la superfície amb molt poca potència, cosa que fa possible generar-ne i emetre'n una rèplica falsa amb maquinari assequible. Mitjançant aquesta tècnica, coneguda com a *spoofing*, un atacant pot induir el receptor a calcular una solució de posició, velocitat i temps (PVT) errònia sense que l'usuari se n'adoni [8]. La gravetat i l'accessibilitat creixents d'aquests atacs han impulsat el desenvolupament de mecanismes capaços de verificar l'autenticitat del senyal rebut [1, 2].

En aquest escenari, Galileo ha adoptat una estratègia d'autenticació en dues capes que actuen sobre aspectes diferents del senyal. La primera, l'Open Service Navigation Message Authentication (OSNMA), treballa a nivell de dades: protegeix la integritat del missatge de navegació i es troba en servei des de 2025 [3]. La segona, el Signal Authentication Service (SAS), treballa a nivell de rang: permet comprovar que les mesures de distància provenen realment del satèl·lit legítim i encara es troba en fase de proves [4]. La combinació de totes dues capes situa Galileo com l'únic GNSS que ofereix protecció simultània davant el *spoofing* de dades i de senyal.

El grup de recerca SPCOMNAV de la Universitat Autònoma de Barcelona ha contribuït des de les fases conceptuals fins a la validació experimental d'aquestes solucions [5, 6, 7]. Fruit d'aquest treball, el grup ha desenvolupat SASlib, una biblioteca Python de codi obert que implementa la cadena completa de processament d'un receptor SAS [2], de la qual l'estudiant és coautor.

Aquest TFG hi aporta cinc contribucions principals: (i) l'arquitectura modular de tres capes i el motor d'execució dirigit pels esquemes JSON, que orquestra els tretze blocs del receptor; (ii) la cadena de gestió de dades auxiliars i de processament criptogràfic —descàrrega i parseig dels fitxers RECS, BGD i SLOG, gestió de claus TESLA i desxifrat de les seqüències re-xifrades—; (iii) la interfície gràfica web per configurar, executar i monitoritzar el *pipeline*; (iv) la integració amb el mòdul de processament de senyal, validada amb senyal E6-C xifrat real; i (v) l'estació web de monitoratge que n'executa el *pipeline* de manera periòdica i en publica l'estat de funcionament.

La resta de l'article s'organitza de la manera següent. La Secció 2 presenta l'estat de l'art de l'autenticació a Galileo. La Secció 3 defineix els objectius i l'abast del treball. La Secció 4 descriu la metodologia i l'arquitectura del sistema. La Secció 5 exposa i discuteix els resultats obtinguts. Finalment, la Secció 6 en recull les conclusions i les línies de treball futures.

2 ESTAT DE L'ART

La generació i emissió d'un senyal GNSS fals és avui factible amb maquinari de baix cost, de manera que qualsevol receptor convencional pot ser enganyat [8]. El *spoofing* —una forma d'interferència intencionada que induïx el receptor a calcular una posició o un temps erronis, a diferència del *jamming*, que en denega el servei— s'ha convertit en una amenaça de primer ordre, agreujada pel context geopolític actual [1]. Les defenses contra aquesta amenaça es classifiquen habitualment en dues famílies: les tècniques *no criptogràfiques*, que detecten l'atac al receptor sense alterar el senyal, i les *criptogràfiques*, que hi introdueixen informació impredecible per autenticar-lo [8].

2.1 Defenses contra el spoofing

Les tècniques no criptogràfiques no modifiquen el senyal: exploten manifestacions físiques per detectar l'atac al receptor, com ara el control automàtic de guany (AGC) i la potència rebuda, la consistència dels temps d'arribada (RAIM), el monitoratge de la qualitat del senyal (SQM) i la detecció de múltiples pics de correlació (MPD), l'angle d'arribada amb diverses antenes, la polarització o les mesures inercials [8]. Són econòmiques i no requereixen canvis al senyal, però tenen un abast limitat davant atacs sofisticats. Les tècniques criptogràfiques introdueixen informació impredecible al senyal i actuen a dos nivells:

- a *nivell de dades*, l'autenticació del missatge de navegació —Navigation Message Authentication (NMA)— impedeix falsificar les dades de navegació;
- a *nivell de rang* o de senyal, l'autenticació del codi d'eixamplament —Spreading Code Authentication (SCA)— en xifra els *chips* i impedeix fins i tot que l'atacant faci el seguiment del senyal [8].

2.2 Autenticació a Galileo: OSNMA i SAS

L'OSNMA és el primer servei civil de tecnologies GNSS que implementa defenses criptogràfiques a nivell de dades: autentica el missatge de navegació de la banda E1-B mitjançant cadenes de claus Timed Efficient Stream Loss-tolerant Authentication (TESLA), és d'accés lliure i està orientat al mercat de masses [9]. Els seus components principals són la cadena de claus TESLA, el codi d'autenticació Message Authentication Code and Key (MACK), el Digital Signature Message (DSM) signat amb Elliptic Curve Digital Signature Algorithm (ECDSA) i un arbre de Merkle per validar la clau pública [10]; el servei inicial es va declarar operatiu el 2025 [3].

El SAS complementa l'OSNMA amb autenticació a nivell de rang (SCA) i es basa en el xifrat del senyal pilot E6-C. Al segment de terra, fragments del senyal xifrat (les Encrypted Code Sequence (ECS)) es tornen a xifrar amb una clau encara no revelada, i donen lloc a les seqüències re-xifrades (Re-Encrypted Code Sequence (RECS)), que inclouen l'instant d'emissió i es publiquen al servidor del Galileo Service Centre (GSC) abans de l'emissió del senyal [4]. El receptor les desxifra —un cop l'OSNMA en revela la clau— per reconstruir localment les ECS i autenticar el rang, procés que es detalla a la Secció 2.3 [7]. Com

• E-mail de contacte: oscarsarda566@gmail.com

• Menció: Tecnologies de la Informació — NIU: 1631879

• Treball tutoritzat per: Rafael Terris-Gallego (Departament de Telecomunicació i Enginyeria de Sistemes)

• Curs 2025/26

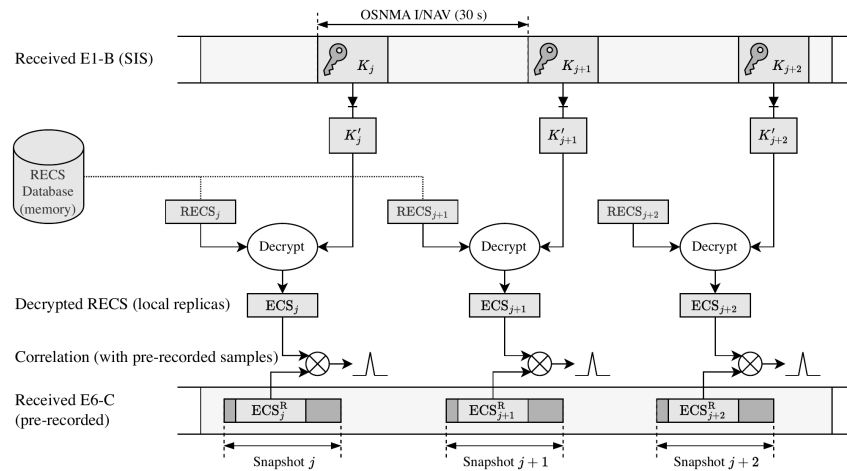


Fig. 1: Operació del receptor SAS al llarg del temps (adaptat de [16]).

que la clau de desxifrat prové de l'OSNMA, el receptor no ha d'emmagatzemar cap clau secreta, fet que evita el maquinari resistent a manipulacions i en redueix el cost [8]. El Global Positioning System (GPS) segueix una estratègia anàloga amb Chips-Message Robust Authentication (CHIMERA), que combina l'autenticació de dades i de rang en un únic servei a la banda L1-C; Galileo, en canvi, ofereix dos serveis diferenciats —OSNMA per a les dades i SAS per al rang— i lidera així el desplegament de l'autenticació GNSS a tots dos nivells [8].

