

SPECIFICATION DE RECONSTRUCTION SIGL

Ce texte est l'autorité de reconstruction ; le texte masqué ci-dessus est historique. Le projet contient web, ios, esp, and et mac. Web, iOS, Android et le service BLE macOS sont implémentés. Il n'y a ni cookie, ni polling, ni DID dans une URL GET. Les versions Web, iOS, Android et macOS sont indépendantes et augmentent à chaque changement de leur code.

Serveurs et stockage

aiohttp, derrière Caddy HTTPS, sert un catalogue extensible de serveurs. Chaque entrée contient au minimum hôte, titre, initiales de secours, couleur et SID ; chaque serveur possède une clé P-256 publique compressée de 33 octets dont les 6 derniers octets forment son SID. Le favicon de référence est un PNG transparent publié par son propre serveur sous GET /favicon.png. Le nombre et les noms des serveurs ne sont pas imposés par la spécification : l'application charge le catalogue binaire et adapte son affichage. Un unique fichier JSON de persistance, lu et écrit directement par app.py, conserve les collections sites(hôte,titre,initiales,couleur,SID,clé), users(target,UID,number,admin,pseudo,avatar), sessions(DID,privé,UID,target,début,total), tickets(ticket,target,UID,expiration,consommé), counters(target,UID,total,début) et backgrounds(target,UID,MIME,image). Le DID est conservé uniquement dans la session active pour adresser l'écran ; il n'est jamais recopié dans un ticket. SQLite est interdit ; JSON est réservé à ce fichier local et interdit dans les messages réseau.

Web

IHM

Sans h, GET / crée un DID aléatoire de 5 octets et rend la page publique. Elle est sombre, liquid glass, couvre les zones sûres et définit theme-color RGB. La version Web est 0.1.160 et apparaît en haut à gauche. À 8 px des coins du viewport : menu en haut à droite, deux accès fixes vers les autres serveurs en bas à gauche, domino en bas à droite. Chaque accès serveur mesure 58×58 et affiche exclusivement le PNG /favicon.png du serveur cible ou, si l'image est absente ou illisible, les initiales du titre dans un carré discret. Le succès de chargement masque explicitement les initiales et l'échec masque explicitement l'image : les deux ne sont jamais visibles ensemble. Aucun emoji de serveur n'est conservé comme secours. En page personnelle uniquement, une croix de retrait discrète de 10×10 apparaît en haut à droite de l'accès ; le bouton +, transparent à contour pointillé, ouvre à chaque ouverture une matrice recalculée uniquement composée des serveurs absents, à l'exclusion du serveur courant ; si aucun serveur ne manque, le bouton + est masqué. Cette sélection locale est persistante. Les accès personnels sont des SigLinks navigate.

Le menu Web contient Photo de fond, Spécification PDF vers /spec.pdf et une entrée Contact. Cette dernière ouvre une fenêtre superposée avec laurent.fourniercalmet@gmail.com et 06 19 29 14 85, liés respectivement par mailto:laurent.fourniercalmet@gmail.com et tel:+33619291485.

Les dimensions intrinsèques et CSS des favicons sont définies dans l'en-tête initial : 26×26 dans les accès de 58×58 et 34×34 dans le titre. Elles s'appliquent avant tout chargement d'image et interdisent qu'un favicon occupe brièvement tout le bouton.

Le catalogue livré avec cette version ajoute l'entrée Sleep (pico7.sigl.ai, initiale S, fond #1c1f26). Sa page publique applique le profil calme de NANO : fond noir, bouton central SIGL très discret qui dérive doucement, horloge discrète au-dessus, une traînée d'étoiles fines, dispersées et lentes qui suit sa position, et un panneau d'information ouvert par toucher. Cette animation est purement locale ; elle n'utilise ni cookie, ni polling, ni échange réseau supplémentaire. Le menu, les accès serveurs et

le domino restent au premier plan et inchangés ; aucune rubrique ni cadre Admin n'est rendu sur cette page publique.

