# Estratègies Avançades d'Optimització i Resolució de Conflictes en la Interfície Northbound de Huawei FusionSolar: Anàlisi Tècnica de l'Error 407 i Arquitectures de Persistència de Dades per a Sistemes de Monitoratge Fotovoltaic

La integració de sistemes de gestió d'energia basats en el núvol amb infraestructures de generació distribuïda ha esdevingut un pilar fonamental de la transició energètica contemporània. En aquest context, l'ecosistema FusionSolar de Huawei, gestionat a través del sistema SmartPVMS (Smart Photovoltaic Management System), ofereix una de les plataformes més robustes per al monitoratge d'actius solars a escala global. Tanmateix, l'accés a les dades mitjançant la seva interfície de dades cap al nord, coneguda com a Northbound API, presenta reptes tècnics significatius, especialment pel que fa a la gestió del trànsit i el compliment de les restriccions de flux imposades pel fabricant. L'aparició recurrent del codi d'error 407, definit formalment com a "Access Frequency Too High", constitueix el principal obstacle per als desenvolupadors que intenten implementar solucions de visualització de dades històriques d'alta granularitat, com les consultes horàries.1

L'arquitectura del SmartPVMS està dissenyada per prioritzar la disponibilitat del servei i la integritat del sistema NetEco, el motor que sustenta la plataforma FusionSolar. Aquesta priorització es manifesta en una política de control de flux extremadament conservadora que, tot i protegir els servidors de Huawei contra atacs de denegació de servei o sobrecàrregues innecessàries, limita severament la flexibilitat de les aplicacions de tercers.1 Per resoldre la problemàtica de les fallades immediates en la descàrrega de dades horàries, cal realitzar una dissecció profunda tant de la mecànica de l'API com de la implementació específica del codi client, identificant els punts de fricció on la lògica de l'aplicació entra en conflicte amb els algorismes de limitació de velocitat del servidor.

## Anatomia Tècnica de la Restricció de Flux i l'Error 407

L'error 407 no s'ha d'interpretar com una fallada de l'endpoint o del servidor, sinó com una resposta regulatòria activa de la infraestructura de Huawei. El sistema SmartPVMS utilitza un enfocament de limitació per nivells que afecta tant la concurrència immediata com la quota total diària.1 Aquesta distinció és crítica: molts desenvolupadors assumeixen que si no han superat la quota diària, les seves peticions haurien de ser acceptades, ignorant que les restriccions de concurrència per minut són molt més estrictes i s'apliquen de manera implacable.2

### La Barrera de la Concurrència per Minut

La documentació oficial per a la interfície de dades horàries de plantes (getKpiStationHour) especifica una restricció que sovint es passa per alt: només es permet una petició concurrent per minut.1 Aquesta limitació és la causa directa de l'error 407 quan l'usuari intenta descarregar dades per a diversos dies de manera seqüencial en un bucle ràpid. Si l'aplicació executa una crida per al "Dia 1" i, immediatament després de rebre la resposta, executa la crida per al "Dia 2", el servidor detectarà que no ha transcorregut l'interval de seguretat necessari (sovint interpretat com a 60 segons de rellotge de servidor) i bloquejarà la segona petició instantàniament.2

Aquest comportament explica per què l'error apareix "de seguida". No es tracta d'un exhauriment de la capacitat de dades, sinó d'una violació del protocol temporal de la interfície. El sistema de Huawei implementa un algorisme de cubell de fitxes (token bucket) on el fitxer per a consultes històriques es regenera amb una cadència extremadament lenta en comparació amb altres interfícies més permissives, com la de dades en temps real.1

### Càlcul de Quotes Diàries i Capacitat de Càrrega

Més enllà de la barrera del minut, existeix una quota diària total que es calcula en funció del nombre de plantes gestionades pel compte de l'usuari Northbound. Aquesta mètrica busca escalar el servei de manera proporcional a la mida de la instal·lació, però sovint resulta insuficient per a operacions de recuperació de dades històriques massives.

