# JC - Gateway e Monitor servizi

La porta da cui passa ogni richiesta e l'elenco dei servizi accesi.

**Cosa ci guadagni.** Un solo punto di controllo: sicurezza e instradamento si decidono una volta, non in ogni programma.


- JC1 - A cosa servono
  - JC1.1 - Una porta sola per tutto
  - JC1.2 - Dove stanno nella mappa
- JC2 - Il viaggio di una richiesta
  - JC2.1 - Dal browser al servizio giusto
  - JC2.2 - Come si controlla chi sei
  - JC2.3 - Le due strade: Gateway e Ponte col COBOL
- JC3 - Monitor servizi
  - JC3.1 - Vedere quali servizi sono accesi
  - JC3.2 - Cosa succede se un servizio è spento
- JC4 - Per chi sviluppa
  - JC4.1 - Aggiungere una rotta
  - JC4.2 - Errori 401, 403 e 503: cosa vogliono dire
  - JC4.3 - Se qualcosa non va

# JC1 - A cosa servono

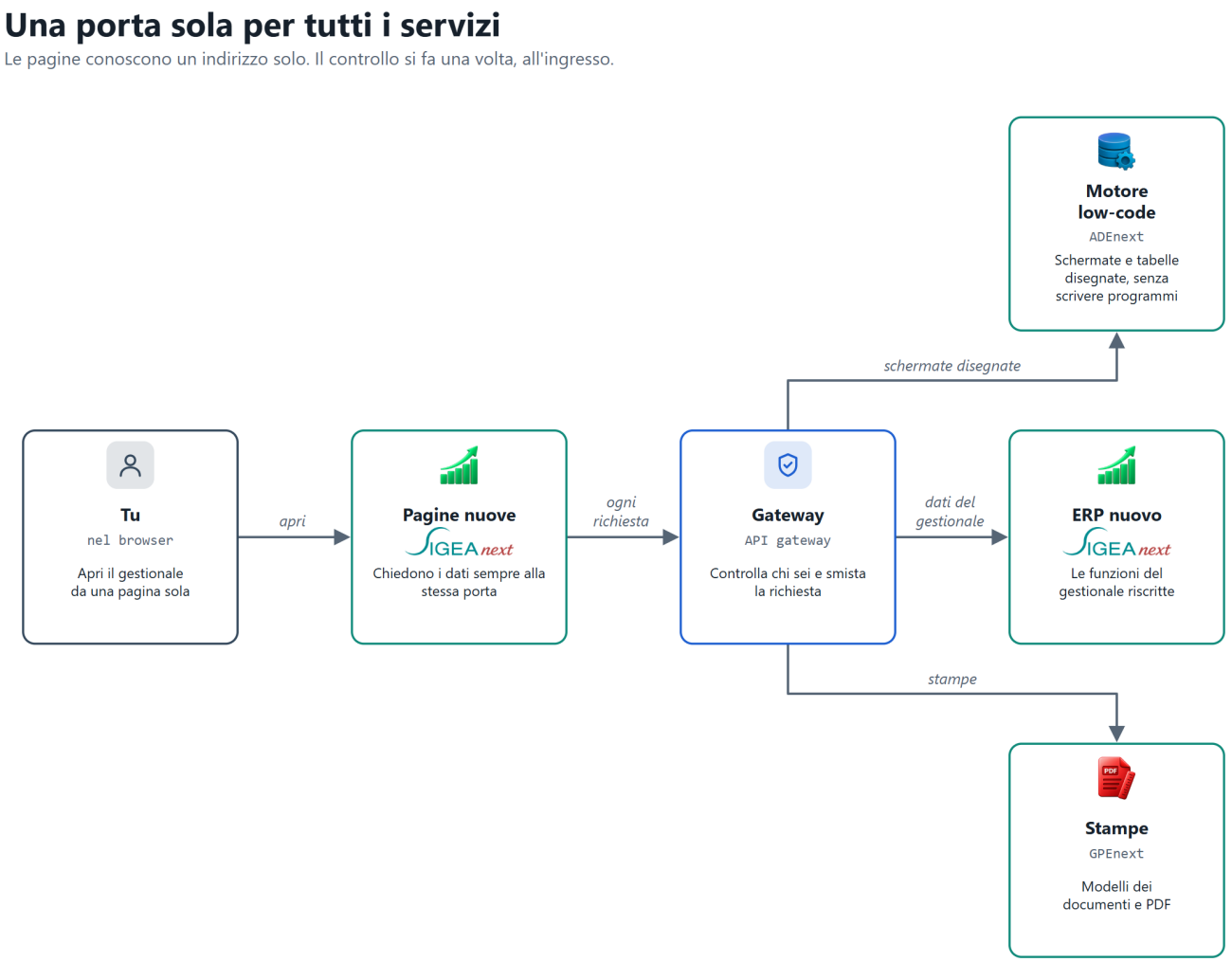
# JC1.1 - Una porta sola per tutto

**In una frase.** Il Gateway è l'unica porta da cui le pagine nuove chiedono i dati; il Monitor servizi è l'elenco che gli dice quali servizi sono accesi e dove stanno.

ADSWnext non è un programma solo. È fatto di più servizi, ognuno con il suo compito: uno tiene le schermate disegnate, uno i dati del gestionale, uno le stampe. Se ogni pagina dovesse sapere dove sta ogni servizio, e ogni servizio dovesse farsi da solo i controlli d'ingresso, basterebbe una dimenticanza per lasciare una porta aperta.

Per questo c'è una porta sola. Le pagine mandano ogni richiesta al **Gateway**. Lui controlla che tu sia entrato, guarda cosa stai chiedendo e la consegna al servizio giusto. Per sapere dove sta quel servizio consulta il **Monitor servizi**: un elenco che i servizi stessi tengono aggiornato, perché ognuno si iscrive quando si accende.

Tu non vedi niente di tutto questo. Vedi la pagina che si riempie.



*Le pagine chiedono tutto a una porta sola: il Gateway controlla e smista verso i servizi*

> **Cosa ci guadagni.**
> • **I controlli d'ingresso si fanno in un punto solo.** Chi non è entrato si ferma al Gateway e non arriva a nessun servizio. Una regola cambiata lì vale subito per tutti.
> • **Un servizio nuovo si aggiunge senza toccare gli altri.** Si accende, si iscrive al Monitor servizi, e al Gateway basta una riga che dice quali richieste sono sue. Le stampe di GPEnext sono entrate così.
> • **Se un servizio è spento, il resto lavora.** Ferme le stampe, le tabelle si aprono lo stesso. Dopo un aggiornamento ogni servizio torna disponibile da solo, appena si è iscritto di nuovo.