Comportement

Le fond public est stocké sur son serveur avec UID nul. Le fond personnel est stocké par (target,UID). Une photo est envoyée par POST /background sous UID(4)|mime(1)|image, MIME 1 JPEG, 2 PNG, 3 HEIC, maximum 20 Mo. UID 00000000 est public. À la sélection ou au chargement, un calque de fond apparaît par opacité linéaire en 3 secondes.

IHM : domino et identité

Le domino mesure 100×100. Le QR SVG est 21×21, 100 px, modules de 4 px, quiet zone de 8 px et code strictement gatewaySID(6)|DID(5). Les neuf modules noirs arrondis du carré central 3×3 du repère supérieur droit servent d'indicateur sans modifier la géométrie du QR : ils restent noirs si aucun service BLE SIGL local n'est détecté, deviennent rouge foncé dès que le service répond, puis vert foncé seulement après acquittement de la lecture exacte des 11 octets. Ces changements sont instantanés et les teintes restent suffisamment sombres pour être lues comme des modules QR. Le toucher du QR bascule le plein écran natif de l'élément main, en public comme en personnel, sans traitement graphique particulier du QR ni modification de sa taille ; il ne provoque jamais de logout. Pendant ce plein écran, le calque photo est rattaché à main afin que le fond reste visible, puis revient à la page à la sortie. La page ouvre /ws sans paramètre et envoie DID(5) comme premier message WebSocket. Une page personnelle ajoute à gauche une bande noire translucide de 60×100 en 3 secondes. Le QR garde ses coins droits arrondis et rend ses coins gauches carrés. La bande est une grille fixe 20/60/20 px : pseudo 20 px, avatar centré dans un cadre blanc 50×50, puis compteur jaune HH:MM:SS 20 px.

Après consommation d'un ticket, la target affiche Page personnelle de <pseudo>, relit son fond, son identité et son compteur. À la création, le pseudo est tiré parmi Pierre, Paul, Jacques ; l'avatar parmi 🐱 🐼 🐝 🐉 🦋. Ils persistent par (target,UID). Toucher le pseudo le remplace dans la bande par un champ mono-ligne de 50 px, limité à 20 caractères et sans débordement. Toucher l'avatar ouvre la table : 🐱 🐼 🐝 🐉 🦋 🐱 🐼 🐝 🐉 🦋 🌞 ⚡ 🌿 🚀 💡 ★. La sauvegarde est POST /profile, binaire : UID(4)|pseudoLength(1)|pseudo_UTF8|avatar_UTF8.

Comportement : identité et compteurs

Le serveur est la référence du compteur : il reprend (target,UID), avance tant qu'il est actif, et le fige au logout ou navigate. Le Web initialise son affichage avec cette valeur et l'incrémente chaque seconde. Dans le domino, le compteur est jaune vif, monospace 10 px, centré dans sa bande fixe de 20 px ; le toucher du compteur personnel déclenche le logout et son curseur est un pointeur de lien, jamais un curseur d'édition de texte. Les compteurs bleus restent figés et ne sont jamais remis à zéro.

Protocole de communication

Le Web ouvre GET /ws et envoie DID(5), reçoit targetSID|ticket ou L, ouvre ensuite GET /?h=tickethex, et émet sur ce même WebSocket 01|targetSID(6) pour Navigate ainsi que POST /logout, /background et /profile. Indépendamment de ce protocole distant, la page tente une seule détection événementielle du service local sur https://localhost:48321/health, puis sur l'alias certifié https://127.0.0.1:48321/health. Elle maintient ensuite POST /domino sur l'adresse acceptée, avec le corps binaire exact gatewaySID(6)|DID(5). La réponse reste ouverte pendant toute la vie de la page ; l'octet 01 est l'acquiescement qui rend le module vert foncé et la fermeture du flux retire le domino. Une nouvelle tentative n'est déclenchée que par un événement de focus si aucune liaison n'est active : il n'existe ni polling ni temporisation. Les contenus et séquences sont définis dans les tableaux communs ci-dessous.

