

Centro Notifiche email in-app

Invio “per conto” dell’utente, deliverability e logo nell’email

Guida operativa: architettura d’invio con Resend, autenticazione del dominio (SPF / DKIM / DMARC) e logo del brand nella casella (BIMI + VMC).

Stack: microfrontend SIS + Cloudflare · Provider email: Resend · DNS: Cloudflare

Documento interno — v1

Indice

1. Obiettivo
2. Strategia d’invio (“on behalf of”)
3. Il form e il mittente
4. Flusso tecnico ed endpoint
5. Invio a gruppi (mailing list)
6. Deliverability: SPF · DKIM · DMARC
7. Logo nell’email: BIMI + VMC
8. Checklist finale
9. Appendice: record ed esempi

1. Obiettivo

Un **Centro Notifiche** dentro l’app dove un utente loggato compone un messaggio e lo invia via email a un **gruppo selezionato** (mailing list ricavata dai dati: studenti, referenti, docenti di una classe, ecc.).

Vincolo chiave: l’email deve risultare inviata **dall’utente loggato** — il destinatario vede l’utente come mittente e, rispondendo, scrive direttamente a lui.

Decisione presa

Modalità

“on behalf of”

(opzione A) su

Resend

: nessun OAuth, nessuna richiesta di DNS alle scuole.

2. Strategia d’invio (“on behalf of”)

Ogni email viene spedita dal **tuo** dominio verificato in Resend, ma “vestita” con l’identità dell’utente:

```
From: "Mario Rossi – Scuola X <notifiche@tuodominio.it>" // dominio TU0, verificato
Reply-To: mario.rossi@sua-email.it // le risposte tornano all'utente
Subject: ...
Body: ... + firma "Inviato da Mario Rossi tramite School Manager"
```

- Il destinatario **vede il nome dell’utente** e, se risponde, scrive **all’utente**.
- SPF/DKIM/DMARC restano allineati al **tuo** dominio → alta deliverability, niente spam.

Regola d'oro di sicurezza

Il mittente (display name) e il `Reply-To` li imposta

il backend leggendoli dal token

dell'utente loggato —

mai

dal client. Così nessuno può spacciarsi per un altro.

Perché non il `From` letterale `@scuola.it`

Mostrare l'indirizzo reale nel `From` (stile QuickSchools, "via relay...") richiede un **relay proprio** che accetti From arbitrari (Resend non lo permette: pretende il dominio del From verificato). È un progetto infrastrutturale a parte (reputazione IP, warm-up, monitoraggio). Per partire si usa **nome + Reply-To**: il risultato pratico — nome giusto e risposte all'utente — è lo stesso per il 99% dei casi.

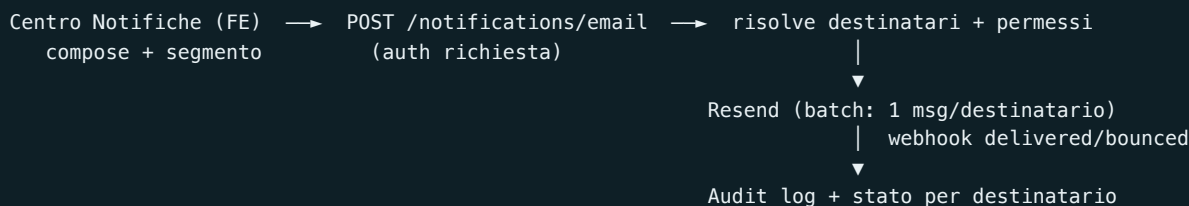
3. Il form e il mittente

Il form **non** ha un campo dove digitare l'email mittente (sarebbe spoofing). Il mittente arriva **dal profilo** dell'utente loggato ed è mostrato in **sola lettura**, solo per trasparenza.

```
Invii come: Mario Rossi | ← read-only, dal profilo
                mario.rossi@scuola.it
A (gruppo): [ Classe 1A LS ▼ ] · 24 destinatari
Oggetto:    [ _____ ]
Messaggio:  [ _____ ]
                        [ Invia ]
```

- **Utente senza email a profilo** → non la si chiede nel form: si blocca l'invio con "aggiungi un'email al tuo profilo".
- **Utente con più indirizzi** (raro) → eventuale *select* tra indirizzi già verificati e legati all'account, mai testo libero. Per la v1 non serve.