2.3 Funcionament del receptor SAS

La Figura 1 il·lustra l'operació del receptor SAS al llarg del temps [16]. El procés combina dades pre-enregistrades i material revelat en viu, i es pot resumir en quatre passos:

1. *Captura.* El receptor pre-enregistra instantànies (*snapshots*) del senyal pilot E6-C xifrat (part inferior de la figura).
2. *Dades auxiliars.* Descarrega del GSC les RECS —les ECS re-xifrades— i les manté en memòria.
3. *Revelació de la clau.* Quan l'OSNMA revela a l'E1-B la clau TESLA K_j (cada 30 s), el receptor en deriva la clau de desxifrat K'_j .
4. *Desxifrat i correlació.* Amb K'_j desxifra la RECS i n'obté la rèplica local ECS, que correlaciona amb les mostres E6-C pre-enregistrades; un pic nítid de correlació autentica el rang del satèl·lit.

Aquests quatre passos es corresponen directament amb les etapes de la cadena de processament que implementa aquest treball (Secció 4): captura, gestió de dades auxiliars, gestió de claus i, finalment, desxifrat i correlació.

2.4 Receptors GNSS per programari i treballs relacionats

Un receptor GNSS definit per programari captura el senyal en banda base i en delega el processament —adquisició, seguiment, correlació i càlcul de la solució— a codi executat en un ordinador de propòsit general, en lloc de maquinari

dedicat [11]. Aquesta aproximació aporta flexibilitat i reproductibilitat: permet modificar els algorismes i repetir els experiments sobre senyal enregistrat sense canviar el maquinari, cosa que la fa idònia per prototipar i validar mecanismes d'autenticació encara en definició com el SAS.

Sobre aquesta base han sorgit biblioteques obertes especialitzades. *osnmalib* és la biblioteca de referència per a l'OSNMA [12, 13]. Pel que fa a l'autenticació a nivell de senyal, el grup SPCOMNAV ha contribuït de manera sostinguda a la definició i la validació del SAS —originàriament conegut com a ACAS— amb estudis de modes d'operació i requisits, guies d'implementació i plataformes de prova [14, 5, 6, 15, 16, 8]; SASlib és la materialització en programari de codi obert d'aquesta línia de treball [2].

3 OBJECTIUS I ABAST

Objectiu principal. Desenvolupar el mòdul de programari que orquestra el flux de dades i la interfície d'usuari d'un receptor GNSS capaç de processar el servei SAS de Galileo, integrant l'autenticació a nivell de dades (OSNMA) i de senyal (SAS) en un sistema coherent, usable i configurable.

Objectius específics.

- Dissenyar una arquitectura modular per blocs i el model de dades compartit (*ExecutionContext*) amb esquemes de configuració JSON.
- Implementar el motor d'execució (*PipelineRunner*: execució seqüencial, gestió d'estats, *logging* i control d'errors).
- Desenvolupar els blocs de gestió de dades auxiliars i de processament criptogràfic (descàrrega i parseig RECS/BGD, claus TESLA, desxifrat de l'ECS).
- Desenvolupar la interfície gràfica web (configuració, visualització del *pipeline*, monitoratge i import/export de configuracions).
- Integrar el mòdul amb el TFG de Telecomunicacions i validar el sistema amb dades reals del servei SAS.

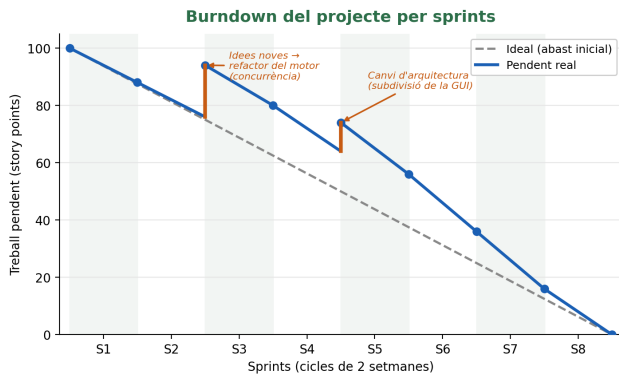


Fig. 2: Burndown del projecte (*story points* per *sprint*), seguit amb Linear.

- Desenvolupar una estació web de monitoratge que executi el *pipeline* de manera periòdica i en publiqui l'estat de funcionament.

Abast. Formen part de l'abast d'aquest TFG l'arquitectura i el motor d'execució, la gestió de dades auxiliars i el desxifrat RECS/ECS, la interfície gràfica, la integració amb el mòdul de senyal i la documentació. En queda fora, com a responsabilitat del TFG de Telecomunicacions, el processament de senyal de radiofreqüència (adquisició i seguiment E1/E6, correlació i generació de pseudodistàncies).

4 METODOLOGIA I PROCÉS DE DESENVOLUPAMENT

4.1 Metodologia de desenvolupament

El projecte ha seguit una metodologia de desenvolupament àgil iterativa, amb *sprints* de dues setmanes, adaptada al context d'un TFG. La tria es justifica perquè el projecte depenia de fonts externes i evolutives: la dependència amb l'equip i amb el TFG de Telecomunicacions —el bloc de correlació de senyal no està disponible fins a la meitat del projecte—, l'evolució activa de les especificacions del servei SAS, la disponibilitat incerta de les dades de producció del Galileo Service Centre i l'aparició d'idees i requisits nous durant el desenvolupament que van fer evolucionar l'arquitectura de la biblioteca. Una aproximació en cascada hauria requerit fixar tots els requisits abans de començar, cosa inviable en aquest escenari, i hauria posposat la validació fins al final.

Cada iteració segueix el cicle de planificació, disseny de les interfícies del component, implementació, verificació contra vectors de prova oficials i revisió amb el tutor. La gestió de tasques s'ha dut a terme amb Linear —organitzada en cicles i *issues* per fase—, en lloc de l'eina prevista inicialment, pel seu suport natiu a *sprints*, *backlog* i estimacions. La Figura 2 mostra el *burndown* del projecte per *sprints*: el treball pendent —en *story points*— al llarg dels vuit cicles. Els trams en què el treball pendent puja corresponen a ampliacions d'abast quan van sorgir idees noves que van modificar l'arquitectura (la refactorització del motor i la subdivisió de la interfície), un comportament propi d'una aproximació àgil i no en cascada.

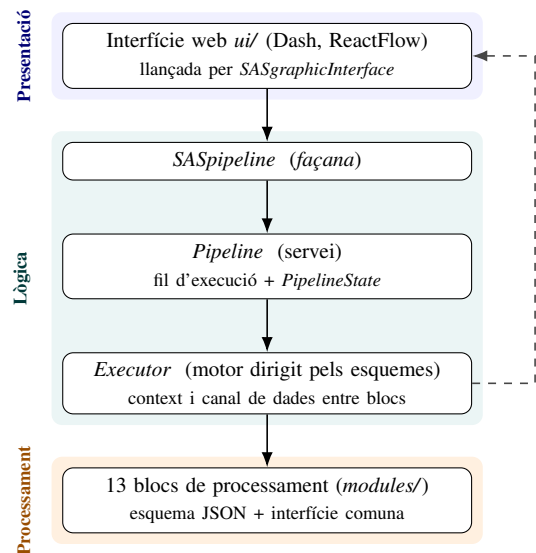


Fig. 3: Arquitectura de programari de tres capes.

4.2 Arquitectura del sistema

El sistema s'estructura en tres capes (Figura 3) amb responsabilitats clarament separades, en la línia de les arquitectures de receptors GNSS definits per programari [11]. La capa de presentació és una aplicació web (paquet *ui/*, arrencada des de *SASgraphicInterface*) que es comunica amb la resta del sistema únicament a través d'una façana. La capa de lògica és el motor d'orquestració i es descompon, al seu torn, en tres peces: la façana *SASpipeline*, que declara els blocs i el seu ordre i n'exposa una instància única a la interfície; el servei *Pipeline*, que posseeix el fil d'execució i l'estat observable; i el motor *Executor*, dirigit pels esquemes JSON. La capa de processament és el conjunt de tretze blocs independents que comparteixen una mateixa interfície.

Sobre aquesta estructura s'apliquen diversos patrons de disseny. El patró *Pipeline* estructura el flux SAS com una cadena ordenada de blocs, però l'execució no és estrictament seqüencial: alguns blocs es poden executar en paral·lel i d'altres només s'activen sota certes condicions, segons el que declara cada esquema, fet que permet encavalcar tasques independents i reduir el temps total d'execució (Secció 4.3). Una interfície de bloc comuna en permet la intercanviabilitat i el *testing* unitari de cada bloc per separat. El patró *facade* (*SASpipeline*) aïlla la interfície de la complexitat del motor. Finalment, el motor publica els esdeveniments del cicle de vida de cada bloc —inici, finalització i error— mitjançant *callbacks* (patró *Observer*) als quals se subscriu el servei *Pipeline*; aquest els consolida en un estat compartit i segur entre fils (*PipelineState*) que la interfície Dash refresca per sondeig periòdic.