iOS

IHM

SiglPro est SwiftUI plein écran, fond sombre avec photo locale persistante AppStorage. Version en haut à gauche. Le bouton menu blanc est toujours visible au premier plan, en haut à droite ; il ouvre un panneau tactile en anglais contenant choix/suppression de photo, spécification PDF <https://pico1.sigl.ai/spec.pdf>, Contact e-mail et téléphone ci-dessus. Version, menu et journal déroulant sont ajoutés par les zones sûres haute et basse, autour des boutons : ils restent visibles, ne recouvrent pas les boutons et le journal ne passe jamais sous l'indicateur d'accueil. Le journal est en anglais, très petit, chaque ligne commence par la date complète YYYY-MM-DD HH:MM:SS, affiche le plus récent en haut et permet de faire défiler les événements anciens vers le bas. Les boutons 120×120 commencent en haut de la zone sûre, sont alignés en grille de trois colonnes, et la grille devient défilable verticalement si elle dépasse la hauteur disponible ; titres sur deux lignes au plus, favicon centré, compteur HH:MM:SS bas. La pastille du numéro utilisateur est un disque blanc portant un numéro noir, placé 3 pt au-dessus du pseudo. Le PNG de chaque favicon est enregistré dans Application Support sous un nom dérivé de sa référence et restauré avant le premier rendu ; il n'est téléchargé que si ce fichier persistant manque. Les données binaires ne sont jamais placées dans UserDefaults. En cas d'échec, les initiales du titre le remplacent, jamais un emoji de serveur. Une entrée d'index inconnue dans le catalogue binaire est ignorée sans invalider les entrées de confiance déjà reconnues. Ils sont triés par compteur synchronisé décroissant, le plus élevé en premier ; l'horloge du bouton jaune reste indépendante et avance chaque seconde. La version iOS est 0.2.70.

Comportement

États : bleu public ; orange entre la capture et la réponse HTTP 201 de /present, avec preview caméra strictement dans les 120×120. Jaune seulement dès cet ACK, texte noir et compteur en marche ; T synchronise ensuite sa valeur serveur ; bleu après L, compteur figé. Un seul jaune est autorisé. Chaque instance d'application conserve au plus une liaison /phone par couple (target,UID) et la réutilise tant qu'elle est vivante ; plusieurs téléphones, avec des UID distincts, peuvent être simultanément reliés au même serveur. Une liaison d'une target ne ferme jamais celle d'une autre target. Les callbacks d'un socket obsolète et les messages tardifs d'une autre target sont ignorés. Les fondus d'image sont strictement décoratifs : ils ne retardent ni trame, ni ACK, ni couleur de bouton.

Protocole de communication

Chaque iPhone garde un secret P-256 Keychain. Pour targetSID : $H = \text{hash_to_curve}(\text{targetSID})$, $N = xH$, $\text{UID} = \text{SHA-256}(\text{"SIGL-MINI-UID-1"} \parallel \text{targetSID} \parallel N) [0..3]$. L'UID est clair dans les messages mais n'est pas une clé publique.

Android

L'application Android reprend les mêmes états, le même catalogue binaire et les mêmes trames que l'iPhone. Elle affiche les serveurs en grille défilable de trois colonnes, conserve l'ordre du catalogue quand les compteurs sont égaux, puis les trie par compteur persistant décroissant. Chaque bouton reprend la composition iOS : titre centré sur deux lignes au plus en haut, ligne centrale avec un favicon de 42 dp puis l'identité, compteur HH:MM:SS centré en bas en 14 sp gras. Le compteur est jaune sur un bouton bleu, noir sur un bouton jaune et blanc pendant l'état orange. L'identité place la pastille du numéro utilisateur, 17 dp et texte 9 sp, immédiatement au-dessus du pseudo, sans espacement ajouté ni padding typographique. Comme l'application iOS, son activité principale autorise uniquement l'orientation portrait. La caméra s'ouvre uniquement dans le bouton sélectionné. Chaque couple (target,UID) conserve au plus un WebSocket /phone, réutilisé tant qu'il reste actif ; les liaisons de targets ou d'UID distincts restent indépendantes. L'application charge