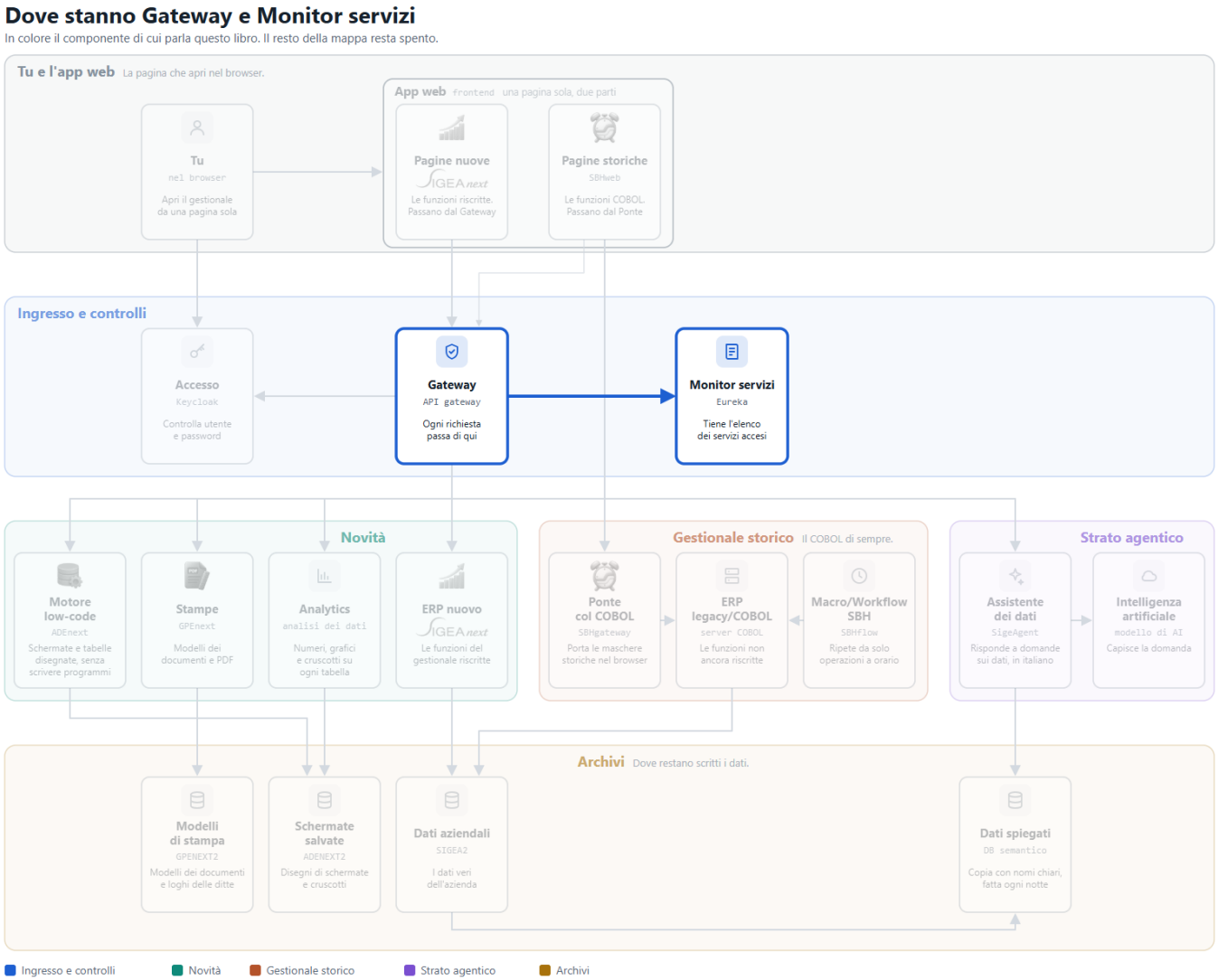
Questo libro racconta il viaggio di una richiesta, come si guarda quali servizi sono accesi e cosa vogliono dire gli errori 401, 403 e 503.

> **Per chi sviluppa.** Il Gateway è `backend/middleware/it_gsba_api_gateway` (Spring Cloud Gateway, porta 8080). Il Monitor servizi è `backend/middleware/it_gsba_eureka_server` (Eureka, porta 8761). Le rotte stanno in una classe sola, `GatewayConfig`.

# JC1.2 - Dove stanno nella mappa

**In una frase.** Gateway e Monitor servizi stanno nello strato «Ingresso e controlli», fra le pagine che apri e i servizi che rispondono.

Sapere dove stanno aiuta a capire una cosa: i servizi non si chiamano dal browser. Dal browser si arriva solo al Gateway, e da lì si prosegue.




*La mappa di ADSWnext con accesi il Gateway e il Monitor servizi*

**Chi chiama il Gateway.** Le pagine nuove di SIGEAnext, per ogni cosa: aprire una tabella, salvare, stampare, analizzare. Le pagine storiche di SBHweb lo chiamano solo per la stampa in PDF; le maschere fanno un'altra strada, raccontata in Le due strade: Gateway e Ponte col COBOL.

**Chi chiama lui.** I servizi delle Novità: ADEnext per le schermate disegnate e i cruscotti di Analytics, SIGEAnext per i dati del gestionale, GPEnext per le stampe. Dove è acceso, anche l'Assistente dei dati. Per controllare il tuo lasciapassare usa le chiavi dell'Accesso.

**Chi usa il Monitor servizi.** I servizi, che si iscrivono all'elenco quando si accendono. E il Gateway, che lo consulta per sapere a quale indirizzo inoltrare. Tu non lo usi mai direttamente: lo guarda solo chi gestisce il server.

**Cosa vuol dire per te.** Entri una volta e il controllo vale per tutti i servizi. Gli archivi non si raggiungono mai dal browser: in mezzo ci sono sempre il Gateway e un servizio.