4.3 Motor d'execució i context d'execució compartit

El motor central (*Executor*) està dirigit pels esquemes JSON: cada bloc declara al seu esquema quines dades consumeix i quines produeix, i exposa la mateixa interfície d'execució —la interfície de bloc comuna esmentada més amunt. El motor encadena els blocs segons aquestes declaracions: en construeix les entrades a partir d'un context

compartit i comprova en temps d'execució que cada bloc produeix exactament les sortides que ha declarat. L'esquema també pot indicar que un bloc s'executi en segon pla o només sota una certa condició; per exemple, la cadena de descàrrega i desxifrat de les RECS només s'activa quan el satèl·lit objectiu està xifrat.

El canvi de disseny més rellevant respecte de la concepció inicial afecta el model de dades compartit. En un primer moment, tots els blocs intercanviaven dades a través d'un únic objecte global (un *singleton*). En incorporar l'execució concurrent d'alguns blocs, concentrar tota la comunicació en un únic objecte mutable es va revelar fràgil —apareixien condicions de carrera i un acoblament excessiu— i es va optar per separar responsabilitats: el pas de dades entre blocs es va traslladar al canal dirigit pels esquemes; el *singleton* es va reduir a identificar l'execució i gestionar-ne les rutes de sortida; i l'estat que el motor comparteix amb la interfície (els *logs* i els resultats de cada bloc) es va aïllar en un component propi i segur entre fils (*PipelineState*). Aquesta reorganització ha eliminat la causa de les condicions de carrera, ha reduït l'acoblament i ha simplificat els tests d'integració.

4.4 Blocs de processament i el seu flux

La cadena de processament SAS consta de tretze blocs agrupats en quatre etapes funcionals —arrencada, captura, generació de l'ECS i processament de senyal i autenticació—, il·lustrades a la Figura 4 [2]. A continuació es descriuen les quatre etapes.

Les tres primeres etapes són, en la seva pràctica totalitat, aportació d'aquest TFG:

- els mòduls d'*arrencada* preparen el receptor abans de l'emissió del senyal: inicialitzen la configuració (*ConfigInit*), obtenen la referència temporal i la seva incertesa via NTP (*TimeReferenceSynchronizer*), descarreguen i parsegen l'almanac per estimar la visibilitat dels satèl·lits (*AlmanacManagement*) i descarreguen del GSC —en un arxiu TAR— i parsegen els fitxers RECS, BGD i SLOG seguint l'especificació del SAS, d'on recuperen els bits de les RECS, els valors de Broadcast Group Delay (BGD) i l'estat del servei [4];
- els mòduls de *captura* estimen la finestra de mostres necessària (*UncertaintyModule*) i comanden l'enregistrament de la instantània E6-C (*SnapshotRecording*);
- els mòduls de *generació de l'ECS* constitueixen el nucli criptogràfic: *TESLAkeyManagement* recupera la clau TESLA un cop revelada a l'E1-B via OSNMA —a través d'un servidor *osnmalib.eu*— i *RECSDecryption* aplica Advanced Encryption Standard (AES) en mode Cipher Block Chaining (CBC), segons l'Interface Control Document (ICD) vigent del SAS [4], sobre les RECS per obtenir les seqüències desxifrades (ECS), que serveixen de rèpliques locals per a la correlació.

La quarta etapa —*processament de senyal i autenticació*— combina el processament de senyal de radiofreqüència, a càrrec del TFG de Telecomunicacions, amb el veredict d'autenticació. El bloc *SignalCorrelation* correlaciona les mostres E6-C enregistrades amb les ECS i en treu

les observables (retard de codi, Doppler i carrier-to-noise density ratio (C/N_0)); a continuació, *EphemeridesManagement* descarrega i parseja les efemèrides (RINEX NAV) i en calcula les elevacions, i *SatelliteAuthenticator* —aportació d'aquest TFG— aplica el criteri d'autenticació sobre les observables: compara el pic de correlació amb el soroll de fons (o la relació pic primari/secundari) i verifica la diferència de retard de codi respecte del senyal auxiliar [4]. Finalment, *PVTComputation* en calcula la solució PVT autenticada.

L'enregistrament de la instantània s'interfície, a través d'una Application Programming Interface (API) dedicada, amb una plataforma ràdio definida per programari (SDR) multibanda basada en dues plaques BladeRF ubicada a la UAB.

4.5 Cadena criptogràfica

El nucli criptogràfic del receptor és la reconstrucció de les seqüències desxifrades (ECS) a partir de les RECS i de la clau TESLA revelada per l'OSNMA. El procediment, repartit entre els blocs *TESLAkeyManagement* i *RECSDecryption*, segueix l'ICD vigent del SAS [4].

Un cop l'OSNMA revela i verifica la clau TESLA K_j de la subtrama a l'E1-B, *TESLAkeyManagement* en recupera el valor i *RECSDecryption* en deriva la clau de desxifrat aplicant-hi SHA-256:

$$K'_j = \text{SHA256}(K_j), \quad (1)$$

una clau AES de 256 bits. El vector d'inicialització es construeix a partir de l'instant Galileo System Time (GST) de la subtrama:

$$IV = \text{trunc}_{128}(\text{SHA256}(P_j)), \quad (2)$$

$$P_j = \text{GST}_{\text{SF},j} \parallel \text{RAND} \parallel \mathbf{0}_{88}, \quad (3)$$

on trunc_{128} pren els 128 bits més significatius, $\parallel$ denota la concatenació, $\text{GST}_{\text{SF},j}$ és el GST de l'inici de la subtrama (32 bits), RAND és l'indicador d'aleatorització de la RECS (8 bits) i $\mathbf{0}_{88}$ és un farciment de 88 zeros. Finalment, cada RECS es desxifra amb AES-256 en mode CBC amb la clau i el vector anteriors:

$$\text{ECS}_j = \text{AES256}_{\text{CBC}}^{-1}(K'_j, \text{RECS}_j, IV). \quad (4)$$

Les seqüències ECS_j obtingudes —a les quals s'elimina el farciment de zeros final quan escau— són les rèpliques locals que es correlacionen amb les mostres E6-C pre-enregistrades. Aquest mateix farciment serveix de comprovació: només surt net quan la clau i el vector d'inicialització són correctes, fet que permet validar la cadena criptogràfica sense disposar encara del senyal correlat.

4.6 Model de blocs dirigit pels esquemes

Tots els blocs comparteixen la mateixa estructura, fet que en facilita la intercanviabilitat i el manteniment: una funció *run_pipeline* que conté la lògica i un esquema JSON que en descriu el comportament de manera declarativa. L'esquema és la font única de veritat del bloc i desacobla l'especificació dels paràmetres de la implementació de l'algoritme [2]. Hi figuren tres tipus d'informació: les claus que el bloc consumeix i produeix, que defineixen el flux de dades amb la

