igc-net: Pilot-Owned Flight Identity and Privacy-Controlled Sharing for the Soaring Community

Norwegian University of Science and Technology
Distributed Systems Engineering Lab
Norway *

Version 0.3 May 2026

Abstract

Soaring pilots upload their flights to a growing number of portals and applications. Each portal assigns its own identity to the same file, controls access on its own terms, and provides no way for the pilot to carry their history, privacy preferences, or social connections to another service. *igc-net* [1] is an open protocol that changes this. Every pilot who uses an igc-net-enabled portal receives a portable, cryptographic identity that belongs to them, not to the portal. Their flight history is anchored to this identity across every participating service. They choose which portals may see their private flights, and they can revoke that access at any time. They can share non-public flights with specific pilots through groups, without changing any flight's publication mode.

Version 0.3 of the protocol formalises these capabilities: a two-credential pilot identity model, three publication modes enforced at the protocol layer, private-access key grants and revocation, private and public groups, and a social follow mechanism. The network remains decentralised. No central server holds the authoritative copy of a pilot's identity or history. For portals, igc-net is not a competing service; it is the open exchange layer beneath existing portals, giving them cross-portal flight discovery, access-control enforcement, and a shared social graph without requiring bilateral integrations.

This paper is written for pilots, portal teams, club administrators, and anyone interested in the future of soaring flight data. Readers who want the wire-format specification, conformance requirements, and implementation guidance should consult the normative protocol documents [1] and the companion technical report [2].

Contents

*Contact: mariusz.nowostawski@ntnu.no. This paper describes protocol version 0.3. For the original protocol design and wire-format detail, see the companion technical report v0.1 [2].

1 Introduction

The International Gliding Commission (IGC) file format [5], standardised by the Fédération Aéronautique Internationale, is the universal record of a soaring flight. Every modern flight instrument produces IGC files. A best-effort public census completed on 9 April 2026 identified at least 16 active public portals and services handling IGC upload, viewing, scoring, or storage. XContest alone logs approaching 500,000 flights per season [7]; DHV-XC, OLC [8], Leonardo, SeeYou Cloud, and others add further volume.

Despite this universality at the file-format level, the ecosystem for *managing* IGC files is deeply fragmented. A pilot who completes a cross-country flight typically uploads the same IGC file to multiple portals independently. Each portal assigns its own identifier to the file, stores it in its own database, and applies its own access rules. If the pilot later wants to keep a flight private from some portals but share it with others, there is no mechanism to do so. If a portal closes, the pilot's history hosted on that portal is lost.

The problem igc-net was originally designed to address was interoperability: giving every IGC file a universal, content-addressed identity so that portals and archives could discover and exchange flights without bilateral agreements. The technical report v0.1 [2] describes that design in detail, including the project's evolution from an IPFS prototype in Go (2023) through an evaluation of alternative peer-to-peer stacks (2024) to the adoption of the Iroh [6] networking stack and a Rust [3] reference implementation.

Version 0.3 extends the protocol significantly. The new capabilities are not primarily about interoperability between portals. They are about giving the *pilot* control: control over their identity, control over who sees their flights, and control over how they share with the people they choose. This paper describes those capabilities and why they matter.

2 igc-net in Three Ideas

Before describing the features of v0.3 in detail, it helps to understand the three ideas on which igc-net is built.

Content addressing. Every IGC file is identified by the BLAKE3 cryptographic hash of its exact bytes. Two pilots independently hashing the same file will always compute the same identifier. This means a flight has one universal identity across every portal, archive, and tool on the network. The technical report [2] describes the hashing and blob model in detail.

Pilot-owned identity. When a pilot registers on an igc-net-enabled portal, they receive a cryptographic identity that belongs to them, not to the portal. The portal may store it on the pilot's behalf, but the identity is portable: a pilot can take it to another portal, and all their flights and access preferences travel with it. If their home portal closes, their identity survives.