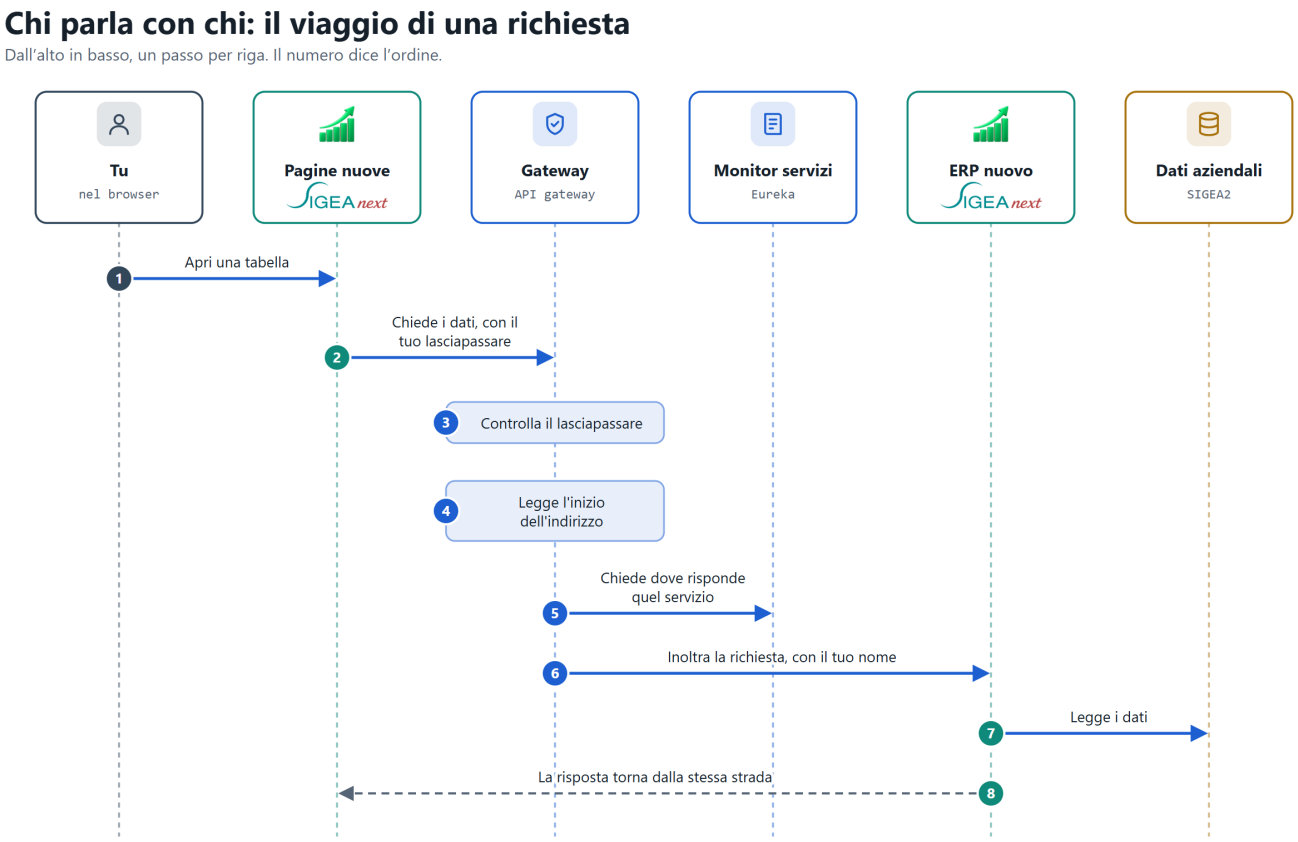
> **Per chi sviluppa.**
> • Tessere della mappa: `gateway` (porta 8080) e `eureka` (porta 8761).
> • Il Gateway non emette lasciapassare: li emette solo l'Accesso (Keycloak). Il Gateway li verifica.
> • Sul server le porte dei servizi interni e del Monitor servizi sono aperte solo dal server stesso. Da un altro PC si entra dall'app web e dal Gateway.

# JC2 - Il viaggio di una richiesta

# JC2.1 - Dal browser al servizio giusto

**In una frase.** Come fa una richiesta partita dalla tua pagina ad arrivare proprio al servizio che la sa soddisfare, e a tornare con la risposta.

Ogni volta che apri una tabella, premi **Salva** o chiedi una stampa, la pagina manda una richiesta. Le richieste sono tante e i servizi sono diversi. Il Gateway le smista tutte con una regola sola, semplice da controllare: guarda come comincia l'indirizzo.




*Il viaggio di una richiesta: dalla pagina al Gateway, al servizio, e ritorno*

Il viaggio, in ordine.

**La pagina chiede.** La richiesta parte dal browser e porta con sé il tuo lasciapassare, cioè la prova che sei entrato.

**Il Gateway controlla.** Se il lasciapassare manca o non è valido, il viaggio finisce qui. Come lo controlla è in Come si controlla chi sei.

**Il Gateway legge l'inizio dell'indirizzo.** Ogni servizio ha il suo:

| L'indirizzo comincia con | La richiesta va a | Per fare cosa |
|---|---|---|
| `/api/ade` | ADEnext | Schermate disegnate, cruscotti di Analytics |
| `/api/sigea` | SIGEAnext | Dati del gestionale |
| `/api/gpe` | GPEnext | Stampe e modelli |
| `/api/sigeagent` | Assistente dei dati | Domande sui dati (solo dove è acceso) |

**Chiede al Monitor servizi dove sta quel servizio.** Il Gateway conosce il nome del servizio, non il suo indirizzo. L'indirizzo lo prende dall'elenco.

**Inoltra.** Toglie dall'indirizzo il pezzo iniziale, che serviva solo a lui, scrive sulla richiesta il tuo nome utente e la consegna.

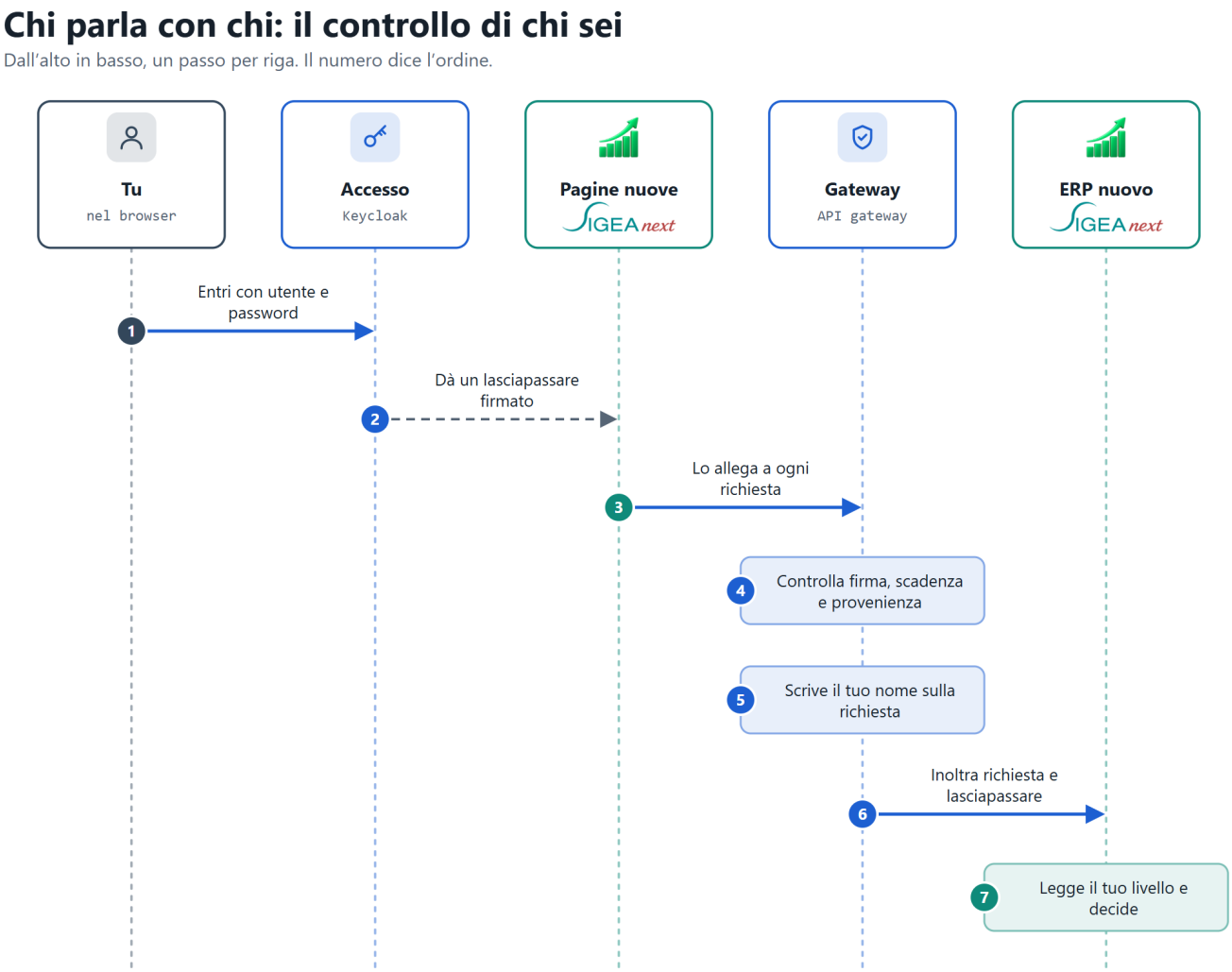
**La risposta torna dalla stessa strada.** Il servizio risponde al Gateway, il Gateway alla pagina.