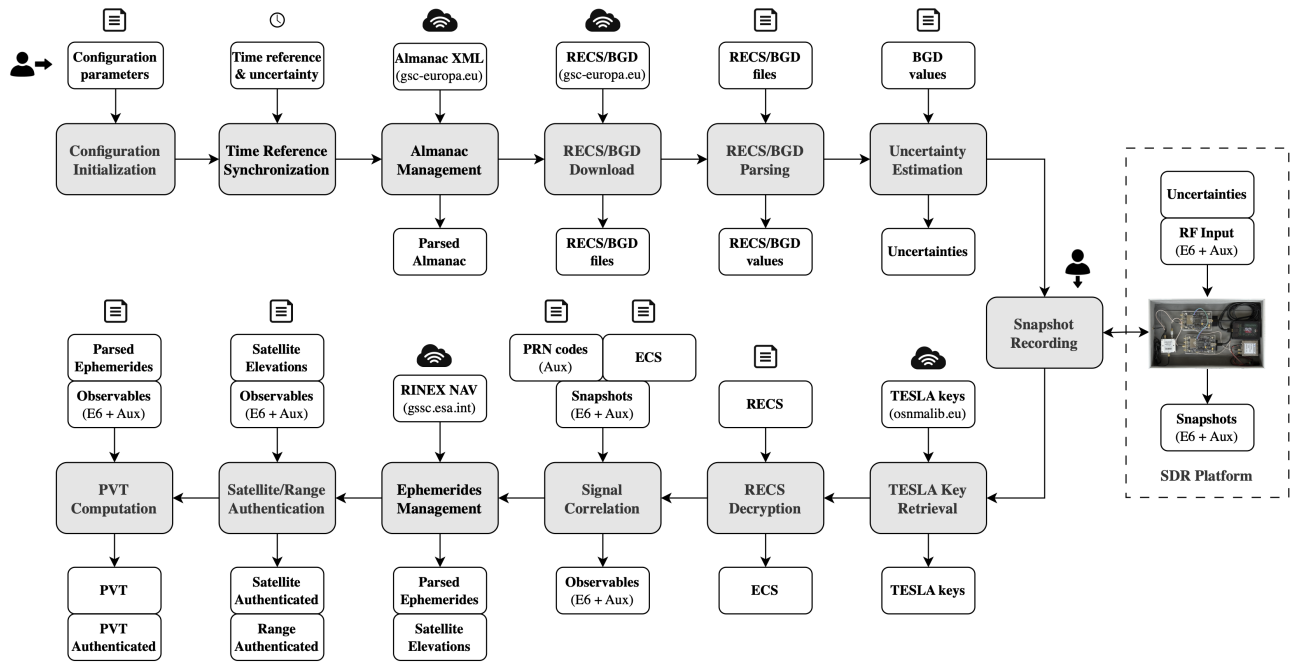


Fig. 4: Arquitectura i mòduls de SASlib (adaptat de [2]).

resta de blocs; les metadades d'execució (*parallel_with* i *required_when*), que en regulen la concurrència i l'activació condicional; i la llista de paràmetres configurables, organitzats en seccions, cadascun amb un identificador, un tipus, un valor per defecte, un rang vàlid, una descripció i el tipus de *widget* amb què es presenta.

A partir d'aquesta única declaració, el sistema deriva automàticament tres coses. Primer, el motor d'execució encadena els blocs segons les claus *consumes/produces* i en valida el contracte en temps d'execució (Secció 4.3). Segon, cada valor introduït es valida contra el tipus i el rang declarats. I tercer, la interfície gràfica genera el formulari de configuració del bloc, amb el control adequat per a cada paràmetre. Aquesta estratègia dirigida pels esquemes implica que afegir un paràmetre —o fins i tot un bloc sencer— només requereix editar-ne l'esquema, sense tocar ni el motor ni la interfície. Més endavant se'n mostra un exemple simplificat (Figura 6).

La configuració es gestiona a dos nivells interoperables [2]: un fitxer de configuració *global* del *pipeline*, que recull els paràmetres de tots els blocs juntament amb metadades d'execució (identificador, marca de temps i versió) i que, en carregar-se, es distribueix als fitxers de configuració de cada bloc; o bé l'edició directa d'aquests fitxers per bloc. Tots dos formats es poden convertir l'un en l'altre. A més, cada execució desa els paràmetres i les sortides en directoris amb marca de temps, de manera que es preserva l'estat complet de la prova i se'n garanteixen la traçabilitat i la reproduïbilitat; una configuració completa es pot exportar i importar en format JSON.

4.7 Interfície gràfica

La interfície és una aplicació web basada en Dash i React-Flow (paquet *dash-diagram*). Internament s'estructura en quatre subcapes amb dependències unidireccionals, de la base cap amunt:

- el *nucli* (*core*): els tipus, l'estat de la interfície i la

configuració; no té cap dependència interna;

- els *serveis* (*services*): l'única frontera amb el *backend* —embolcallen la façana *SASpipeline*—, a més de la persistència i la serialització de les configuracions;
- els *components* (*components*): les peces visuals —el graf ReactFlow, les finestres modals, els formularis generats a partir de l'esquema i les gràfiques de resultats;
- els *callbacks* (*callbacks*): la lògica reactiva de Dash, que connecta els components amb els serveis.

Aquesta separació en capes manté la interfície desacoblada del motor: només la capa de serveis hi accedeix, sempre a través de la façana *SASpipeline*.

Funcionalment, la interfície representa el *pipeline* com un graf de blocs connectats, cadascun enllaçat en temps d'execució amb la seva funció del *backend*, i en mostra l'estat —pendent, en execució, completat o error— (Figura 5), que actualitza sondejant periòdicament l'estat compartit *PipelineState*. En seleccionar un bloc s'obre una finestra modal amb el formulari generat a partir del seu esquema, els controls d'execució i els *logs* corresponents, que es mostra en gran a l'Apèndix A.2 (Figura 10). La interfície permet executar blocs individuals o el *pipeline* complet, inclou un panell de monitoratge amb els *logs* en temps real i les mètriques d'autenticació per satèl·lits.

4.8 Integració amb el TFG de Telecomunicacions i *osnmalib*

El mòdul de gestió s'integra amb dos components externs que no reimplementa: la implementació d'OSNMA del grup de recerca i el mòdul de processament de senyal del TFG de Telecomunicacions.

D'una banda, la implementació del protocol OSNMA —cadena de claus TESLA, verificació ECDSA, MACK i arbre de Merkle— és preexistent del grup (biblioteca *osnmalib*). SASlib no la reescriu: el bloc *TESLAkeyManagement*

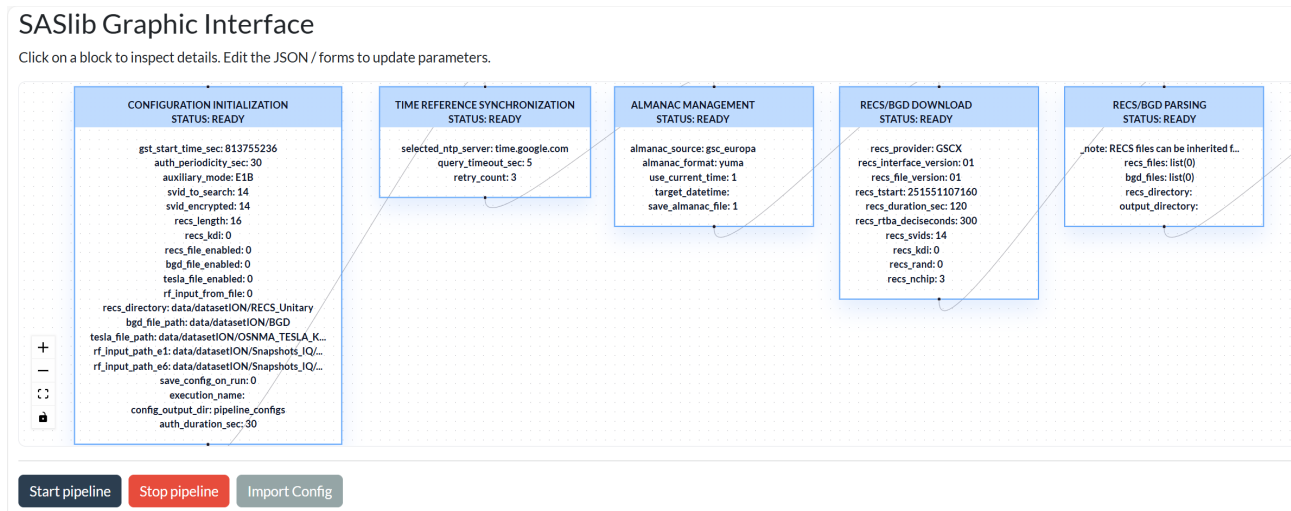


Fig. 5: Interfície gràfica de SASlib (de [2]).

n'obté les claus TESLA revelades —a través d'un servidor *osnmalib*— per desxifrar les RECS.

D'altra banda, el processament de senyal de radiofreqüència (correlació E1/E6, pseudodistàncies i càlcul de PVT) és responsabilitat del TFG de Telecomunicacions. La interfície entre tots dos mòduls s'ha resolt mitjançant fitxers intermedis: l'intercanvi de les pseudodistàncies E6 es realitza a través de fitxers, cosa que desacobla els dos mòduls i en permet la validació independent. Aquesta decisió ha facilitat la detecció de problemes a la frontera d'integració i la verificació de la compatibilitat de formats entre tots dos sistemes.

4.9 Estació de monitoratge

Com a contribució addicional respecte a l'abast inicial s'ha desenvolupat una estació de monitoratge independent, una aplicació Flask que executa el *pipeline* de SASlib de manera periòdica, cada deu minuts.