| **Interfície** | **Limitació de Concurrència** | **Fórmula de Quota Diària** |
| --- | --- | --- |
| Real-Time Plant Data | 1 petició / minut | $\lceil \text{Nombre de plantes}/100 \rceil$ cada 5 minuts |
| Hourly Plant Data | 1 petició / minut | $\lceil \text{Nombre de plantes}/100 \rceil + 24$ |
| Daily Plant Data | 1 petició / minut | $\lceil \text{Nombre de plantes}/100 \rceil + 24$ |
| Monthly Plant Data | 1 petició / minut | $\lceil \text{Nombre de plantes}/100 \rceil + 24$ |
| Yearly Plant Data | 1 petició / minut | $\lceil \text{Nombre de plantes}/100 \rceil + 24$ |

1

Segons aquestes fórmules, un usuari que gestioni entre 1 i 100 plantes tindrà una quota diària per a dades horàries de només 25 crides.1 Atès que la interfície horària només retorna les dades d'un únic dia natural calculat a partir del paràmetre collectTime, descarregar un mes complet d'històrics (30 o 31 dies) esdevé matemàticament impossible de realitzar en un sol cicle de 24 hores sota un sol compte, ja que la demanda (30 crides) supera la quota disponible (25 crides).1 Aquesta restricció obliga a la implementació d'estratègies de cache i sincronització progressiva.

## Anàlisi de Patologies en el Codi de l'Aplicació Client

L'anàlisi del codi present a huawei\_client.py i la seva interacció amb el framework Streamlit revela diverses vulnerabilitats estructurals que exacerben el problema de l'error 407. Tot i que el client implementa una classe estructurada amb gestió de sessions mitjançant la llibreria requests, la lògica de control d'errors i el cicle de vida de la connexió no estan alineats amb les necessitats del SmartPVMS.1

### Insuficiència de la Lògica de Reintent (Backoff)

El client utilitza un bucle de reintent per als mètodes get\_kpi\_station\_hour i get\_kpi\_station\_day que s'activa quan es detecta el failCode 407. El temps d'espera es calcula com wait = 5 \* (attempt + 1), el que resulta en esperes de 5, 10 i 15 segons respectivament.1 Atès que la limitació real del servidor de Huawei per a peticions concurrents és d'un minut, aquests intervals són críticament curts.2 L'aplicació torna a intentar la petició quan el servidor encara manté el bloqueig actiu, provocant que s'esgotin ràpidament els intents de reintent (max\_retries = 3) sense haver superat mai la finestra temporal de restricció.1

Aquesta ineficiència es tradueix en una fallada total del mètode després de només 30 segons aproximats d'espera acumulada, quan el que realment necessitava el servidor era un silenci de xarxa de com a mínim 61 segons. A més, el fet que la funció retorni None en cas de fallada final sol provocar errors de tipus TypeError o AttributeError en la lògica de Streamlit que intenta processar aquestes dades, ja que l'aplicació no està preparada per gestionar l'absència de resposta de manera elegant.3

### Inconsistència en la Gestió del failCode 407

S'ha detectat una asimetria en la implementació de la seguretat de flux dins del client. Mentre que les consultes de KPIs horaris i diaris tenen una certa lògica de reintent, els mètodes fonamentals com get\_station\_list i get\_station\_real\_kpi manquen totalment d'aquesta protecció.1 Aquest és un punt cec crític: si la petició inicial per obtenir la llista de plantes falla a causa d'una ràfega de trànsit prèvia, tota l'aplicació queda inoperativa immediatament perquè no disposa dels stationCodes necessaris per a les crides posteriors.1 Una gestió robusta de l'API Northbound requereix una aplicació universal de la lògica de control de flux a tots els mètodes que interactuen amb el núvol.

### El Perill dels Logins Recursius i el Bloqueig de Sessió

El mètode get\_station\_list conté una crida recursiva al detectar els codis d'error 305 (Sessió no iniciada) o 401 (Permisos insuficients/Token invàlid).1 La lògica intenta fer un self.login() i després tornar a cridar-se a si mateixa. Si per qualsevol motiu el login té èxit però el servidor triga uns segons a propagar l'estat de la sessió a través de la seva infraestructura distribuïda, la recursió podria disparar una seqüència de peticions ràpides que el sistema de protecció de Huawei interpretaria com un atac, disparant l'error 407 no per freqüència de dades, sinó per freqüència de peticions d'autenticació.1

Cal recordar que el login també té les seves pròpies restriccions de trànsit: si un usuari introdueix la contrasenya incorrecta 5 vegades en 10 minuts, el compte es bloqueja durant 30 minuts. Així mateix, el nombre de logins reeixits està limitat a 5 vegades cada 10 minuts per usuari Northbound.1 Una aplicació que no persisteixi correctament el XSRF-TOKEN a la memòria de sessió i forci un login en cada refresc de pàgina (com passa sovint en Streamlit) esgotarà aquesta quota en qüestió de minuts.1

## Solucions Estratègiques per a la Descàrrega d'Històrics

Per superar les limitacions imposades pel fabricant sense comprometre l'estabilitat de l'aplicació, s'han de considerar solucions que modifiquin el paradigma de consulta: passar d'un model de "dades sota demanda" a un model de "sincronització intel·ligent".