> **Per chi sviluppa.**
> • Le rotte sono in `backend/middleware/it_gsba_api_gateway/src/main/java/it/gsba/api_gateway/config/GatewayConfig.java`. Ognuna ha `stripPrefix(2)` e il filtro `AuthenticationFilter`.
> • La destinazione è `lb://NOME-DEL-SERVIZIO`: il nome con cui il servizio si è iscritto al Monitor servizi.
> • L'Assistente dei dati fa eccezione: ha un indirizzo fisso e la sua rotta esiste solo con `SIGEAGENT_ENABLED=true`.
> • Davanti al Gateway c'è il server web dell'app (`frontend/docker/nginx.conf`), che gli passa tutto ciò che comincia per `/api/`. Le richieste lunghe possono durare fino a dieci minuti.

# JC2.2 - Come si controlla chi sei

**In una frase.** Il Gateway non ti chiede la password: controlla il lasciapassare che hai ricevuto entrando, e passa ai servizi il tuo nome.

Utente e password li scrivi una volta sola, nella pagina dell'Accesso. In cambio ricevi un **lasciapassare**: un piccolo documento firmato che dice chi sei e che livello hai, e che scade dopo poco. Da quel momento la pagina lo allega a ogni richiesta. La password non viaggia più.




*Il controllo di chi sei: l'Accesso firma il lasciapassare, il Gateway lo verifica, il servizio legge il livello*

Per ogni richiesta il Gateway controlla quattro cose:

- che la **firma** sia quella dell'Accesso, quindi che il lasciapassare non sia stato fabbricato o ritoccato;
- che non sia **scaduto**;

- che venga dall'**Accesso di questa installazione** e non da un altro;
- che l'abbia chiesto un **programma ammesso**, cioè l'app web o uno dei servizi.

Se una sola non torna, la richiesta si ferma con l'errore 401 e nessun servizio la vede.

Se tutto torna, il Gateway scrive sulla richiesta il tuo **nome utente** e la inoltra insieme al lasciapassare. Prima però cancella qualunque nome fosse già scritto: così nessuno può presentarsi ai servizi con il nome di un altro.

Il **livello** (lettore, base, avanzato, staff) sta dentro il lasciapassare. Lo legge il servizio che riceve la richiesta, e decide lui se quella cosa la puoi fare. Se non puoi, risponde con l'errore 403. I livelli sono spiegati in Lettore, base, avanzato, staff (libro «JB - Accesso e permessi», pagina «JB3.1 - Lettore, base, avanzato, staff»).

> **Nota.** Poche cose passano senza lasciapassare, perché chi le chiede non può averlo: il controllo «sei acceso?» che il server fa ai servizi, le immagini dentro una stampa, la procedura di prima installazione.

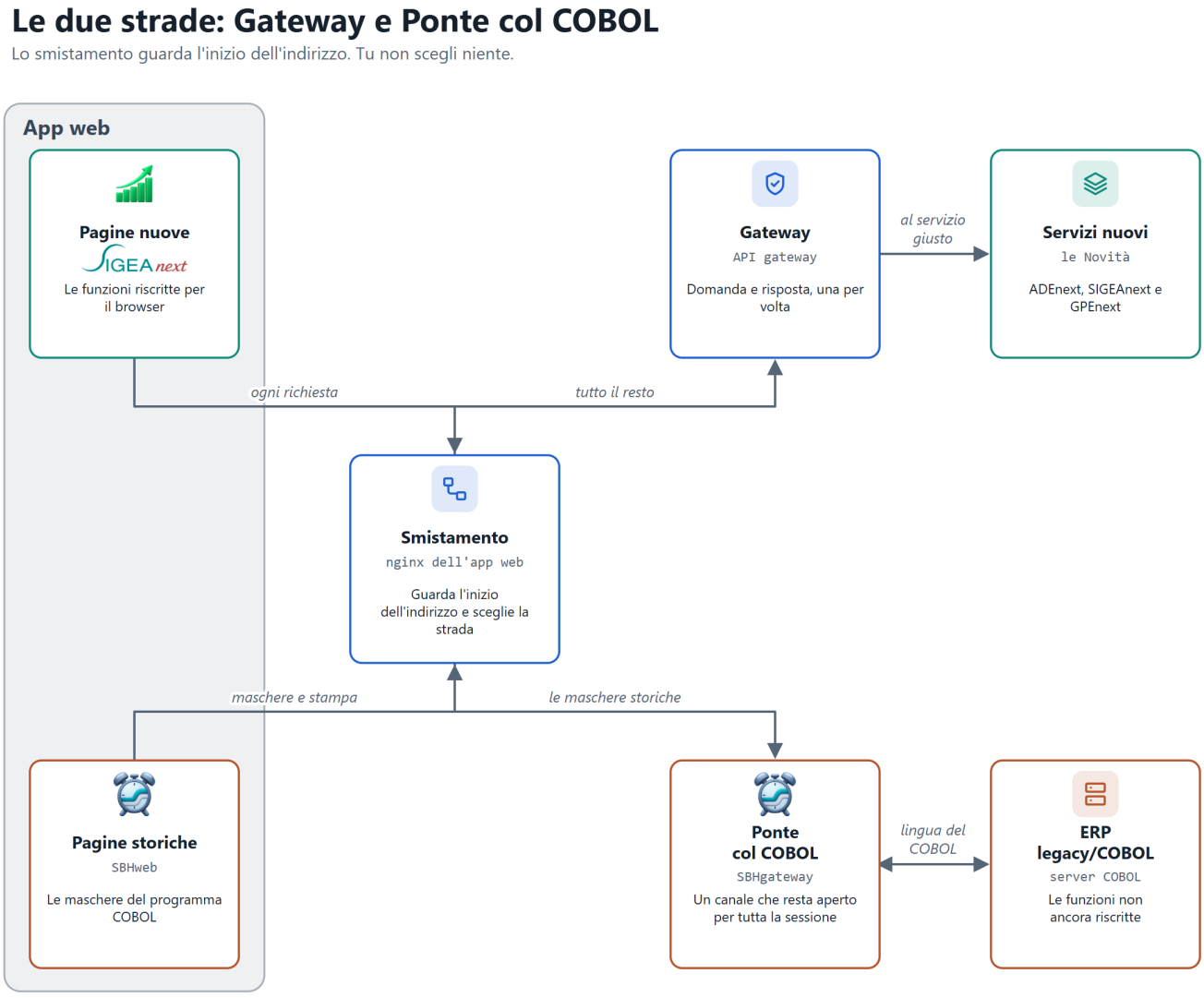
> **Per chi sviluppa.**
> • Il controllo è in `SecurityConfig` e `SsoConfig` del Gateway: firma verificata con le chiavi pubbliche dell'Accesso (JWKS), poi scadenza, emittente e programma che ha chiesto il lasciapassare (`SSO_CLIENT_AMMESSI`).
> • `AuthenticationFilter` scrive `X-User-Id` con il nome utente del lasciapassare e toglie sempre quello arrivato dal client.
> • L'interruttore è `AUTH_ENABLED`. Spento serve solo in sviluppo: il Gateway lascia passare tutto e scrive un utente di comodo. Se è acceso ma mancano gli indirizzi dell'Accesso, il Gateway rifiuta ogni lasciapassare.
> • Anche i servizi verificano il lasciapassare per conto loro: una richiesta che non passa dal Gateway non entra.