Decentralised network. igc-net uses a gossip overlay so that announcements propagate between nodes without a central server. Any node can join the network, announce flights, and receive announcements from others. No central authority approves nodes or controls access to the network. This means no single service controls who participates, and no single outage takes the network down.

These three ideas work together. Content addressing gives flights a universal identity. Pilot-owned identity gives pilots portable ownership and access control over those flights. The decentralised network ensures that no portal, company, or institution is the single point of control for either.

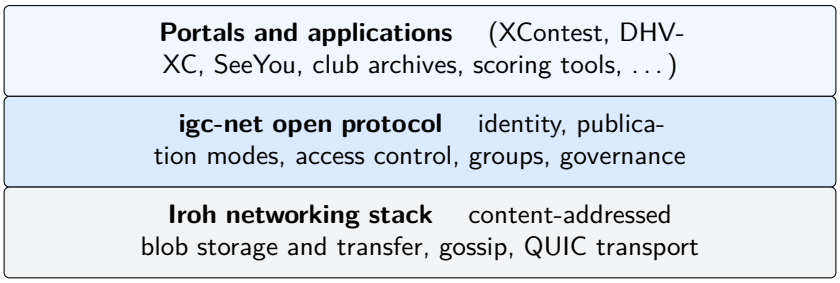

Figure 1: igc-net sits between the transport layer and existing portals. It is not a portal. It is the open protocol layer that portals integrate.

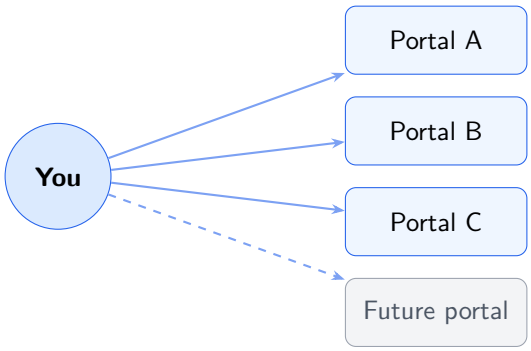

Figure 2: One pilot identity across multiple portals. The pilot's `pilot_id` is a portable cryptographic credential, not a portal-assigned account number. Every igc-net portal recognises it.

3 Your Identity, Across Every Portal

When a pilot registers on an igc-net-enabled portal, three credentials come into existence. Understanding these is the key to understanding what igc-net v0.3 gives pilots.

Your root identity (`pilot_id`). This is a cryptographic keypair generated once. The public part, expressed as an `igcnet:id:` prefixed hex string, is your stable identity across the entire network. Every flight you upload is anchored to this identifier. It never changes, even if you move to a different portal, even if your home portal disappears. Your `pilot_id` is not an account number assigned by a service. It is a cryptographic credential you own.

Your login credential (`pilot_auth_did`). This is a rotatable authentication identity, held in a digital wallet or managed by your home portal on your behalf. It is what portals check when you log in. Because it is separate from your root identity, you can rotate it if it is ever compromised without affecting your flight history.

Your access key (`private_access_keypair`). This is the credential that controls which portals can read your non-public flights. You grant it to portals you trust; you revoke it when you don't. Revocation takes effect immediately across the network. The distinction between logging in and granting private access is fundamental to the igc-net model, and is described in Section 4.

What this means in practice. Before igc-net, a pilot's identity on Portal A and Portal B were unrelated. Portals had no common language for "this is the same pilot." With igc-net, a pilot can log in to any participating portal with their `pilot_id` and carry their full flight history, publication preferences, access grants, and social connections with them. If their home portal closes tomorrow, their identity and history are recoverable on any other igc-net portal.

For technically curious readers, the full cryptographic model, including how `pilot_id` is derived, how `pilot_auth_did` is bound to it, and how rotation is handled, is specified in `10-core.md`, `60-keys-and-access.md`, and `65-pilot-auth-did.md` in the protocol specification [1].

Table 1: Publication modes and their effect on what others can see.