### Arquitectura de Descàrrega amb Retard Semàntic

La primera intervenció creativa consisteix a implementar un "throttler" semàntic que garanteixi el compliment estricte de la finestra d'un minut. En lloc de confiar en el reintent després de l'error, l'aplicació ha de controlar proactivament el temps transcorregut entre peticions d'històrics.

Python

import time  
  
class RobustHuaweiClient(HuaweiClient):  
 def \_\_init\_\_(self, \*args, \*\*kwargs):  
 super().\_\_init\_\_(\*args, \*\*kwargs)  
 self.last\_history\_request = 0  
  
 def \_wait\_for\_slot(self):  
 elapsed = time.time() - self.last\_history\_request  
 if elapsed < 65: # Marge de seguretat sobre els 60s  
 time.sleep(65 - elapsed)  
 self.last\_history\_request = time.time()

L'aplicació ha de ser conscient que descarregar dades horàries per a un mes sencer requerirà, com a mínim, 30 minuts d'execució contínua. En el context de Streamlit, aquest tipus d'operacions no s'han de realitzar mai dins del fil principal de renderització de la interfície d'usuari, ja que provocarien que la pàgina semblés bloquejada o que el navegador tanqués la connexió per timeout.6

### Implementació de Persistència Local (SQLite / Parquet)

Atès que les dades històriques horàries de dies passats són immutables (una vegada generades, no canvien), és ineficient i arriscat consultar-les a l'API cada vegada que un usuari visualitza un gràfic. La solució professional és crear una capa de persistència local.7

1. **Captura en Temps Real**: Utilitzar l'endpoint getStationRealKpi cada 5 o 15 minuts. Aquesta interfície permet múltiples crides i retorna el rendiment del dia actual (day\_power), els ingressos (day\_income) i l'estat de salut de la planta.1
2. **Emmagatzematge**: Guardar aquestes captures en una base de dades local (SQLite és ideal per a la seva simplicitat en Python).
3. **Sincronització Nocturna**: Configurar una tasca programada (background cron job) que s'executi durant la nit per cridar a getKpiStationHour per al dia anterior. Aquesta crida única diària servirà per consolidar les dades i assegurar que l'històric local és idèntic al de la plataforma oficial de Huawei.1
4. **Visualització**: L'aplicació Streamlit llegirà sempre de la base de dades local. Si l'usuari demana un rang de dates que no està a la base de dades, l'aplicació pot mostrar un missatge de "Dades en procés de descàrrega" en lloc d'intentar una crida en temps real que podria desencadenar l'error 407.9

### Gestió de la Diversitat de Dispositius i Tipus de Dades

El SmartPVMS classifica els dispositius mitjançant IDs numèrics que són essencials per a les consultes de dades detallades a nivell de dispositiu (getDevRealKpi o getDevHistoryKpi).1 Quan es realitza una anàlisi horària, sovint és necessari creuar les dades de la planta amb les de l'inversor o el comptador per obtenir una visió completa del flux d'energia.

| **ID de Tipus de Dispositiu** | **Nom del Dispositiu** | **Mètriques Clau Disponibles** |
| --- | --- | --- |
| 1 | Inversor de cadena (String Inverter) | Voltatges PV, corrents PV, temperatura interna |
| 10 | Instrument de monitoratge ambiental (EMI) | Irradiació, temperatura del panell, velocitat vent |
| 17 | Comptador de xarxa (Grid Meter) | Potència activa, factor de potència, energia inversa |
| 38 | Inversor residencial | Rendiment diari, potència mppt |
| 39 | Bateria residencial | Estat de càrrega (SOC), potència de càrrega/descàrrega |
| 47 | Sensor de potència (Power Sensor) | Consum de la xarxa, potència de l'habitatge |

1

L'error 407 també pot aparèixer si s'intenten consultar massa dispositius simultàniament. Huawei limita les crides a un màxim de 100 dispositius del mateix tipus per petició.1 Per a sistemes amb múltiples inversors o bateries, el codi ha d'agrupar els ESNs (Electronic Serial Numbers) de manera intel·ligent abans d'enviar la petició POST.