> **Per approfondire.**
> • Integrazione Autenticazione JWT Angular + API Gateway: `docs/guide/sviluppo/GUIDA-AUTENTICAZIONE-JWT.md` (descrive una versione precedente, in cui si entrava dal Gateway: oggi si entra dall'Accesso)
> • Il disegno di oggi: `docs/devops/auth/AUTENTICAZIONE.md`

# JC2.3 - Le due strade: Gateway e Ponte col COBOL

**In una frase.** Le pagine nuove passano dal Gateway, le maschere storiche dal Ponte col COBOL: la strada la sceglie l'app, non tu.

L'app web è una sola, ma dentro ha due parti. Le **pagine nuove** parlano con i servizi nuovi: fanno una domanda, ricevono una risposta, fine. Le **pagine storiche** di SBHweb parlano con il programma COBOL, che ragiona in un altro modo: tiene aperta una conversazione per tutta la sessione, maschera dopo maschera. Sono due modi di lavorare diversi e servono due strade.




*Le due strade: lo smistamento manda le maschere storiche al Ponte e tutto il resto al Gateway*

**La strada del Gateway.** La prendono tutte le richieste delle pagine nuove. Ognuna viene controllata e consegnata al servizio giusto, come racconta Dal browser al servizio giusto.

**La strada del Ponte col COBOL.** La prendono le maschere storiche. Fra la pagina e il Ponte si apre un canale che resta aperto per ore, anche quando non scrivi niente. Il Ponte traduce fra il browser e il programma COBOL. Come funziona è in Come funziona il Ponte col COBOL (libro «JH - SBHweb», pagina «JH1.2 - Come funziona il Ponte col COBOL»).

**Chi sceglie la strada.** Il server web dell'app, guardando l'inizio dell'indirizzo. Le richieste delle maschere storiche vanno al Ponte. Tutte le altre vanno al Gateway.

**L'incrocio: la stampa in PDF.** Quando da una maschera storica scegli l'uscita `GPE_NEXT_PDF`, il programma COBOL prepara solo i dati. Arrivano alla pagina dal Ponte. Poi la pagina li manda a GPEnext, e lì prende la strada del Gateway, con il tuo lasciapassare come ogni altra richiesta. È il motivo per cui le stampe nuove e quelle storiche usano gli stessi modelli.

> **Cosa ci guadagni.**
> • Le funzioni storiche si usano nel browser senza aspettare che siano riscritte.
> • Quando una funzione viene riscritta cambia strada da sola: tu apri la stessa app.
> • Le due strade non dipendono l'una dall'altra: le maschere storiche non passano dal Gateway, le pagine nuove non passano dal Ponte.

> **Per chi sviluppa.**
> • Lo smistamento è in `frontend/docker/nginx.conf`: `/api/sbh/` va al Ponte (`backend/middleware/it_gsba_sbh_gateway`, porta 5086) con un WebSocket; il resto di `/api/` va al Gateway.
> • Il Ponte non ha una rotta nel Gateway e non è iscritto al Monitor servizi.
> • La stampa da SBHweb chiama il Gateway su `/api/gpe/print/`.
> • Le chiamate interne al Ponte (`/api/sbh/interno/`) dal browser non esistono: il server web risponde 404.

# JC3 - Monitor servizi

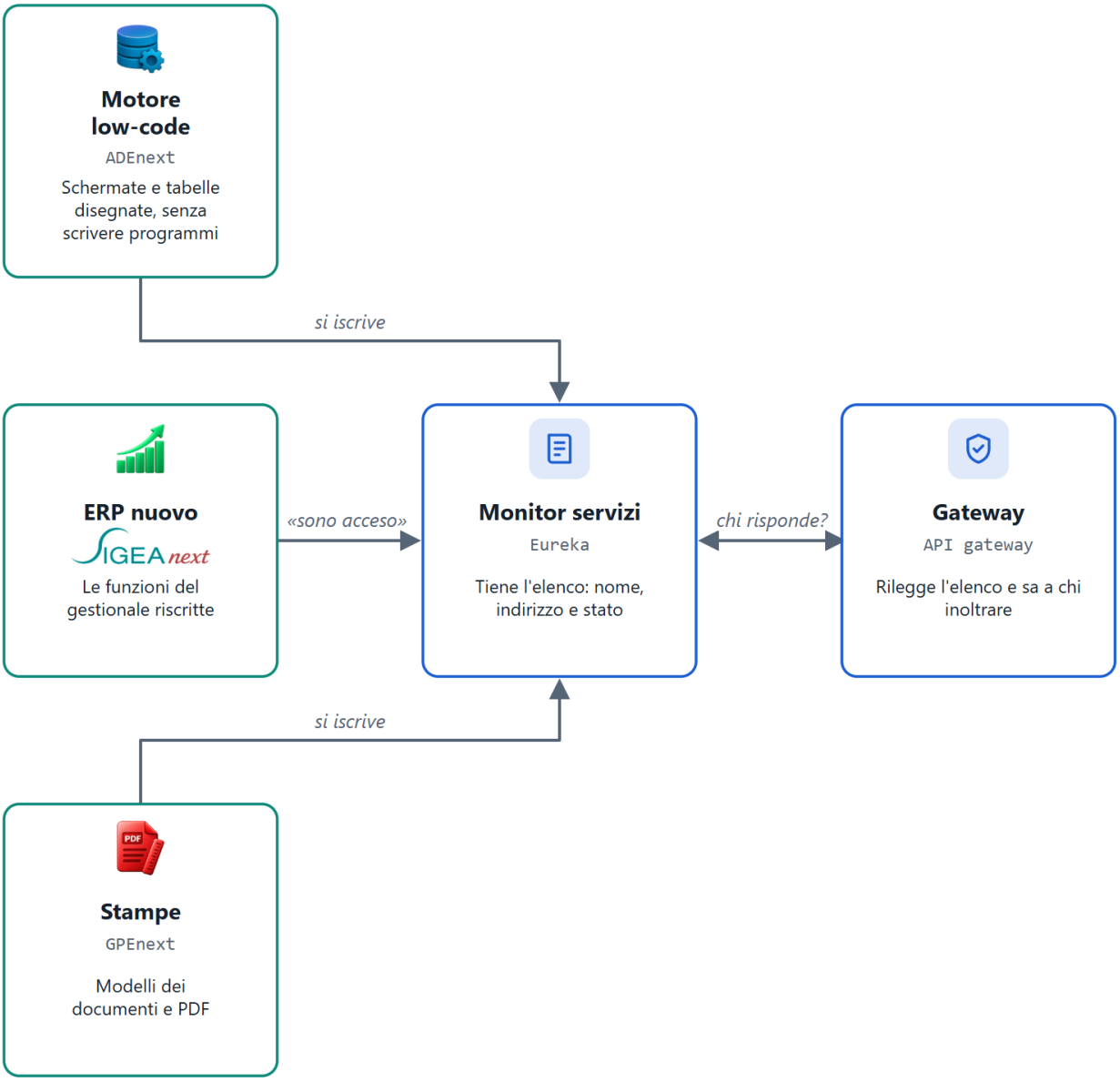
# JC3.1 - Vedere quali servizi sono accesi