S(1)|count(1)|siteIndex(1)|SID(6) depuis /api/sites, avec une liste de confiance locale pour les URL, et mémorise ce catalogue dans un fichier de préférences. Le PNG de chaque favicon est enregistré avec sa référence dans cette même table persistante, décodé avant la composition du premier écran et téléchargé uniquement si l'entrée manque ; en cas d'échec, les initiales du titre le remplacent, jamais un emoji de serveur. Les échanges de login, logout et navigare restent exclusivement binaires et événementiels, sans BLE, JSON ni polling. La hauteur de ligne de la pastille reste centrée pour que le glyphe soit réellement au milieu du cercle. La version Android est indépendante (0.2.12) ; version, menu, fond personnel persistant et journal compact daté restent visibles sans modifier le protocole commun.

Service BLE macOS

Le service natif macOS 0.1.2 est un exécutable sans interface lancé à l'ouverture de session. Le serveur Web SIGL est normalement distant et reste toujours joint par HTTP(S) ; ni le client Web ni le service ne supposent sa présence sur la machine de l'écran. Après son chargement HTTP(S), la page communique directement en HTTPS avec le service local, qui écoute exclusivement sur le loopback IPv4 127.0.0.1:48321. À l'installation, une autorité TLS P-256 propre à la machine signe un certificat limité aux seuls SAN localhost et 127.0.0.1, puis sa clé de signature est détruite. L'autorité est approuvée exclusivement pour TLS dans le trousseau système afin que Safari, Chrome et Firefox utilisent la même confiance ; la clé privée du certificat serveur reste dans le dossier utilisateur du service. Le service n'ouvre aucune connexion vers un serveur SIGL, ne lit aucun fichier de celui-ci et n'intervient jamais dans les applications iOS ou Android. Il accepte seulement les origines HTTPS sigl.ai et ses sous-domaines, ainsi que les origines loopback de développement. GET /health confirme sa présence. POST /domino exige exactement les 11 octets du QR ; sa réponse HTTPS binaire commence par 01, reste ouverte et représente l'autorité de vie du domino. Une connexion fermée est annulée et retirée immédiatement ; plusieurs pages correspondent à plusieurs connexions, avec déduplication des valeurs identiques.

Le service utilise CoreBluetooth en rôle périphérique. Il annonce le service primaire 5349474C-424C-4500-0000-000000000001. La caractéristique lisible 5349474C-424C-4500-0000-000000000002 renvoie D0|count(2)|domino(11)*, trié par ordre binaire. Quand aucun domino n'est ouvert, l'annonce s'arrête. Toute modification de la collection reconstruit immédiatement la valeur publiée. L'échange local, la collection et le BLE n'utilisent ni JSON, ni cookie, ni polling.

Séquences de communication Web ↔ iOS

Routes : GET /ws, GET /phone, GET /api/sites, GET /favicon.png, POST /present, /callback, /logout, /background, /profile, et GET /spec.pdf. Toutes les trames de protocole sont binaires et commencent par un opcode. GET /api/sites répond la forme compacte S(1)|count(1)|siteIndex(1)|SID(6) répétée count fois ; count et les entrées sont dynamiques. Le PNG du favicon est une ressource graphique mise en cache, pas une trame du protocole. Aucun échange de protocole n'utilise JSON.