Mode	What anyone can see	What trusted portals can see
Public	Full raw track and all metadata	Same
Protected	Track only — personal details like pilot name and wing stripped from the public version	Full raw file with all personal details
Private	Nothing (flight existence may be announced, but content is not visible)	Full raw file

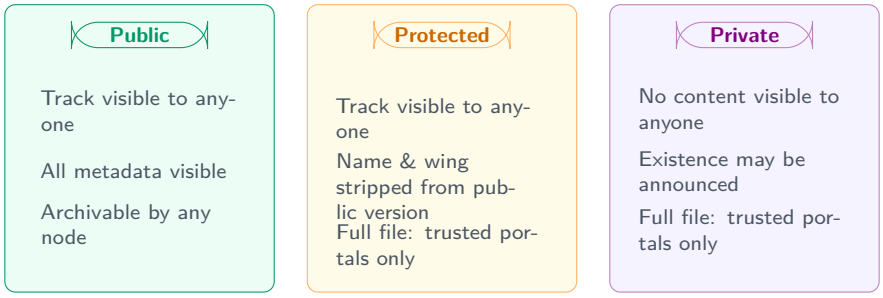

Figure 3: The three publication modes. In all cases the pilot decides; in all cases the protocol enforces.

4 Privacy and Access Control

igc-net v0.3 defines three publication modes for flights. The mode is chosen by the pilot at upload time and can be changed later. The change propagates immediately to every compliant portal.

The protected mode. A protected flight has two artifacts: a public version with personal details stripped, and a private raw companion visible only to trusted portals. This makes it possible to share a flight track publicly for scoring or community visibility without exposing the pilot's name, wing type, or other personal details to anyone who fetches the file.

Private access grants. "Trusted portals" are those to which the pilot has explicitly granted their `private_access_keypair`. This is a deliberate, separate action from logging in. A pilot may log in to a portal and see their public and protected flights without granting private access. Private access is an additional step that the pilot takes when they want a portal to see everything.

Granting is explicit. Revoking is immediate. When a pilot revokes a portal's access key — or rotates it because they no longer trust a portal that cannot be asked to co-operate — every compliant node on the network stops honouring fetch requests signed by the superseded key.

The fundamental guarantee. This is a protocol-level guarantee, not a portal's policy, not a terms-of-service promise. A portal that has lost its access key cannot read the pilot's private content, regardless of what its terms of service say. This is the meaningful difference between protocol-enforced privacy and policy-based privacy.

5 Sharing With the Right People

Publication modes give pilots control over what the whole world can see. Groups give pilots control over sharing with specific people, without changing the publication mode of any flight.

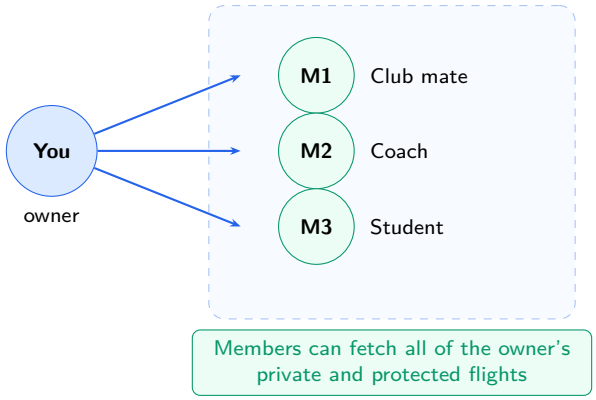

Figure 4: Private group access. The owner adds members; members can fetch the owner's non-public flights via a *GroupFetchProof*. No one outside the group can.

5.1 Private Groups

A pilot creates a private group and adds members — club mates, a coach, an expedition team. Members gain the ability to fetch all of the owner's non-public flights — past and future — without needing the owner's `private_access_keypair`. The fetch authorisation is carried in a *GroupFetchProof*, a signed credential the member presents when requesting a flight.