4. Flusso tecnico ed endpoint



Contratto endpoint (bozza)

```
POST /api/v1/notifications/email // richiede utente autenticato
{
  "segment": { "type": "homeroom", "id": "...", }, // oppure recipientIds espliciti
  "subject": "...",
  "bodyHtml": "...",
}
→ 202 { "messageId": "...", "recipientCount": 24, "accepted": 24, "rejected": 0 }
```

Il backend: **(1)** risolve il segmento in indirizzi, **filtrando su ciò che il mittente ha il permesso di contattare** (un docente → le sue classi; un admin → tutti); **(2)** deriva mittente e Reply-To dal JWT; **(3)** invia via Resend; **(4)** registra l'audit.

5. Invio a gruppi (mailing list)

1 messaggio per destinatario

, non BCC unico. Usa la

batch API di Resend

(fino a 100 messaggi per chiamata).

Perché per-destinatario

- Privacy: nessuno vede la lista altrui
- Personalizzazione (nome, link)
- Stato e bounce **per singolo**
- Nessun limite BCC

Robustezza

- **Rate limit**: coda per liste grandi
- **Webhook** delivered/bounced/complained → aggiorna stato
- **Suppression list**: non ri-mandare ai bounce
- **Audit**: chi ha inviato cosa a chi

Se in futuro diventa comunicazione "di massa": header `List-Unsubscribe` + link di disiscrizione (obbligo anti-spam). Per comunicazioni interne piccole si può partire più leggeri.

6. Deliverability: SPF · DKIM · DMARC

Non è un servizio a cui "accreditarsi": sono **record DNS sul tuo dominio d'invio** (nel vostro caso su **Cloudflare**). Da fare **una volta**. Nulla da chiedere alle scuole.

Record	Chi lo fornisce	A cosa serve
DKIM	Resend (dashboard, verifica dominio)	Firma crittografica dell'email
SPF / Return-Path	Resend	Autorizza i server d'invio
DMARC	Lo scrivi tu (TXT su <code>_dmarc</code>)	Politica + report d'allineamento

```
_dmarc.tuodominio.it  TXT  "v=DMARC1; p=none; rua=mailto:dmarc@tuodominio.it; adkim=s; aspf=s"
```

Ordine di attivazione (importante)

- 1) `p=none` (solo monitoraggio, non blocca) → 2) verifica dai report che DKIM/SPF siano allineati per 1–2 settimane →
- 3) `p=quarantine` → 4) `p=reject`.

L'enforcement (quarantine/reject) è prerequisito del logo BIML.

Leggere i report DMARC (opzionale)

Il `rua=` genera report XML illeggibili a mano. Per vederli in chiaro (facoltativo): **Cloudflare DMARC Management** (gratis, siete già su Cloudflare), oppure EasyDMARC, dmarcian, Postmark DMARC, Valimail. Servono solo a *leggere* i report; DMARC funziona comunque.

7. Logo nell'email: BIMI + VMC

BIMI (*Brand Indicators for Message Identification*) fa comparire il **logo del brand come avatar** accanto al mittente in Gmail, Apple Mail, Yahoo. Il logo mostrato è quello del **tuο dominio** d'invio (quindi il logo dell'app, uguale per tutte le mail — non un logo per-scuola).