P[113]=gatewaySID(6)|DID(5)|UID(4)|N(33)|R(33)|z(32). C'est Schnorr P-256 sur H(targetSID), hash-to-curve RFC9380 XMD SHA-256 SSWU, domaine SIGL-ZKP-V00-CS01-P256_XMD:SHA-256_SSWU_R0_, challenge SIGL-ZKP-V00-CS01-P256-SCALAR, transcript SIGL-ZKP-0-AUTH|targetSID|DID|N|R avec longueur u32 big-endian de chaque champ. La target vérifie UID, N, R et z puis crée le ticket. La gateway indiquée par gatewaySID reste seule propriétaire de la session courante du DID : à la réception du callback authentique, elle ferme uniquement cette session avant de transmettre le ticket au navigateur. Aucun état de session n'est partagé entre gateway et target. Les compteurs restent persistants par target+UID.

Login (basculer)

Les quatre rôles sont distincts : le client Web est l'écran qui affiche le QR, la gateway est le serveur de cet écran, la target est le serveur choisi par le téléphone et le téléphone est l'application iOS ou Android. Le navigateur n'effectue jamais de requête JavaScript vers un autre domaine.

Étape	De	Vers	Message / effet
1	Gateway	Client Web écran	Affiche QR binaire gatewaySID(6) DID(5)
2	Client Web écran	Gateway	Ouvre /ws puis envoie DID(5)
3	Téléphone	Gateway	Ouvre ou réutilise /phone, puis envoie UID(4)
4	Téléphone	Target	Envoie P[113] par POST /present
5	Target	Gateway	Callback serveur-à-serveur targetSID(6) DID(5) ticket(16) ; ticket unique, TTL 5 s
5b	Gateway	Ancien téléphone	Ferme la session locale du DID et envoie L UID(4) elapsed(8) si elle était personnelle
6	Gateway	Client Web écran	Envoie targetSID(6) ticket(16) sur le WebSocket DID, sans page publique intermédiaire
7	Client Web écran	Target	Navigation HTTP normale GET /?h=ticket ; la target consomme le ticket
8	Target	Téléphone	Envoie T UID(4) elapsed(8) number(2) pseudo ; le bouton devient jaune

Le QR et les trames restent binaires. Le POST /present est émis par l'application native, pas par le navigateur ; la target envoie réellement le callback par POST HTTPS /callback à la gateway indiquée par gatewaySID, sans état partagé en mémoire. La gateway traite ce callback dans un ordre atomique : elle fige la session personnelle actuellement liée au DID, envoie L à son téléphone, puis transmet le ticket sur le WebSocket de l'écran. La dernière étape Web est une navigation HTTP de premier niveau. Ce choix évite le CORS et ne crée aucun fetch inter-domaine côté écran.

Logout

Étape	De	Vers	Message / effet
1a	Client Web écran	Target	Sur page personnelle, POST /logout avec UID(4) ; le QR ou le compteur déclenche la demande
1b	Téléphone	Target	Sur bouton jaune, POST /logout avec UID(4)
2	Target	Téléphone	Fige (target, UID), puis envoie une seule trame L UID(4) elapsed(8)
3	Target	Client Web écran	Envoie L sur le WebSocket DID ; l'écran revient à la page publique

La gateway n'intervient pas dans un logout : le client Web et le téléphone parlent directement à la target. Ces requêtes sont soit de même origine pour le navigateur, soit natives pour le téléphone ; aucun CORS n'est engagé.

Navigate

Le scanner n'est jamais ouvert pendant Navigate. Au déclenchement, l'ancien bouton jaune devient immédiatement bleu et la nouvelle cible immédiatement orange, avant tout échange réseau.

Le cas général est TEL→TEL : le téléphone qui possède la session source émet 01 | targetSID(6) sur son WebSocket /phone. La source fige sa session et envoie L, puis 0 au téléphone ; celui-ci se connecte à la target, envoie P, et la target termine le handoff par callback, ticket et T. Le cas même téléphone réutilise exactement ce flux avec le même UID. Le cas Web→TEL est une forme

simplifiée : le Web envoie 01|targetSID(6) sur le WebSocket /ws de l'écran ; le serveur retrouve l'UID de la session et déclenche ensuite le même flux téléphone L → 0 → P → ACK.