**In una frase.** Il Monitor servizi ha una pagina con l'elenco dei servizi iscritti: chi gestisce il server la apre per capire, in un colpo d'occhio, chi è acceso.

Quando una pagina non si apre, la prima domanda è sempre la stessa: quale servizio non risponde? Senza un elenco bisognerebbe provarli uno per uno. Il Monitor servizi l'elenco ce l'ha già, e non lo scrive nessuno a mano: ogni servizio, quando si accende, si iscrive da solo con il suo nome e il suo indirizzo. Poi manda un segnale a intervalli regolari per dire «ci sono ancora».

## Chi si accende si iscrive

Nessuno scrive a mano dove sta un servizio. Il Gateway consulta l'elenco.




*Ogni servizio si iscrive quando si accende; il Gateway consulta l'elenco per sapere a chi inoltrare*

1. **Collegati al server.** La pagina del Monitor servizi si apre solo dal server stesso, non dal tuo PC. È una scelta di sicurezza: il Monitor servizi non ha password.
2. **Apri la pagina.** Sul server, in un browser, vai all'indirizzo `http://localhost:8761`.
3. **Leggi l'elenco dei servizi iscritti.** C'è una riga per servizio, con il nome e lo stato. Con tutto acceso trovi il Gateway (`API-GATEWAY`), ADEnext (`ADE-NEXT-SERVICE`), SIGEAnext (`SIGEA-NEXT-SERVICE`) e GPEnext (`GPE-NEXT-SERVICE`). Lo stato `UP` vuol dire che il servizio risponde.
4. **Cerca chi manca.** Un servizio che non compare non si è iscritto: è spento, sta ancora partendo o non riesce a raggiungere il Monitor servizi.

**Nota.** L'elenco non è istantaneo. Un servizio appena acceso compare dopo qualche decina di secondi. Uno caduto all'improvviso può restare scritto ancora per un po', perché il Monitor servizi aspetta prima di cancellarlo: un attimo di rete lenta non deve far sparire un servizio che sta bene.

**Attenzione.** Nell'elenco non trovi il Ponte col COBOL, l'Assistente dei dati e l'Accesso: non si iscrivono al Monitor servizi. Che manchino è normale.

**Per chi sviluppa.**
• Il Monitor servizi è `backend/middleware/it_gsba_eureka_server`; la pagina è il cruscotto standard di Eureka, sulla porta 8761.
• Nell'installazione la porta è pubblicata solo su `127.0.0.1` (`PORTE_INTERNE_BIND`, vedi `devops/deploy/README.md`).
• Ogni servizio ha anche il suo controllo di stato: `/actuator/health`. Quello del Gateway risponde senza accesso.
• L'attesa prima di cancellare un servizio muto si regola con `EUREKA_SERVER_ENABLE_SELF_PRESERVATION` e `EUREKA_SERVER_EVICTION_INTERVAL_TIMER_IN_MS`.

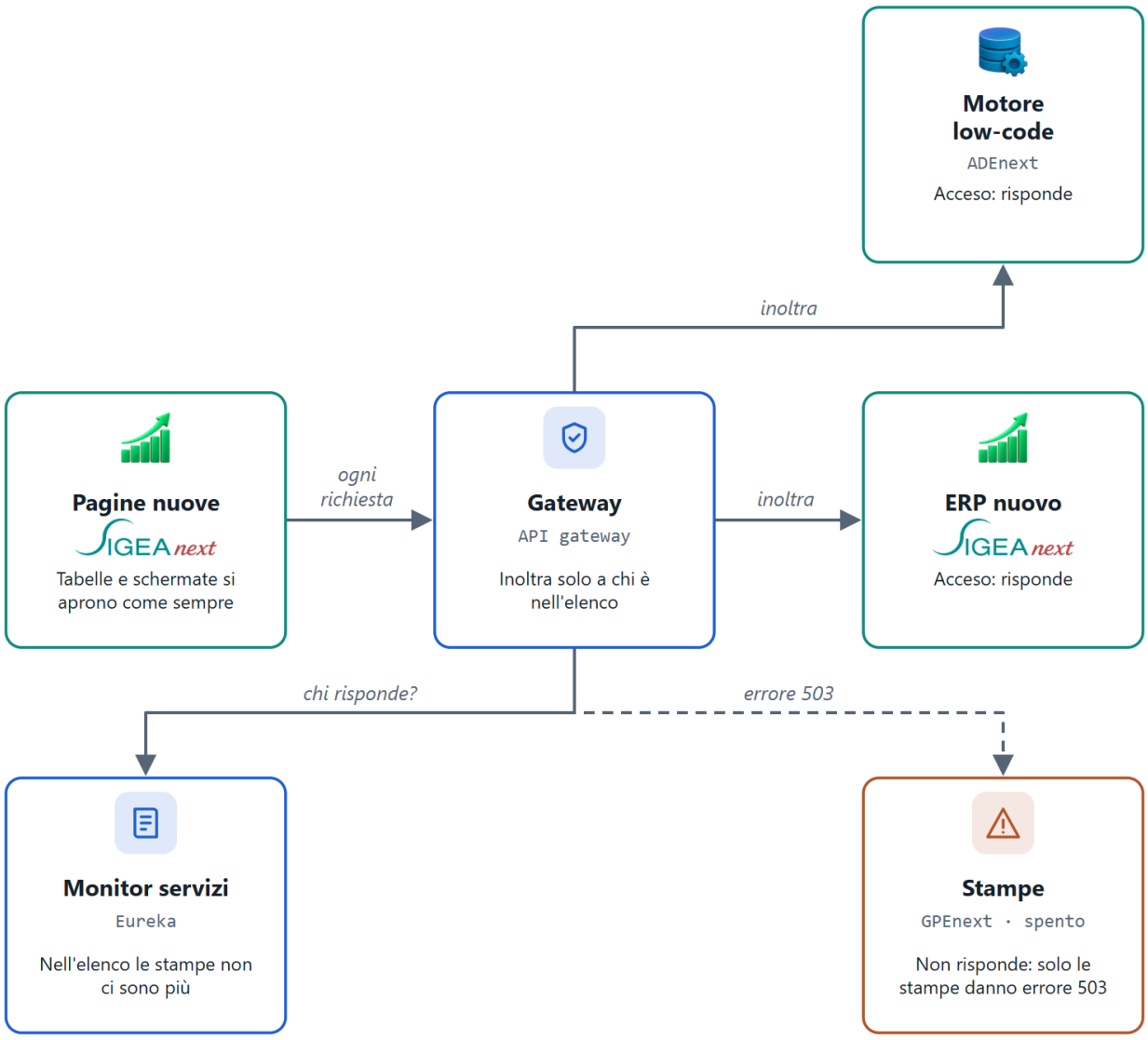
# JC3.2 - Cosa succede se un servizio è spento

**In una frase.** Si ferma solo quello che quel servizio faceva: il resto del gestionale continua a lavorare, e quando il servizio torna riparte da solo.