La seva finalitat és oferir una visió contínua de l'estat del servei SAS al llarg de la constel·lació, de manera anàloga a la plataforma *osnmalib* per a l'OSNMA. A cada execució processa la cadena completa per als satèl·lits visibles i en

publica el resultat en una interfície web. En aquesta portal hi ha informació com quins satèl·lits emeten l'E6-C xifrat i quins autèntiquen el rang, un *skyplot* amb les traces dels satèl·lits, les observables per banda i el veredict de l'autenticació de cada satèl·lit, i l'historial recent d'execucions.

Així, l'estació compleix tres funcions. Primer, permet comprovar que el servei SAS funciona de manera contínua i detectar quan nous satèl·lits comencen a emetre l'E6-C xifrat. Segon, actua com a verificació operativa de la pròpia biblioteca: en executar la cadena completa de manera desatésada cada deu minuts, qualsevol regressió o canvi del servei es fa visible immediatament. I tercer, en reaprofitar la mateixa cadena de processament que SASlib però aïllant-ne la capacitat de monitoratge com a servei independent, valida que el *pipeline* és apte per a una execució autònoma i sostinguda. L'estació s'ha desplegat en una plataforma dins de l'Escola d'Enginyeria, on executa el servei de manera contínua.

5 RESULTATS I DISCUSSIÓ

5.1 Entorn i dades de validació

El sistema s'ha validat amb senyal real capturat amb la plataforma SDR BladeRF a les bandes E1 i E6, amb una freqüència de mostreig de 20 Msps. S'han emprat dos conjunts: primerament les instantànies de la campanya de senyal E6-C xifrat el juny de 2025 (*datasetION*) [7] i les de la campanya actual —instantànies de 5 s enregistrades simultàniament a E1 i E6—, en què s'ha xifrat el senyal dels satèl·lits 14 i 18 de Galileo. A l'Apèndix A.1 s'inclou la configuració completa d'un processament del PRN 18, que en fixa tots els paràmetres de tots els blocs que es guarden automàticament segons un opció del bloc inicial.

5.2 Verificació de la cadena de processament

El sistema integra els tretze blocs de la cadena de processament SAS. Cada bloc s'ha verificat de manera independent contra un conjunt de dades de referència; tot seguit, la cadena completa s'ha validat sobre senyal real en quatre etapes.

```

{
  "block_name": "RECS Decryption",
  // claus que el bloc llegeix del context compartit
  "consumes": ["tesla_keys", "parsed_recs_files"],
  // claus que publica per als blocs següents
  "produces": ["ecs_hex", "ecs_path"],
  // metadades que interpreta el motor d'execució
  "execution": {
    "parallel_with": ["SnapshotRecording"],
    "required_when": "globals.svid_encrypted"
  },
  // cada secció i camp genera el formulari de la GUI
  "sections": [
    {
      "title": "TESLA key selection",
      "fields": [
        {
          "id": "manual_osnma_key", // nom del paràmetre
          "type": "string",         // tipus -> validació
          "component": "input",     // widget de la GUI
          "default": null
        }
      ]
    }
  ]
}
  
```

Fig. 6: Esquema JSON simplificat d'un bloc (*RECSDecryption*).

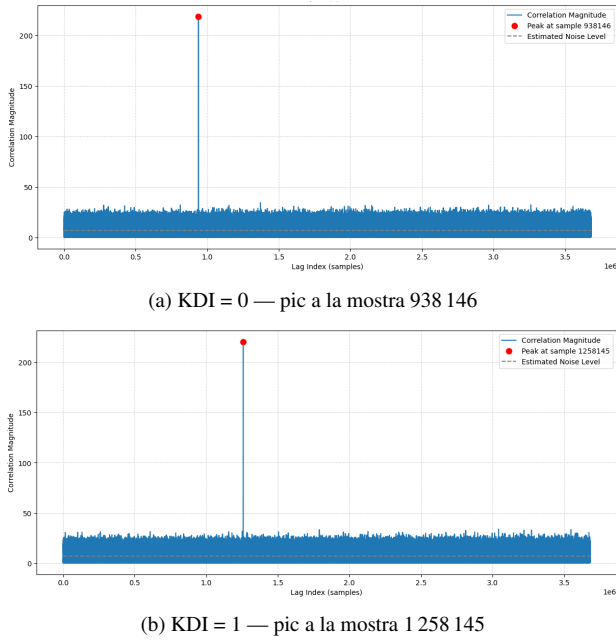


Fig. 7: Pics de correlació E6-C per a les dues configuracions de KDI.

- *Desxifrat i integritat de l'ECS.* S'ha comprovat que el desxifrat reconstrueix correctament les seqüències ECS: per a les RECS de 16 ms, el text desxifrat presenta un conjunt de zeros al final de la seqüència esperada (Secció 4.5), fet que confirma que la clau TESLA i el vector d'inicialització derivats són correctes abans de cap correlació.
- *Detecció dels pics d'ECS.* La correlació de les mostres E6-C del conjunt de referència de la campanya de 2025 (*datasetION*) amb les ECS produeix pics nítids als retards de codi i les freqüències Doppler esperats per a les dues configuracions de l'índex de retard de clau (KDI), amb una separació entre pics de 16 ms —320 k mostres a 20 Msps— coherent amb l'especificació [4] i amb les plataformes de referència [2, 7] (Figura 7).
- *Comparació amb el senyal auxiliar.* La comparació del pic E6-C amb el del senyal públic E1-B permet mesurar l'alineament dels retards de codi, base del criteri d'autenticació. Per al PRN 14 del *datasetION* s'obté un error de rang de 0,697 mostres (10,45 m) entre tots dos pics (Figura 8), valor que el mòdul d'autenticació compara amb el lllindar [2].
- *Autenticació en la campanya actual (PRN 18).* S'ha aplicat la cadena completa sobre una captura pròpia de la campanya actual, en què el satèl·lit E18 i E14 emet l'E6-C xifrat (instantànies de 5 s a 20 Msps, 23 de juny de 2026). Un cop aplicat correccions dels biaixos s'ha determinat una diferència entre l'E6-C desxifrat i l'E1-B públic és de $\Delta = +4,80$ m (+0,320 mostres), per sota del lllindar d'autenticació $\gamma_{\text{auth}} = 13,1$ m: el rang del PRN 18 queda autènticat. (Figura 9). Cal mencionar la prova també s'ha fet amb el PRN 14, resultant autèntic també.

Aquesta validació amb dades pròpies ha posat de manifest les dificultats reals d'un receptor SAS de tipus *snaps-hot*. La principal és la localització temporal de l'ECS dins de la instantània: la marca de temps del fitxer de captura estava esbiaixada i, juntament amb el salt de segon intercalar (GST = UTC +18 s), duia a correlar soroll; l'època correcta es va determinar amb un escombratge d'èpoques discriminant pel quocient pic primari/secundari (PPSP). A més, el biaix entre les mostres d'E1/E6 s'ha de corregir abans del test, fet que confirma la necessitat de la cadena de dades auxiliars (BGD) i de correcció ionosfèrica desenvolupada en aquest treball.

No s'ha desplegat encara una bateria de proves automàtiques que validi la cadena completa de manera contínua. D'una banda, definir un test universal *end-to-end* per a tota la cadena és complex; de l'altra, els blocs més senzills de provar són també els que rarament canvien, de manera que un error s'hi localitzaria i depuraria amb facilitat i automatitzar-ne la prova tindria un cost de còmput poc justificat. Per això s'ha prioritzat la validació amb dades reals; la integració de proves automàtiques a cada *push* es deixarà com a treball futur.

5.3 Interfície gràfica i monitoratge

A més s'ha pogut fer la interfície gràfica que representa el *pipeline* com un graf de blocs connectats i permet executar mòduls individuals o la cadena completa, seguir-ne el progrés i els *logs* en temps real i visualitzar els resultats d'autenticació per satèl·lit (Figura 5). A la pràctica, el formulari generat a partir de l'esquema de cada bloc evita editar JSON a mà i en valida els paràmetres, i la importació i exportació de configuracions completes fa reproduïbles els experiments —com el del PRN 18 de l'Apèndix A.1.