Étape	De	Vers	Message / effet
1a	Téléphone	Source /phone	Envoie 01 targetSID(6) sur la liaison existante
1b	Client Web écran	Source /ws	Un SigLink envoie 01 targetSID(6) sur la liaison existante
2	Source	Téléphone	Figé la session source, envoie L UID(4) elapsed(8), puis 0 targetSID(6) gatewaySID(6) DID(5)
2b	Source	Client Web écran	L'écran reste ouvert jusqu'au ticket ; il navigue vers la nouvelle target au callback, sans redirection publique intermédiaire
3	Téléphone	Target	Reconstruit et envoie P[113] à POST /present ; aucune caméra
4	Target	Gateway	Callback serveur-à-serveur targetSID(6) DID(5) ticket(16)
4b	Gateway	Ancien téléphone	Si la session du DID est encore personnelle, la fige et envoie L UID(4) elapsed(8)
5	Gateway	Client Web écran	Envoie targetSID(6) ticket(16) sur le WebSocket DID
6	Client Web écran	Target	Navigation HTTP normale GET /?h=ticket
7	Target	Téléphone	Envoie T UID(4) elapsed(8) number(2) pseudo ; la cible devient jaune

La gateway possède le DID : elle ferme sa session locale encore active, puis relaie le ticket sur le WebSocket déjà ouvert. Dans un navigateur initié par le téléphone ou le Web, la source a normalement déjà été figée ; l'étape est alors sans effet supplémentaire. Dans une prise d'écran directe par un second téléphone, cette même étape fournit le L qui déconnecte proprement l'ancien téléphone. Le navigateur ne fait aucun appel JavaScript cross-origin ; le callback target→gateway est serveur-à-serveur et la transition de domaine est une navigation HTTP normale.

Le calque de fond iOS est un arrière-plan plein écran, non interactif et exclu du calcul de mise en page : son chargement ne déplace jamais les icônes ni les boutons. La version iOS est affichée en très petit dans un cadre transparent à bord discret.

Lorsqu'un nouveau téléphone présente une preuve pour prendre le même écran, la target ne tente jamais d'accéder à la session de la gateway. Son callback fait fermer par la gateway la session personnelle du DID et envoyer L à l'ancien iPhone avant que le ticket soit transmis à l'écran. Les sessions des autres DID restent actives.

L'ordre est strictement événementiel : vérification de P et création du ticket par la target, callback HTTPS vers la gateway, fermeture de la session du DID et émission éventuelle de L par la gateway, notification WebSocket de l'écran, navigation GET /?h=... , puis émission de T au nouveau téléphone. Aucun délai artificiel ni acquittement anticipé n'est autorisé.

Sur les iPhone étroits, l'espacement horizontal entre les trois boutons est recalculé selon la largeur disponible, jusqu'à zéro, afin d'éviter tout débordement tout en conservant les boutons centrés.

Pendant scanning, un second appui sur le même bouton arrête la caméra, annule le scan et remet immédiatement le bouton à l'état bleu public. Pendant waitingAck, l'appui reste sans effet.

Lorsqu'un nouveau téléphone ouvre une page personnelle sur le même écran, la gateway retrouve la session précédente par le seul DID, indépendamment de la target choisie. Cette session reçoit L, son compteur est figé, puis le même écran ouvre la nouvelle page sans page publique intermédiaire.