In un programma fatto di un pezzo solo, un guasto ferma tutto. Qui ogni servizio ha il suo compito, e il Gateway inoltra una richiesta solo al servizio che la riguarda. Se quel servizio non c'è, la richiesta torna indietro con un errore. Le altre non se ne accorgono.

## Un servizio spento non ferma gli altri

Esempio: le stampe sono ferme. Le tabelle e le schermate continuano a lavorare.




*Le stampe sono spente: le richieste di stampa tornano con l'errore 503, le altre arrivano a destinazione*

Un esempio. GPEnext, il servizio delle stampe, è fermo.

- Apri una tabella: funziona. La richiesta va a SIGEAnext, che è acceso.
- Apri una schermata disegnata: funziona. La richiesta va a ADEnext.
- Chiedi un PDF: non funziona. Il Gateway cerca GPEnext nell'elenco del Monitor servizi, non lo trova e risponde con l'errore 503, che vuol dire «servizio non disponibile». I pulsanti della stampa diventano grigi e dicono perché.

Quando GPEnext riparte non c'è niente da riattivare. Si iscrive di nuovo al Monitor servizi, il Gateway rilegge l'elenco e le stampe tornano. Di solito basta un minuto da quando il servizio è pronto.

Per questo, subito dopo un aggiornamento, può capitare di vedere un errore 503: i servizi stanno partendo e non sono ancora tutti nell'elenco. Aspetta un minuto e riprova.

| Se è spento | Cosa non va | Cosa continua |
|---|---|---|
| GPEnext | Stampe in PDF e designer dei modelli | Tabelle, schermate, analisi |
| ADEnext | Schermate disegnate e cruscotti salvati | Maschere storiche di SBHweb |
| SIGEAnext | Dati del gestionale nelle pagine nuove | Maschere storiche di SBHweb |
| Gateway | Tutte le pagine nuove | Le maschere storiche, che fanno l'altra strada |

**Attenzione.** Il Gateway e il Monitor servizi sono i due pezzi da cui dipendono tutte le pagine nuove. Se si ferma il Gateway, nessuna richiesta arriva ai servizi. Se si ferma il Monitor servizi, il Gateway va avanti per un po' con l'ultimo elenco che ha letto, ma i servizi che ripartono non riescono più a farsi trovare.

**Per chi sviluppa.**
• Il 503 lo dà il Gateway quando per `lb://NOME-DEL-SERVIZIO` non c'è nessuna istanza iscritta.
• Nell'installazione i servizi partono dopo che il Monitor servizi ha risposto al controllo di stato (`depends_on` in `devops/deploy/docker-compose.deploy.yml`).
• Se è il Gateway a essere fermo o in riavvio, il server web dell'app risponde 502: ritrova il Gateway da solo appena torna.

# JC4 - Per chi sviluppa

# JC4.1 - Aggiungere una rotta

**In una frase.** Per far arrivare le richieste a un servizio nuovo basta una rotta nel Gateway: gli altri servizi non si toccano.

Una **rotta** è una riga che dice: le richieste che cominciano così vanno a quel servizio. Stanno tutte in una classe sola, quindi chi vuole sapere cosa è raggiungibile da fuori guarda in un posto solo. Aggiungerne una vuol dire copiare quelle che ci sono.

1. **Dai un nome al servizio.** È il nome con cui si iscrive al Monitor servizi: tutto maiuscolo, con i trattini, per esempio `NUOVO-SERVICE`. Sta in `spring.application.name` del servizio.
2. **Aggiungi la rotta nel Gateway.** Apri `GatewayConfig` e copia una rotta esistente. Cambiano tre cose: il nome della rotta (`nuovo-service`), l'inizio dell'indirizzo (`/api/nuovo/**`) e la destinazione (`lb://NUOVO-SERVICE`). Restano uguali `stripPrefix(2)` e il filtro `authenticationFilter`: vanno messi sempre.
3. **Nel servizio non ripetere l'inizio dell'indirizzo.** Il Gateway toglie `/api/nuovo` prima di inoltrare. Se la pagina chiama `/api/nuovo/clienti`, il servizio riceve `/clienti`.
4. **Fai controllare il lasciapassare anche al servizio.** Prendi come modello la `SecurityConfig` di un servizio esistente. Il nome utente lo leggi dall'intestazione `X-User-Id`, che scrive il Gateway.
5. **Ricostruisci e riavvia solo il Gateway.** I servizi già accesi non vanno fermati.
6. **Prova.** Chiama `/api/nuovo/...` dall'app. Un 503 vuol dire che il servizio non è ancora iscritto al Monitor servizi. Un 404 vuol dire che la rotta c'è ma il servizio non ha quell'indirizzo.
7. **Aggiorna la tabella delle rotte** in `docs/architettura/riferimento/ARCHITETTURA-MICROSERVIZI.md`.

Il server web dell'app non va toccato: manda già al Gateway tutto ciò che comincia per `/api/`. Serve una regola in più solo per i collegamenti che restano aperti a lungo.

> **Attenzione.**
> • Con un numero diverso da 2 in `stripPrefix` il servizio riceve indirizzi sbagliati.
> • Senza `authenticationFilter` il servizio non riceve il nome utente, e chi chiama potrebbe scriverne uno a piacere.
> • Un servizio raggiungibile senza passare dal Gateway non può fidarsi di `X-User-Id`.

> **Per chi sviluppa.**
> • File: `backend/middleware/it_gsba_api_gateway/src/main/java/it/gsba/api_gateway/config/GatewayConfig.java`.

• Verifica: nella cartella del Gateway `./mvnw -o clean package -DskipTests`, vale solo `BUILD SUCCESS`.
• Anche in sviluppo l'app manda già al Gateway tutto ciò che comincia per `/api`.
• Prima di cambiare le regole di sicurezza del Gateway si dice cosa cambia e perché: una modifica piccola può spegnere i controlli.