## Solucions "Off-Road": Kiosk Mode i Multi-Comptes

Quan les restriccions oficials de l'API Northbound resulten completament insuperables per al cas d'ús requerit, existeixen alternatives creatives que aprofiten altres funcionalitats de la plataforma FusionSolar.

### La Tècnica del Scraper de Mode Kiosk

El mode Kiosk és una funcionalitat dissenyada per mostrar dades de la planta en pantalles públiques sense requerir login. En activar-lo al portal web de FusionSolar, es genera una URL amb un paràmetre de hash (kk=...).11

Aquesta interfície, tot i que està pensada per a navegadors humans, consumeix dades d'un endpoint JSON que no està subjecte a les mateixes quotes dràstiques que l'API Northbound.11 Mitjançant tècniques de web scraping o inspecció de peticions de xarxa, és possible extreure la corba de potència del dia actual i el total d'energia generada.11 Aquesta via és especialment útil per a aplicacions de visualització ràpida on el retard de 30 minuts en l'actualització de les dades del Kiosk és acceptable.12

| **Avantatges Mode Kiosk** | **Desavantatges Mode Kiosk** |
| --- | --- |
| No requereix autenticació Northbound | Dades limitades (no hi ha dades de dispositius) |
| Lliure d'errors 407 per freqüència | Actualització cada 30 minuts (no és temps real) |
| Sencill d'implementar amb requests.get() | La URL caduca cada any i cal renovar-la |

11

### Estratègia Multi-Usuari Northbound

Huawei permet als administradors d'empresa crear fins a 10 comptes Northbound independents per cada compte d'instal·lador.14 Com que les limitacions de flux s'apliquen per usuari de l'API i no per planta (en el cas de la concurrència per minut), una aplicació pot distribuir les peticions entre diversos comptes.1

Si es disposa de 5 comptes Northbound, l'aplicació podria teòricament realitzar 5 peticions de dades horàries en un sol minut (una amb cada compte) sense disparar el failCode 407. Aquesta solució requereix una gestió complexa de credencials i estats de sessió, però és la via oficial més efectiva per escalar el volum de dades descarregades en paral·lel.5

## Optimització del Cicle de Vida a Streamlit

Streamlit és un framework excel·lent per a la ciència de dades, però el seu model d'execució és intrínsecament perillós per a les APIs amb limitació de velocitat. Cada vegada que un usuari canvia un selector de data, prem un botó o fins i tot quan la connexió WebSocket es refresca, Streamlit torna a executar el script de dalt a baix.6 Sense precaució, aquest comportament pot generar dotzenes de peticions de login i dades en pocs segons, provocant el bloqueig immediat del compte per error 407.

### Ús de st.cache\_data amb Paràmetres Temporals

Per evitar el "bombardeig" accidental de l'API de Huawei, totes les funcions que interactuen amb el client han d'estar decorades amb @st.cache\_data.9 És crucial definir un ttl (Time-To-Live) que s'ajusti a la naturalesa de la dada.

Python

@st.cache\_data(ttl=600) # Cache durant 10 minuts  
def get\_cached\_realtime\_data(station\_code):  
 client = get\_client\_from\_session()  
 return client.get\_station\_real\_kpi(station\_code)

Aquesta configuració garanteix que, encara que l'usuari refresqui la pàgina 100 vegades, Streamlit només cridarà a l'API de Huawei una vegada cada 10 minuts, protegint així la quota diària i la concurrència.9 Per a dades històriques horàries de dies ja tancats, el cache pot ser molt més llarg (24 hores o més), ja que les dades de l'inversor per a una data passada no canviaran.10

### Preservació de la Sessió amb st.session\_state

L'objecte HuaweiClient i el seu XSRF-TOKEN s'han de guardar a st.session\_state per evitar processos de login innecessaris.6 El mètode de login de Huawei és especialment sensible i el bloqueig per excessives peticions d'autenticació és un dels motius més comuns darrere de l'error 407 "immediat".1