L'estació de monitoratge, desplegada a l'Escola d'Enginyeria, ha confirmat el funcionament sostingut de la cadena: cada deu minuts processa de manera autònoma els satèl·lits visibles i en publica l'estat —taula de la constel·lació, *skyplot*, observables i veredictes per satèl·lit, i historial—, que es mostra en gran a l'Apèndix A.3 (Figures 11 i 12). En el moment d'aquest treball hi consta el PRN 14 com a autènticat pel SAS, coherent amb la validació de rang descrita més amunt.

5.4 Discussió

Els resultats cobreixen els objectius plantejats a la Secció 3: la cadena de processament SAS funciona de manera integrada i, a partir de senyal E6-C xifrat real, produeix rangs autènticats —tant sobre el conjunt de referència de 2025 com sobre una captura pròpia de la campanya actual (PRN 18)—, amb resultats consistents amb les plataformes de referència [2, 7]. Això confirma la viabilitat de l'arquitectura dirigida pels esquemes i de la cadena de gestió, dades auxiliars i desxifrat desenvolupades.

Cal emmarcar-ne, però, les limitacions. Primer, l'autenticació a nivell de posició encara no és possible: només dos satèl·lits (E14 i E18) emeten l'E6-C xifrat i el càlcul d'una PVT autènticat en requereix quatre o més; la funcionalitat queda implementada per a quan la constel·lació xifrada creixi. Segon, la validació amb dades pròpies depèn de paràmetres temporals precisos —l'època de l'ECS i el segon

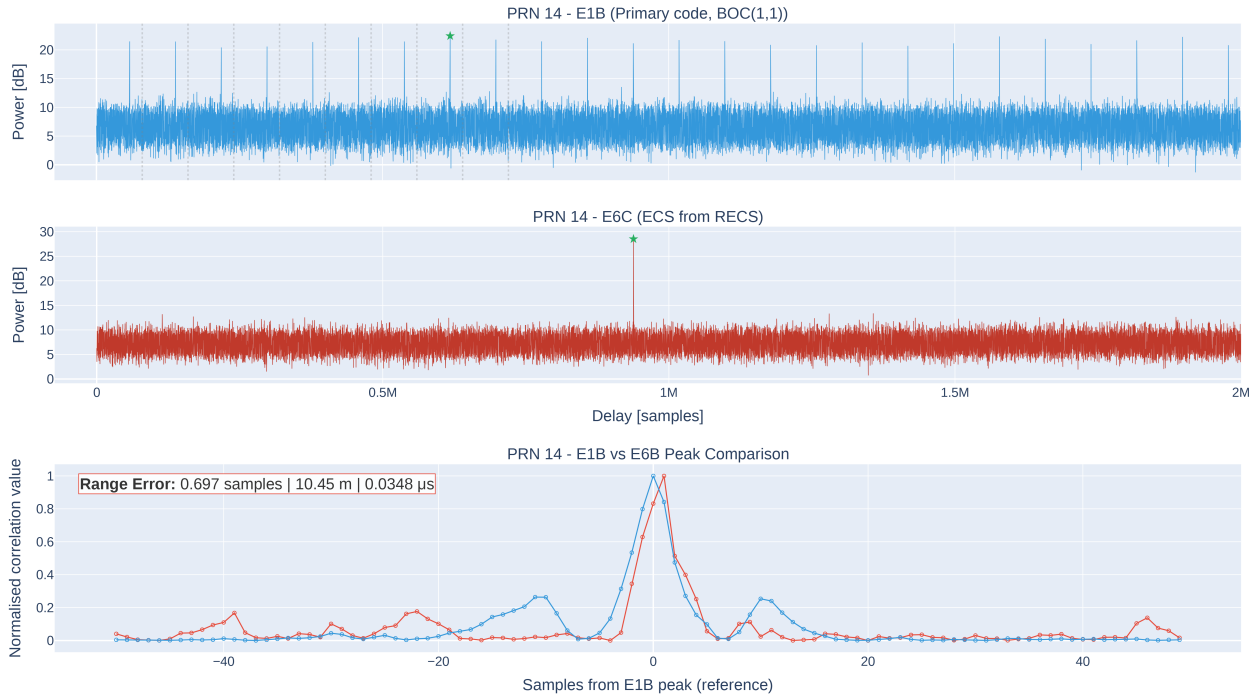
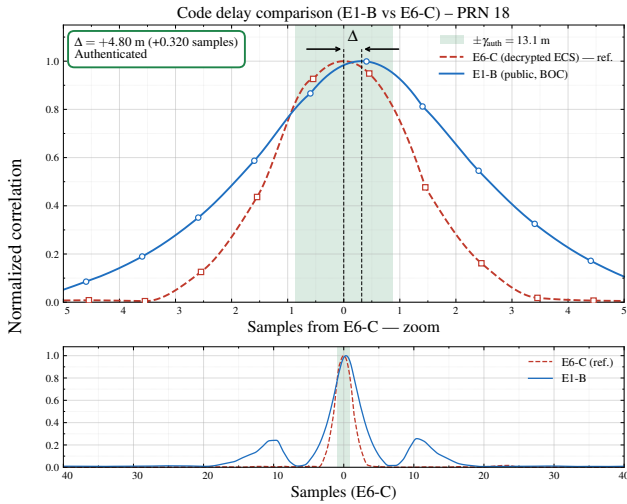


Fig. 8: Correlació E1-B vs. E6-C i comparació dels pics, PRN 14 (de [2]).

Fig. 9: Comparació dels retards de codi E1-B i E6-C per al PRN 18 (campanya actual): $\Delta = +4,80$ m, per sota de $\gamma_{\text{auth}} = 13,1$ m —rang autènticat.

intercalar— i de la correcció de biaixos deterministes; el BGD encara s'obté de fitxers del GSC i el seu parseig està pendent de consolidar. Tercer, la verificació és encara semi-automàtica, sense proves contínues de tota la cadena.

Tot i això, el sistema ja ofereix una cadena SAS completa i reproduïble, en la línia de l'única altra biblioteca oberta de referència —*osnmlib* per a l'OSNMA [12, 13].

6 CONCLUSIONS I LÍNIES FUTURES

Aquest treball ha desenvolupat la capa de gestió, el processament de dades auxiliars i criptogràfic i la interfície d'usuari d'un receptor del servei SAS de Galileo, integrada dins la biblioteca SASlib, i n'ha assolit els objectius majoritàriament. L'arquitectura de tres capes dirigida pels esque-

mes JSON (interfície, motor d'execució i blocs de processament) s'ha mostrat adequada per a la complexitat del sistema, i la refactorització del model de dades compartit n'ha reforçat la robustesa enfront de la concurrència. La interfície gràfica i la integració basada en fitxers amb el mòdul de senyal han permès configurar, executar i validar la cadena de manera reproduïble. La validació amb senyal real ho confirma: la cadena completa autèntica el rang del PRN 18 de la campanya actual ($\Delta = +4,80$ m, sota el llindar) a partir de l'E6-C xifrat, amb resultats coherents amb les plataformes de referència. A més, com a contribució addicional, s'ha desplegat una estació web de monitoratge que executa el servei de manera contínua dins de l'Escola d'Enginyeria.

Com a línies futures es plantegen la validació de l'autenticació a nivell de posició a mesura que més satèl·lits emetin el senyal E6-C xifrat, la integració de proves automàtiques a cada *push* per validar la cadena completa, la integració de la banda E5, l'estandardització de les sortides i la implementació d'algoritmes de detecció de *spoofing* (MPD, VSS, SQM, C/N0, AGC).

ÚS DE LA INTEL·LIGÈNCIA ARTIFICIAL GENERATIVA

En el desenvolupament d'aquest TFG s'han emprat eines d'intel·ligència artificial generativa com a suport. GitHub Copilot s'ha utilitzat per a l'autocompletat de codi en tasques repetitives dels formularis Dash i estructures de configuració JSON). Claude (Anthropic) i ChatGPT (OpenAI) s'han utilitzat per a consultes tècniques puntuals i per a l'estructuració, la redacció i la correcció lingüística dels documents.

No s'han emprat eines d'IA en el disseny ni la implementació dels components criptogràfics, desenvolupats manualment seguint les especificacions oficials [4]. Tot el contingut d'aquesta memòria ha estat revisat, verificat i corregit

per l'estudiant contra les fonts primàries corresponents.