Les noms des serveurs sont centrés sur une ou deux lignes au maximum, avec ellipse et sans débordement, dans les accès de page et dans la matrice d'ajout. Le cadre de la matrice adopte exactement le nombre de colonnes correspondant aux boutons ajoutables, sans colonne vide. En page personnelle, une petite pastille sombre affiche le numéro séquentiel de l'utilisateur sur le coin de l'avatar sans modifier l'avatar sauvegardé ; le premier utilisateur du serveur est le numéro 1 et l'administrateur. Ce numéro appartient à l'identité (serveur, UID), pas à une session : le même utilisateur conserve le même numéro sur plusieurs écrans.

À l'initialisation de l'app iOS, un fond de couleur aléatoire est choisi une seule fois pour cette instance. Il couvre tout l'écran sous les contrôles ; une photo de fond sélectionnée ultérieurement le masque entièrement sans déplacer les contrôles.

La version, le menu et le journal iOS sont des superpositions explicites alignées respectivement en haut et en bas de la vue racine ; ils restent donc visibles au-dessus du fond coloré ou photographique et des boutons.

Un T iOS est T|UID(4)|elapsed(8)|number(2)|pseudoLength(1)|pseudo ; il n'est accepté que pour un bouton orange ou déjà jaune. Tout T retardé après un L est ignoré. Le L reçu pour le bouton personnel ou en transition le repasse immédiatement au bleu et fige son compteur.

Le traitement de L repasse au bleu uniquement le bouton du couple (target, UID) porté par le WebSocket qui l'a reçu. Il ne modifie aucun bouton appartenant à une autre target.

La fermeture de cette liaison est elle-même un événement d'autorité : le serveur ferme les sessions privées de son couple (target, UID) et l'application repasse uniquement le bouton correspondant au bleu. Une rupture WebSocket n'est jamais ignorée et ne déclenche ni attente, ni reconnexion, ni polling.

Les liaisons téléphone sont indexées côté serveur par (target, UID) et non par le seul UID. Pour deux téléphones A et B connectés au même serveur, WS_A et WS_B restent distinctes. La gateway adresse L avec l'UID de sa session DID courante ; après consommation du ticket, la target adresse T avec l'UID du nouveau téléphone.

Chaque bouton de l'application native affiche, à côté du favicon, la pastille du numéro number reçu pour ce serveur ; cette valeur est celle de l'identité persistante (serveur, UID), jamais l'UID lui-même. La numérotation est séquentielle et indépendante sur chaque serveur : le premier utilisateur est 1 et administrateur, puis chaque nouvelle identité reçoit 2, 3, 4, etc., dans la limite du champ binaire non signé de 2 octets. Un numéro déjà attribué reste stable.

Le pseudo reçu dans ce même ACK est affiché en très petit sous la pastille du bouton iPhone, tronqué sur une ligne si nécessaire ; aucun pseudo n'est affiché avant la première visite du serveur.

Le chrono des boutons iOS utilise une police monospace de 14 px ; le journal déroulant occupe toute la largeur de l'écran, sans marge horizontale.

Sur iOS, ce pseudo est noir lorsque le bouton est jaune. Sur le Web, la page personnelle de l'administrateur (numéro 1) affiche en plus un tableau incluant la ligne 1 – Admin, puis les autres utilisateurs du serveur avec numéro, avatar et pseudo.

La barre supérieure iOS reste alignée en haut mais est décalée de 22 px vers l'intérieur de l'écran afin de ne pas toucher l'encoche.

Le catalogue comprend aussi `pico4.sigl.ai` (Vigie Baignade, initiales VB, #2563eb). Les quatre boutons iOS sont remplis en ordre de lecture : rangées de trois, gauche à droite dans chaque rangée puis rangée suivante.

La grille iOS conserve un espacement unique entre les boutons et entre les boutons et les bords de l'écran. Cet espacement est calculé selon la largeur disponible ; les cartes rétrécissent ensemble sur les écrans étroits.

Pour Navigate, l'iPhone réutilise uniquement une liaison `/phone` dont l'état est actif. Si cette liaison n'est plus active ou si l'envoi échoue, la demande est refusée explicitement ; aucune reconnexion, mise en attente, trame de secours ni polling n'est généré.