Python

if 'huawei\_client' not in st.session\_state:  
 client = HuaweiClient(username, password)  
 client.login()  
 st.session\_state.huawei\_client = client

Aquest patró assegura que la sessió HTTP creada per la llibreria requests (amb les seves galetes i capçaleres persistents) es mantingui viva durant tota la navegació de l'usuari, reduint el trànsit de control a la xarxa.1

## Diagnòstic de Fallades en el Codi: On falla realment?

L'anàlisi del codi adjunt permet identificar tres punts crítics on la fallada és imminent sota condicions de càrrega:

1. **La Lògica de Bucle d'Històrics**: Si l'usuari selecciona un rang de dates a la interfície i el codi executa un for date in date\_range: client.get\_kpi\_station\_hour(...), la fallada es produirà a la segona iteració del bucle, exactament als pocs mil·lisegons d'iniciar-se, perquè el servidor encara no ha alliberat el slot de concurrència d'un minut.1
2. **Manca de Gestió del Token Expira**: Tot i que s'intenta gestionar el failCode 305 en alguns mètodes, si el token expira durant una descàrrega llarga de dades horàries, la crida recursiva podria no funcionar correctament si no s'ha netejat prèviament la sessió de requests, provocant que s'enviïn tokens antics i nous en una successió ràpida que dispara el límit de freqüència.1
3. **Temps d'Espera de Timeout**: El client utilitza un timeout estàndard de 10 segons.1 Atès que els servidors de Huawei de vegades triguen més a processar consultes horàries de plantes molt grans, una petició que faci timeout i es torni a intentar immediatament serà bloquejada amb un 407 pel servidor, que encara pot estar processant la petició anterior internament. Es recomana augmentar el timeout a 30 segons per a consultes històriques.1

## Models Matemàtics d'Optimització del Temps de Reintent

Per maximitzar l'èxit de les descàrregues sense caure en el bloqueig, es pot aplicar un model de reintent basat en fitxes de temps real. L'algorisme de retrocés ha de ser conscient del "temps de silenci" obligatori.

Si definim $T\_{limit} = 60$ segons com la restricció de Huawei, i $T\_{request}$ com el moment en què s'inicia una petició, la condició per a la següent petició amb èxit $T\_{next}$ és:

$$T\_{next} > T\_{request} + T\_{limit} + \delta$$

On $\delta$ és un marge de seguretat per compensar la latència de xarxa i la desincronització dels rellotges del servidor. Les proves empíriques suggereixen que un $\delta$ de 5 segons és òptim per a l'entorn europeu (EU5).2

En cas de rebre un 407 malgrat respectar el temps, l'aplicació ha d'assumir que el servidor ha aplicat un bloqueig temporal més llarg (sovint 5 o 10 minuts si es detecten abusos repetits) i ha d'aturar tota activitat durant aquest interval abans de reintentar el login.1

## Conclusions i Full de Ruta de Resolució

La resolució dels problemes amb l'API de Huawei requereix un canvi fonamental en la manera com l'aplicació Streamlit gestiona el flux de dades. No és possible utilitzar l'API de Huawei com si fos una base de dades local de resposta immediata.

* **Acció Immediata**: Cal modificar el bucle de descàrrega de dades horàries per incloure un time.sleep(65) obligatori entre cada crida, fins i tot si no hi ha hagut error. Això solucionarà la fallada "enseguida" que esmenta l'usuari.1
* **Millora del Client**: S'han d'actualitzar els mètodes get\_station\_list i get\_station\_real\_kpi per incloure la mateixa lògica de detecció del 407 que tenen els mètodes horaris, garantint que l'aplicació no mori davant d'una punta de trànsit inicial.1
* **Estratègia de Persistència**: S'ha d'implementar un mecanisme de memòria intermèdia (buffer) que descarregui les dades horàries una sola vegada i les guardi a disc. L'aplicació Streamlit ha de consultar primer aquest històric local. Només si les dades no hi són i ha passat prou temps des de l'última petició, s'ha d'autoritzar una nova crida a l'API.7
* **Alternativa d'Emergència**: Per a visualitzacions ràpides de la corba de producció, es recomana explorar la integració del mode Kiosk, que és molt més resilient a les consultes freqüents i permet una experiència d'usuari més fluida en l'aplicació Streamlit.11