AGRAÏMENTS

L'autor agraeix al tutor d'aquest treball, Rafael Terris-Gallego, l'orientació i el seguiment constants, i al grup de recerca SPCOMNAV de la Universitat Autònoma de Barcelona —dirigit pels professors José A. López-Salcedo i Gonzalo Seco-Granados— l'acollida i els mitjans posats a la seva disposició, també a Enrique Takiguchi-Medina per la plataforma d'enregistrament SDR. Finalment, agraeix a la seva família el suport incondicional al llarg de tot aquest camí.

REFERÈNCIES

- [1] I. Fernandez-Hernandez *et al.*, “Semiassisted Signal Authentication for Galileo: Proof of Concept and Results,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 4, pp. 4393–4404, 2023.
- [2] R. Terris-Gallego, Ò. Sardà-Martí, E. Takiguchi-Medina, A. Galan-Figueras, J. A. López-Salcedo, G. Seco-Granados, and I. Fernandez-Hernandez, “SASlib: An Open Python Library for Galileo Signal Authentication Service,” in *Proc. ESA NAVITEC 2026*, Noordwijk, May 2026, preprint (en revisió).
- [3] European GNSS Service Centre, “Galileo OSNMA Service,” 2025. [En línia]. Disponible: <https://www.gsc-europa.eu/galileo/services/galileo-open-service-navigation-message-authentication-osnma>
- [4] I. Fernandez-Hernandez *et al.*, “Galileo Signal Authentication Service (SAS),” in *Proc. ION GNSS+ 2024*, Baltimore, Sep. 2024, pp. 3292–3307.
- [5] R. Terris-Gallego, J. A. López-Salcedo, G. Seco-Granados, and I. Fernandez-Hernandez, “Operating Modes and Performance Evaluation of Galileo ACAS,” in *Proc. ION GNSS+ 2022*, Denver, Sep. 2022.
- [6] R. Terris-Gallego, J. A. López-Salcedo, G. Seco-Granados, and I. Fernandez-Hernandez, “Preliminary Evaluation of Galileo ACAS using Existing E1-E6 Open Signals,” in *Proc. ION GNSS+ 2023*, Denver, Sep. 2023, pp. 1388–1402.
- [7] R. Terris-Gallego, J. A. López-Salcedo, and G. Seco-Granados, “Testing Galileo SAS with Encrypted Signal-in-Space,” in *Proc. ION GNSS+ 2025*, Baltimore, Sep. 2025.
- [8] R. Terris-Gallego, “Analysis and Implementation of Anti-Spoofing Detection Methods based on Galileo Signal Authentication Service,” Ph.D. dissertation, Universitat Autònoma de Barcelona, Sep. 2024.
- [9] European Union / EUSPA, “Galileo OSNMA SIS ICD, Issue 1.1,” EUSPA, Oct. 2023.
- [10] European Union / EUSPA, “Galileo OSNMA Receiver Guidelines, Issue 1.3,” EUSPA, Jan. 2024.
- [11] K. Borre, D. M. Akos, N. Bertelsen, P. Rinder, and S. H. Jensen, *GNSS Software Receivers*. Cambridge University Press, 2023.
- [12] A. Galan-Figueras, I. Fernandez-Hernandez, L. Cucci, and G. Seco-Granados, “OSNMALib: An Open Python Library for Galileo OSNMA,” in *Proc. NAVITEC*, IEEE, 2022.
- [13] A. Galan-Figueras, C. Iñiguez, I. Fernandez-Hernandez, S. Pollin, and G. Seco-Granados, “Improving OSNMALib: New Formats, Features, and Monitoring Capabilities,” *IEEE J. Indoor Seamless Positioning Navig.*, vol. 3, pp. 117–127, Apr. 2025.
- [14] R. Terris-Gallego, I. Fernandez-Hernandez, J. A. López-Salcedo, and G. Seco-Granados, “Guidelines for Galileo ACAS Implementation,” in *Proc. ICL-GNSS 2022*, Tampere, Jun. 2022.
- [15] R. Terris-Gallego, J. A. López-Salcedo, G. Seco-Granados, and I. Fernandez-Hernandez, “E1-E6 SDR Platform based on bladeRF for Testing Galileo ACAS,” in *Proc. ENC 2023*, Noordwijk, Oct. 2023.
- [16] R. Terris-Gallego, I. Fernandez-Hernandez, J. A. López-Salcedo, and G. Seco-Granados, “Efficient Detection of Galileo ACAS Sequences using E6-B Aiding,” in *Proc. ENC 2024*, Noordwijk, May 2024.

APÈNDIX

A.1 Configuració completa del pipeline (PRN 18)

La configuració JSON següent reproduïx el pipeline complet —tots els blocs— amb què s'ha autenticat el PRN 18 (captura del 23 de juny de 2026, *dataset230626*), descrit a la Secció 5. Se n'han ofuscat les credencials i s'han abreujat les rutes als fitxers.

```
{
  "metadata": { "execution_id": "dataset230626_sat18", "version": "1.0" },
  "module_configs": {
    "ConfigInit": {
      "gst_start_time_sec": 846937156.0,
      "auth_periodicity_sec": 30, "auxiliary_mode": "E1B",
      "svid_to_search": "18", "svid_encrypted": "18",
      "recs_length": 16, "recs_kdi": 0,
      "bgd_file_enabled": false, "tesla_file_enabled": true,
      "rf_input_from_file": true, "recs_file_enabled": true,
      "recs_directory": ".../RECS/parsed_rcs",
      "tesla_file_path": ".../tesla_keys/converted.json",
      "rf_input_path_e1": ".../0756_..._E1_s_5s_opensky.bin",
      "rf_input_path_e6": ".../0755_..._E6_s_5s_opensky.bin",
      "calibration_active": false, "auth_duration_sec": 5.0,
      "scheduled_time_doy": "174 12:19:19"
    },
    "TimeReferenceSynchronizer": {
      "selected_ntp_server": "time.google.com",
      "query_timeout_sec": 5.0, "retry_count": 3
    },
    "AlmanacManagement": {
      "almanac_source": "gsc-europa", "almanac_format": "xml",
      "use_current_time": false,
      "target_datetime": "2026-06-23T12:19:19Z", "save_almanac_file": true
    },
    "UncertaintyModule": {
      "enabled": true,
      "satellite_clock_uncertainty_ms": 0.05,
      "receiver_clock_uncertainty_ms": 0.1,
      "propagation_uncertainty_ms": 500.0
    },
    "SnapshotRecording": {
      "manual_mode": true,
      "manual_snapshot_path": ".../0755_..._E6_s_5s_opensky.bin",
      "api_base_url": "https://<host>:5173",
      "api_username": "<user>", "api_password": "<obfuscat>",
      "frequency_e6": 1278750000, "frequency_e1": 1575420000,
      "sample_rate": 20000000, "bandwidth": 10000000, "gain": 60,
      "duration_seconds": 5, "record_now": false,
      "poll_interval_seconds": 5, "download_timeout_minutes": 10,
      "set_id": "a1", "info_tag": "dataset230626"
    },
    "BGDandRECSdownloader": {
      "recs_provider": "GSCX",
      "recs_interface_version": "01", "recs_file_version": "01",
      "recs_tstart": "174 12:19:00", "recs_duration_sec": 60,
      "recs_rtba_deciseconds": 10, "recs_svids": "18",
      "recs_kdi": 0, "recs_rand": 0, "recs_nchip": 4,
      "ssh_enabled": false, "ssh_host": "<host>", "ssh_port": 22,
      "ssh_username": "<user>", "ssh_remote_host": "<host>",
      "ssh_remote_port": 50567, "ssh_local_port": 50567,
      "online_server_url": "https://<host>:50567"
    },
    "BGDandRECSparser": {
      "recs_files": [".../01_261741218000_..._18_0_0_4_01.bin",
        ".../01_261741219000_..._18_0_0_4_01.bin"],
      "bgd_files": [], "recs_directory": ".../RECS",
      "output_directory": ".../RECS/parsed_rcs_pipeline"
    },
    "TESLAkeyManagement": {
      "manual_mode": true,
      "osnma_url": "https://osnmalib.eu/septentrio/subframe_json",
      "filter_enabled": false, "export_format": "json",
      "multi_fetch_enabled": false,
      "multi_fetch_duration_s": 400, "multi_fetch_interval_s": 28
    },
    "RECSDecryption": {
      "prefer_manual_key": false, "prn_encrypted_list": "18",