Requisiti (servono tutti)

1. **DMARC in enforcement**: `p=quarantine` o `p=reject` (non `none`).
2. **Logo in SVG Tiny 1.2 (profilo SVG P/S)**, quadrato — una variante specifica di SVG, non quello comune.
3. **VMC — Verified Mark Certificate** **A PAGAMENTO**: certifica che il logo è tuo.
4. Record DNS **BIMI** (`default._bimi.tuodominio.it`) che punta all'SVG + al certificato.

Chi si paga

Il VMC lo emettono

solo due autorità

:

DigiCert

ed

Entrust

. Costo indicativo

~1.000–1.500 \$/anno

. Gmail/Apple poi lo leggono gratis.

Vincolo del marchio

Il VMC richiede che il logo sia un

marchio registrato ®

(trademark depositato). Se "Empowered School Manager" non è registrato, esiste il

CMC

(*Common Mark Certificate*, sempre DigiCert/Entrust) per loghi non registrati, ma con supporto più limitato (Gmail privilegia il VMC + spunta blu).

```
default._bimi.tuodominio.it TXT "v=BIMI1; l=https://tuodominio.it/bimi/logo.svg; a=https://tuodominio.it/bimi/vmc.pem"
```

BIMI è un *nice-to-have* di branding: **non serve** per far arrivare le email — quello lo garantiscono già SPF/DKIM/DMARC.

8. Checklist finale (in ordine)

Fase 1 — Invio funzionante

- ☐ Verifica del **dominio d'invio** in Resend
- ☐ Aggiungi in **Cloudflare** i record **DKIM** + **SPF/Return-Path** forniti da Resend
- ☐ Endpoint `POST /notifications/email`: mittente/Reply-To dal **token**, segment-resolver con permessi
- ☐ Invio **batch, 1 messaggio/destinatario**; form con mittente **read-only** dal profilo
- ☐ **Webhook** Resend → stato per destinatario + **suppression list** + **audit log**

Fase 2 — Autenticazione robusta

- ☐ Pubblica **DMARC** `p=none` con `rua`

- ☐ Monitora i report 1–2 settimane (Cloudflare DMARC Management)
- ☐ Passa a `p=quarantine` → poi `p=reject`

Fase 3 — Logo nell’email (opzionale)

- ☐ Registra il marchio ® (se non già fatto) — prerequisite del VMC
- ☐ Converti il logo in **SVG P/S** quadrato
- ☐ Acquista il **VMC** da **DigiCert** o **Entrust**
- ☐ Pubblica il record **BIMI** (logo + certificato)

9. Appendice: record ed esempi

Esempio: invio “on behalf of” con Resend (batch)

```
// Backend – pseudocodice
const sender = currentUser(); // dal JWT, non dal client
if (!sender.email) throw Error("Mittente senza email a profilo");

const recipients = resolveSegment(body.segment, sender.permissions); // filtra sui permessi

await resend.batch.send(
  recipients.map(r => ({
    from: `${sender.fullName} – ${schoolName} <notifiche@tuodominio.it>`,
    replyTo: sender.email,
    to: r.email,
    subject: body.subject,
    html: body.bodyHtml + footer(sender),
  }))
);
```

Riepilogo record DNS (sul tuo dominio, su Cloudflare)

Nome	Tipo	Valore (esempio)
(forniti da Resend)	CNAME/TXT	DKIM + Return-Path della verifica dominio
<code>_dmarc</code>	TXT	<code>v=DMARC1; p=none; rua=mailto:dmarc@tuodominio.it; adkim=s; aspf=s</code>
<code>default._bimi</code>	TXT	<code>v=BIMI1; l=.../logo.svg; a=.../vmc.pem</code> (fase 3)

Sintesi: **Resend** (verifica dominio) + **Cloudflare** (DKIM/SPF/DMARC del tuo dominio) → invio “per conto” dell’utente con nome + Reply-To. DMARC in `reject` abilita, in fase 3, il logo nell’email via **BIMI + VMC** (DigiCert/Entrust). Nessun DNS richiesto alle scuole.