L'èxit en l'automatització del monitoratge fotovoltaic amb Huawei depèn de la paciència algorítmica. El sistema SmartPVMS no és una API pensada per a l'alta velocitat, sinó per a la sincronització industrial periòdica. Respectar aquesta filosofia de disseny és l'única via per eliminar l'error 407 i garantir la disponibilitat de les dades per a l'anàlisi energètica.1

#### Treballs citats

1. api.pdf
2. Imaster NetEco V600R023C00 Northbound Interface Reference-V6 (SmartPVMS) - Scribd, s'hi ha accedit el dia de febrer 5, 2026, <https://www.scribd.com/document/719784540/iMaster-NetEco-V600R023C00-Northbound-Interface-Reference-V6-SmartPVMS>
3. ACCESS\_FREQUENCY\_IS\_TO, s'hi ha accedit el dia de febrer 5, 2026, <https://github.com/tijsverkoyen/HomeAssistant-FusionSolar/issues/89>
4. Imaster Neteco v600r022c00 Northbound Interface Reference v6 | PDF | Json - Scribd, s'hi ha accedit el dia de febrer 5, 2026, <https://www.scribd.com/document/894990804/Imaster-Neteco-v600r022c00-Northbound-Interface-Reference-v6>
5. Set up API access – Huawei FusionSolar API - Solytic, s'hi ha accedit el dia de febrer 5, 2026, <https://solytic.com/knowledge/set-up-api-access-huawei-fusionsolar-api/>
6. Advanced concepts of Streamlit, s'hi ha accedit el dia de febrer 5, 2026, <https://docs.streamlit.io/get-started/fundamentals/advanced-concepts>
7. how-to-rate-limit-streamlit, s'hi ha accedit el dia de febrer 5, 2026, <https://blog.schogini.com/html_files/how-to-rate-limit-streamlit.html>
8. Session State for Streamlit - Towards Data Science, s'hi ha accedit el dia de febrer 5, 2026, <https://towardsdatascience.com/session-state-for-streamlit-dc9ea1685ea/>
9. Caching overview - Streamlit Docs, s'hi ha accedit el dia de febrer 5, 2026, <https://docs.streamlit.io/develop/concepts/architecture/caching>
10. st.cache\_data - Streamlit Docs, s'hi ha accedit el dia de febrer 5, 2026, <https://docs.streamlit.io/develop/api-reference/caching-and-state/st.cache_data>
11. Way to get (limited) data without developer account · BlazejosP huawei-sun2000-API-CLI, s'hi ha accedit el dia de febrer 5, 2026, <https://github.com/BlazejosP/huawei-sun2000-API-CLI/discussions/16>
12. Home-Assistant-FusionSolar-Kiosk/README.md at master - GitHub, s'hi ha accedit el dia de febrer 5, 2026, <https://github.com/tijsverkoyen/Home-Assisant-FusionSolar-Kiosk/blob/master/README.md>
13. Integrate FusionSolar into your Home Assistant. - GitHub, s'hi ha accedit el dia de febrer 5, 2026, <https://github.com/tijsverkoyen/HomeAssistant-FusionSolar>
14. Northbound user limit to 10 on FusionSolar - HUAWEI Developer Forum, s'hi ha accedit el dia de febrer 5, 2026, <https://forums.developer.huawei.com/forumPortal/en/topic/0203124393941647104>
15. st.cache\_data - Streamlit Docs, s'hi ha accedit el dia de febrer 5, 2026, <https://docs.streamlit.io/1.30.0/develop/api-reference/caching-and-state/st.cache_data>
16. Streamlit Session State: Unlocking the Power of Persistent User Data | by why amit - Medium, s'hi ha accedit el dia de febrer 5, 2026, <https://medium.com/@whyamit101/streamlit-session-state-unlocking-the-power-of-persistent-user-data-41d85758bb02>
17. Session State - Streamlit Docs, s'hi ha accedit el dia de febrer 5, 2026, <https://docs.streamlit.io/develop/api-reference/caching-and-state/st.session_state>
18. Best practice for secure, per-user authentication/session management in multi-page Streamlit apps (with query params)?, s'hi ha accedit el dia de febrer 5, 2026, <https://discuss.streamlit.io/t/best-practice-for-secure-per-user-authentication-session-management-in-multi-page-streamlit-apps-with-query-params/117252>