Lorsqu'une nouvelle liaison `/phone` reprend le même UID sur une target, le serveur sélectionne uniquement les sessions privées de cette target, fige leurs compteurs, envoie une seule trame `L|UID(4)|elapsed(8)` directement aux anciennes liaisons, puis ferme celles-ci. Les sessions du même utilisateur sur d'autres targets restent inchangées.

Dans le Web, la page personnelle de l'admin (numéro 1) comporte la configuration de la page publique de son serveur : image de fond publique, couleur de fond et sous-titre. Ces réglages sont persistés par serveur dans un fichier et chaque rendu de page relit la valeur courante ; un sous-titre déjà enregistré ne redevient donc jamais « Page publique ». L'image publique est distincte de l'image personnelle de l'admin. Aucun sélecteur de photo n'est affiché sur une page publique.

Cette configuration précède la table des utilisateurs dans le même cadre admin ; ses champs sont compacts, sans second encadrement ni marge inutile.

Sur iOS, les numéros et pseudos reçus dans T sont persistés par URL de serveur. Cette URL reste la référence de l'identité affichée ; les valeurs sont restaurées au lancement et ne sont jamais partagées entre serveurs.

Un pseudo est tiré aléatoirement uniquement lors de la première création de l'identité (`target,UID`). Après une édition ou une restauration, la valeur persistée est conservée telle quelle ; elle n'est jamais remplacée par un nouveau nom prédéfini.

Lorsqu'une nouvelle liaison `/phone` reprend le même couple (`target,UID`), chaque session privée de ce couple est d'abord figée et reçoit `L|UID|elapsed`, puis l'ancienne liaison est fermée. Les liaisons des autres UID sur la même target restent ouvertes.

Disposition admin Web

Sur la page personnelle de l'administrateur (utilisateur 1), la rubrique Admin occupe elle aussi une page entière, avec un trait fin de séparation. Elle contient deux cadres empilés verticalement, configuration de la page publique puis table des utilisateurs, sous un séparateur « Admin » ; leur largeur et leur hauteur de contenu restent mesurées pour éviter de gros blocs. Les boutons des autres serveurs et le domino QR restent tous deux en footer fixe, toujours visibles et au-dessus de toute rubrique ou contenu. Sur écran étroit, les deux cadres conservent cet empilement sans débordement. Le menu propose un accès direct à la rubrique Admin et son panneau reste au-dessus du contenu ; cet accès reste local et ne provoque aucun rechargement réseau. Cette disposition ne s'affiche pas sur une page publique ni sur une page personnelle non administrateur.

Règle générique du catalogue et des clients

La spécification ne dépend d'aucun nom, domaine, favicon, couleur ni nombre de serveurs. Un serveur est une entrée autonome du catalogue (`host,title,initials,colour,SID,publicKey`) ; l'ajout ou le retrait d'une entrée ne change ni les opcodes ni les séquences. Chaque instance Web

publie son entrée et chaque client charge la liste binaire S | count | index | SID avant d'afficher les accès disponibles.

Sur iOS comme sur Android, l'application cliente représente toutes les entrées reçues, les trie selon le compteur persistant, les place en grille de trois colonnes lorsque la largeur le permet et conserve l'ordre de lecture gauche-droite puis haut-bas. Les états bleu/orange/jaune, le scanner limité au bouton, le WebSocket événementiel unique par liaison et les trames binaires sont identiques ; seuls les composants graphiques, le sélecteur de caméra et le stockage local propres à la plateforme diffèrent.

Le serveur Web doit créer automatiquement les clés et la persistance d'une nouvelle entrée de catalogue au premier démarrage, et Caddy doit simplement router son hôte vers le même service. Aucune constante de code ou de spec ne doit limiter le catalogue à trois, quatre ou six serveurs.