This has a concrete practical consequence: a pilot can keep all their training flights private from the public while making them visible to their coach, without changing any flight's publication mode, and without handing their access key to anyone.

The owner controls membership entirely. They add and remove members. Removal is immediate: a member who has been removed can no longer generate valid proofs and is refused at fetch time.

5.2 Public Groups

A public group is an opt-in collective. Any member's flights — regardless of publication mode — become accessible to all other current members. Membership is by invitation and explicit acceptance. A pilot who joins a club group makes their flights visible to all club members and gains access to all of the other members' flights in return. Leaving the group ends access in both directions.

5.3 Follow

A pilot can follow another pilot to receive notifications when they upload a new flight. Access to the followed pilot's flights still follows their publication mode, elevated automatically if the two pilots share a group.

5.4 Cross-Portal Implication

Groups and follow are part of the protocol specification, not a feature any portal has to build separately. When a portal integrates igc-net, its pilots' groups and social connections come with them. A club group whose members are spread across Portal A, Portal B, and Portal C works the same regardless of which portal each member uses. The protocol handles the proof and access enforcement; the portals observe the result.

6 What This Means for Portals

igc-net is not a competing portal. It is the open protocol layer beneath existing portals. A portal that integrates igc-net continues to own its UX, its community, its analytics, and its monetisation strategy. It gains access to everything igc-net provides.

You inherit a network, not a user base. Pilots who use igc-net bring their identity, flight history, access preferences, and social graph. A portal does not build any of that from scratch; it connects to what already exists.

Cross-portal flight discovery without scraping. A portal running an igc-net node receives announcements of new flights from every other participating portal. It can index, search, and serve those flights without bilateral API agreements, data-sharing contracts, or web scraping.

Access-control enforcement without building key management. The protocol defines when a fetch request is authorised and when it is not. A portal implementing the reference protocol layer does not need to build its own key management system, revocation logic, or group membership enforcement. It gets these from the protocol.

Group and social graph without building a friend system. Private groups, public groups, and follow are protocol primitives. A portal that integrates igc-net has them; a portal that builds them itself has to maintain them across every update to the protocol.

Archival durability. Flights indexed by archival nodes — including university-infrastructure nodes operated by NTNU DSE Lab — remain retrievable as long as at least one such node continues to serve them.

Integration path. igc-net uses BLAKE3 + Ed25519 + RFC 8785 canonical JSON. The transport is Iroh [6]. The reference implementation is a Rust library crate [3] with a gRPC sidecar interface for portal backends. Portal operators should consult the *Portal Operator Guide* [1] and the normative specification for integration requirements.

The protocol is open and versioned. The specification is the authoritative source of truth; the reference implementation is informative.

7 Who Stands to Benefit

Pilots. igc-net gives pilots what they have never had: a portable, cryptographic identity that belongs to them; control over who sees their flights at the protocol level, not just at the portal level; selective sharing with specific people without making flights public; and assurance that their history survives if any single portal disappears.

Portals. Portals that integrate igc-net join a growing network of participants. They can discover flights from other portals without scraping, use a shared identity system instead of building their own, and offer their users cross-portal privacy and group features without implementing these features themselves. The protocol is designed as enabling infrastructure, not as competition.

National federations and archives. The geographic-filtering capability from the original igc-net design (described in the technical report [2]) allows any node to automatically archive all flights whose bounding box overlaps a configured region. A national federation can maintain a complete archive of flights in its jurisdiction without bilateral agreements, additional pilot uploads, or scraping. NTNU DSE Lab is continuing work on university-hosted archival capacity for igc-net as a public-interest infrastructure service.

Researchers. A cross-platform corpus of IGC flights, with consistent content-addressed identifiers and pilot-controlled access, is a better foundation for atmospheric research, thermal modelling, site usage studies, and soaring safety analysis than a collection of platform-specific

exports. The open layer also makes research methods easier to reproduce: the `raw_igc_hash` serves as a stable, platform-independent citation target for a specific flight.