```

```

    "recs_kdi": 0, "target_epoch_override": 846937159,
    "tesla_json_path": ".../tesla_keys/converted.json",
    "recs_directory": ".../RECS/parsed_rcs",
    "window_before_sec": 60.0, "window_after_sec": 60.0,
    "use_next_key_if_needed": true, "include_padding": false,
    "save_ecs": true, "output_basename": "ecs_dataset230626_sat18"
  },
  "SignalCorrelation": {
    "sample_rate_hz": 20000000.0, "start_sample": 0,
    "num_samples": 100000000,
    "ni_periods_e6": 16, "nc_coherent_e6": 1,
    "ni_periods_aux": 20, "nc_coherent_aux": 1,
    "doppler_min_hz": 4400.0, "doppler_max_hz": 6000.0,
    "doppler_step_hz": 50.0,
    "correct_code_doppler": true, "correct_range_rate": true,
    "fine_doppler_search": true, "fine_doppler_range_hz": 200.0,
    "pdi_method": "NPD1", "exp_n_coh": 2.0,
    "interpolation_mode": "quadratic-polynomial-3-points",
    "ppsp_exclusion_chips": 50
  },
  "EphemeridesManagement": {
    "ephemerides_source": "esa_gssc", "ephemerides_type": "broadcast",
    "precise_product_type": "rapid", "use_current_time": false,
    "target_datetime": "2026-06-23T12:19:19Z",
    "prn_filter": "18", "save_ephemerides_file": true
  },
  "SatelliteAuthenticator": {
    "kappa": 3, "auto_calibrate_offset": false,
    "receiver_offset_samples": 0,
    "bgd_e1_e6_m": 0, "iono_e1_delay_m": 0,
    "min_snr_db": 10, "min_ppsp_db": 5,
    "sigma_hwbrx_m": 0, "sigma_bgd_m": 0, "sigma_iono_e1_m": 3.5,
    "sigma_mp_e1_m": 3, "sigma_mp_e6_m": 0.6,
    "sigma_noise_e1_m": 0.4, "sigma_noise_e6_m": 2.5,
    "generate_plots": true, "save_results": true
  },
  "PVTComputation": {
    "pvt_algorithm": "least_squares", "min_satellites": 4,
    "use_authenticated_only": false, "min_elevation_deg": 5.0,
    "weight_by_cn0": false, "weight_by_elevation": false,
    "min_ppsp_db": 3.0, "min_cn0_dbhz": 38.0,
    "reference_latitude": 41.5007, "reference_longitude": 2.1136,
    "reference_altitude": 146.0
  }
}

```

A.2 Finestra de configuració de la interfície

Detall de la finestra modal de configuració d'un bloc a la interfície gràfica, amb el formulari generat automàticament a partir de l'esquema JSON del bloc.

Configuration Initialization

Initial configuration parameters for the SAS pipeline - Central configuration for all blocks

User Parameters

Main user-configurable parameters for authentication

Starting Time to Authenticate (GST sec)

813755236

GST start time in seconds (0 = auto-detect from RECS)

Authentication Duration (sec)

30

Total duration of the authentication period in seconds

Authentication Periodicity (sec)

30

Time interval between RECS files in seconds

Auxiliary Signal Band

E1-B

Select auxiliary signal band (E1-B, E1-C, E6-B)

Test Parameters

Parameters for testing

SVID to Search

14

List of all satellite vehicle IDs to search for correlation (valid: 1-36)

SVID to Search Encrypted

14

Subset of SVIDs that are encrypted and require RECS decryption (valid: 1-36)

RECS Length

16

Length of RECS data in milliseconds (1 to 16 ms)

KDI (Key Delay Indicator)

0 - Current Key

Key Delay Indicator: 0=use current key, 1=use next key

Snapshot File

☒

Enable to use IQ file, disable for live RF input (recorder)

TESLA Key File

☒

Enable to use TESLA keys from local file, disable to download from internet

RECS Local File

☒

Enable to use local RECS files, disable to download from server

BGD Local File

☐

Enable to use local BGD files, disable to download from server

Save

Run

Edit JSON

Close

Fig. 10: Finestra de configuració d'un bloc, *ConfigInit* (de [2]).

A.3 Estació de monitoratge

Captures de l'estació de monitoratge desplegada a l'Escola d'Enginyeria, que executa el *pipeline* de SASlib de manera periòdica (Secció 5).

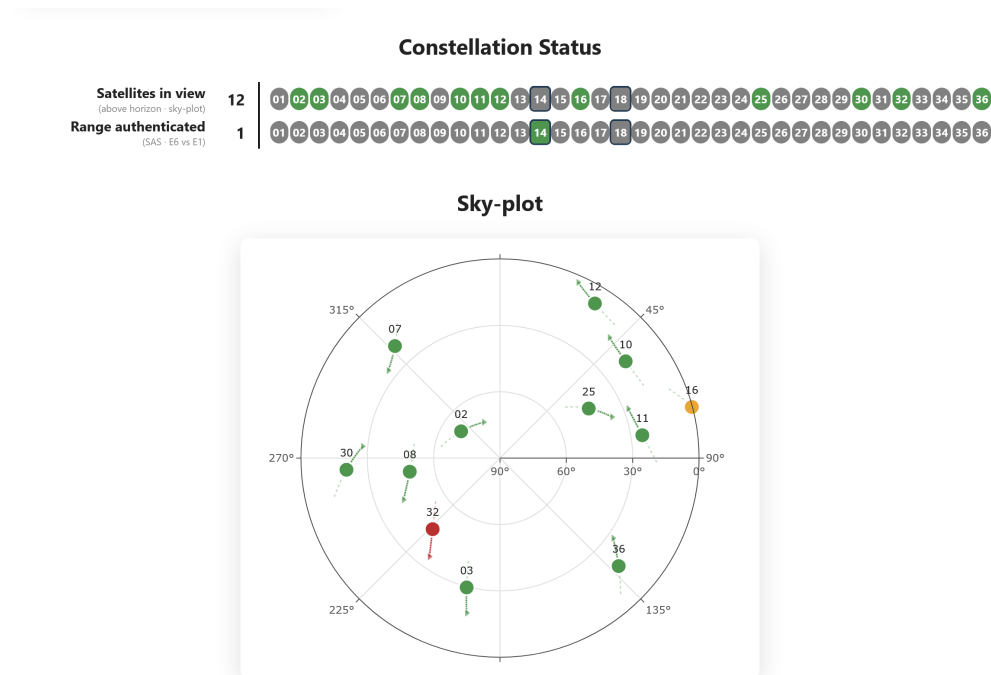


Fig. 11: Estació de monitoratge: estat de la constel·lació i *skyplot* dels satèl·lits visibles.

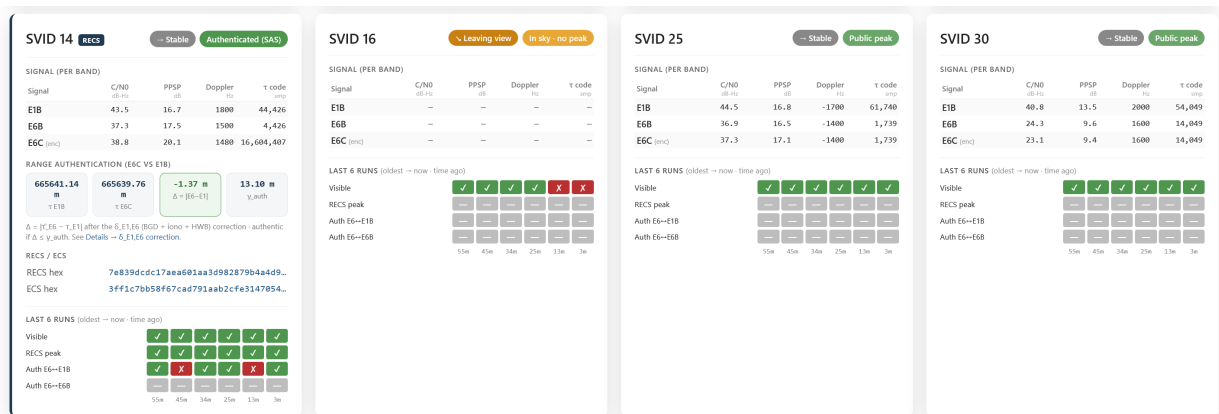


Fig. 12: Estació de monitoratge: targetes per satèl·lit amb les observables i el veredict de d'autenticació (el SVID 14 hi consta com a autènticat pel SAS).