8 Protocol Status and Roadmap

Version 0.3 formalises the following capabilities, all specified in the normative documents [1]:

- Two-credential pilot identity: `pilot_id` (root) and `pilot_auth_did` (rotatable authentication DID)
- Three publication modes: `public`, `protected`, `private`
- Sanitised artifact production for protected flights (personal details stripped from the public artifact)
- `private_access_keypair` grants, rotation, and revocation
- Private groups (owner-managed; members hold *GroupFetchProof*)
- Public groups (opt-in collective; invite, accept, leave)
- Social follow (upload notifications; access follows publication mode)
- Governance plane: ownership claims, approvals, challenges, resolutions, mode changes, deletion
- Portal authentication trust model: WebAuthn PRF, passkey enrollment, broker-model PIN management
- G-record presence as a protocol field (`g_record_present`)

The reference Rust implementation [3] implements the base protocol and gRPC sidecar interface. The Python wrapper [4] provides bindings for inspection, hashing, and basic interaction.

Near-term roadmap. Version 0.4 is expected to address governance hardening (resolver federation, permissioned upgrades), incremental archive sync (efficient catch-up for new nodes joining an established network), and mobile integration (direct publication from flight instrument applications). Version 0.5 will focus on analytics: a standardised mechanism for portals to publish derived metadata — thermal annotations, wind estimates, scoring output — attached to the same shared flight identity. Version 1.0 will be the first production-stability release.

For the authoritative version history and requirement identifiers, consult the specification repository [1].

9 Conclusion

The soaring community generates a substantial and growing corpus of flight data. For most of its history, managing that data has meant accepting whatever terms the portal you uploaded to offered. Your privacy settings were a portal's settings. Your identity was an account number. Your history belonged to a database you did not control.

igc-net v0.3 changes the terms. Your `pilot_id` is yours. Your publication modes are enforced by the protocol, not by a terms-of-service page. Your club group works across portals. Your revocation of a portal's access is immediate and network-wide.

For portals, igc-net is not a threat. It is an open layer that makes their pilots' data more portable and their own integration surface more capable, without requiring them to build identity, access control, or group infrastructure from scratch.

We invite pilots, portal teams, club administrators, researchers, and national federations to read the specification, try the reference implementation, and help turn igc-net into the durable, open infrastructure layer the soaring community deserves.

Protocol specification: https://github.com/igc-net/specs

Rust implementation: https://github.com/igc-net/igc-net-rs
Python wrapper: https://github.com/igc-net/igc-net-py
Project site: https://igc-net.github.io

Acknowledgements

igc-net is developed at NTNU Distributed Systems Engineering Lab, part of the Norwegian University of Science and Technology, a public university and non-profit research environment. The authors thank the soaring community for feedback on earlier versions of the protocol design.

References

[1] igc-net Project, "igc-net Protocol Specification (v0.3)," https://github.com/igc-net/specs, 2026.

[2] Norwegian University of Science and Technology, Distributed Systems Engineering Lab, "igc-net: An Open Peer-to-Peer Protocol for Publishing, Discovering, and Exchanging IGC Flight Files," Technical Report v0.1, April 2026.

[3] igc-net Project, "igc-net-rs: Rust Reference Implementation," https://github.com/igc-net/igc-net-rs, 2025–2026.

[4] igc-net Project, "igc-net-py: Python Wrapper," https://github.com/igc-net/igc-net-py, 2025–2026.

[5] FAI/IGC, "Technical Specification for IGC-approved GNSS Flight Recorders," International Gliding Commission, 2023.

[6] n0.computer, "Iroh: A toolkit for building distributed apps," https://docs.iroh.computer, 2024–2026.

[7] XContest, "World Paragliding Rankings and Statistics," https://www.xcontest.org/world/en/, accessed April 2026.

[8] OLC, "Online Contest – Pair and Worldwide Gliding Competition," https://www.onlinecontest.org, accessed April 2026.