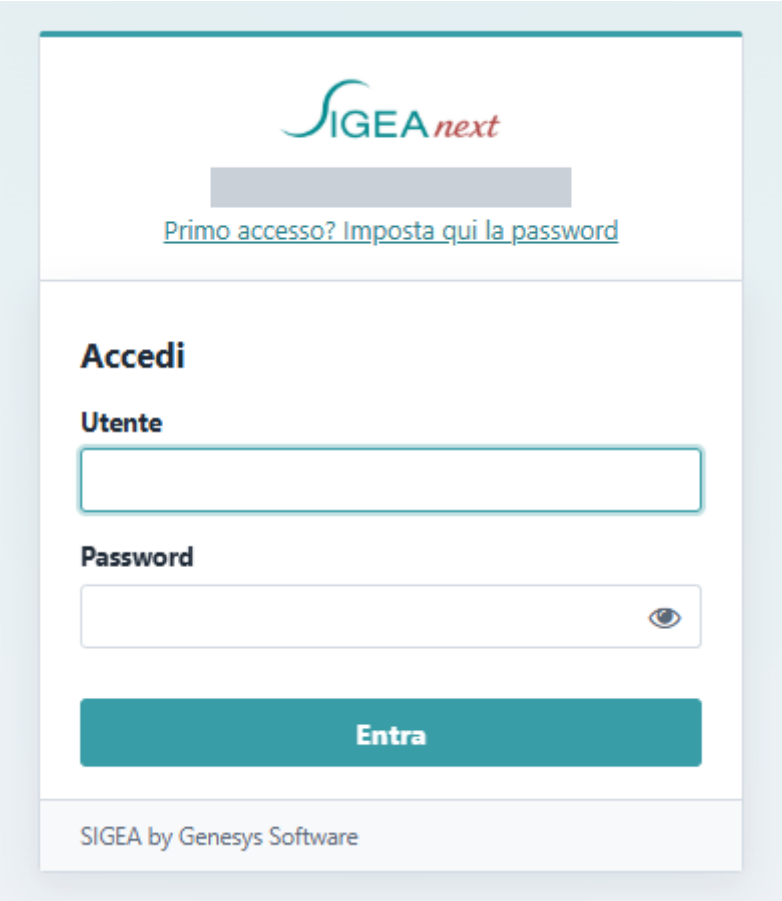
# JC4.2 - Errori 401, 403 e 503: cosa vogliono dire

**In una frase.** Tre numeri dicono dove si è fermata una richiesta: all'ingresso (401), sul permesso (403) o davanti a un servizio spento (503).

Un errore con un numero sembra un guasto. Quasi sempre è una risposta precisa: dice chi ha fermato la richiesta e perché. Saper leggere questi tre numeri evita di chiamare l'assistenza quando basta rientrare o aspettare un minuto.

| Numero | Vuol dire | Cosa fai |
|---|---|---|
| **401** | Il Gateway non sa chi sei: il lasciapassare manca o è scaduto | Rientra con utente e password |
| **403** | Sei entrato, ma il tuo livello non basta per quella cosa | Chiedi il livello a chi amministra |
| **503** | Il servizio che deve rispondere non è acceso o non è ancora pronto | Aspetta un minuto e riprova |

**401: rientra.** L'app ti riporta da sola alla pagina di accesso. Scrivi utente e password e riprendi.




*La pagina di accesso: è qui che l'app ti riporta dopo un errore 401*

**403: è una questione di livello.** Compare l'avviso **Accesso negato**. Non è un guasto e riprovare non serve: quella operazione chiede un livello più alto del tuo. Chi amministra può dartelo dalla pagina dei ruoli. I livelli sono in Cosa può fare ogni livello (libro «JB - Accesso e permessi», pagina «JB3.2 - Cosa può fare ogni livello»).

**503: aspetta un minuto.** Compare l'avviso **Errore server 503**. Il Gateway funziona, ma il servizio a cui è destinata la richiesta non è nell'elenco del Monitor servizi. Subito dopo un aggiornamento è normale: i servizi stanno ripartendo e si iscrivono uno alla volta. Se dopo qualche minuto l'errore c'è ancora, avvisa chi gestisce il server e di' quale pagina stavi aprendo: basta per capire quale servizio è fermo.

> **Nota.** Il 401 e il 503 li decide il Gateway. Il 403 lo decide il servizio che riceve la richiesta, perché è lui che sa quale livello serve per ogni operazione.

> **Per chi sviluppa.**
> • Gli avvisi li mostra `frontend/shell/src/shared/auth/error.interceptor.ts`: 401 porta a `/auth/login`, 403 dà «Accesso negato», da 500 in su «Errore server» con il numero.
> • 401 con un lasciapassare che sembra buono: controlla emittente, chiavi e `SSO_CLIENT_AMMESSI` (`docs/devops/auth/AUTENTICAZIONE.md`).
> • 401 fra due servizi: la chiamata non porta il lasciapassare. Le regole sono in

`docs/devops/chiamate-fra-servizi.md`.

- 502 al posto di 503: è fermo il Gateway, non un servizio.

# JC4.3 - Se qualcosa non va

**In una frase.** I problemi più comuni con Gateway e Monitor servizi, come «vedo questo, faccio questo», e le parole da sapere.

Quando una pagina non risponde, la domanda utile è: dove si è fermata la richiesta? Il numero dell'errore lo dice quasi sempre. Qui trovi cosa fare caso per caso.

## Per chi usa il gestionale

- **Mi ritrovo alla pagina di accesso mentre lavoravo.** Il lasciapassare non vale più. Rientra: è l'errore 401.
- **Vedo «Accesso negato».** Il tuo livello non basta per quella operazione. Chiedi a chi amministra: è l'errore 403.
- **Vedo «Errore server 503» subito dopo un aggiornamento.** I servizi stanno ripartendo. Aspetta un minuto e riprova.
- **Vedo «Errore server 503» su una cosa sola, per esempio le stampe.** È fermo quel servizio. Il resto puoi usarlo. Avvisa chi gestisce il server.
- **Vedo «Server non raggiungibile» o «Errore server 502».** Non risponde il Gateway, oppure è caduta la rete. Controlla la connessione, poi avvisa chi gestisce il server.
- **Le pagine nuove non vanno, le maschere storiche sì.** Può succedere: fanno due strade diverse. Il problema è dalla parte del Gateway.

I tre numeri sono spiegati in Errori 401, 403 e 503: cosa vogliono dire.

## Per chi gestisce il server

- **Un servizio dà 503 e non passa.** Apri la pagina del Monitor servizi e guarda se è nell'elenco: vedi Vedere quali servizi sono accesi. Se manca, guarda il registro di quel servizio: non è partito o non raggiunge il Monitor servizi.
- **Tutti i servizi danno 503.** Controlla che il Monitor servizi sia acceso. Senza di lui i servizi non riescono a iscriversi.
- **Tutti ricevono 401, anche appena entrati.** Il Gateway non riesce a verificare i lasciapassare: controlla gli indirizzi dell'Accesso nella configurazione. Se mancano, li rifiuta tutti.
- **Dal mio PC la pagina del Monitor servizi non si apre.** È voluto: si apre solo dal server.

- **Un servizio spento compare ancora nell'elenco.** Il Monitor servizi aspetta prima di cancellarlo.

# Parole da sapere

- **Gateway.** La porta unica: controlla chi sei e manda ogni richiesta al servizio giusto.
- **Monitor servizi.** L'elenco dei servizi accesi, con nome e indirizzo.
- **Servizio.** Un programma che fa una cosa sola: le stampe, i dati del gestionale, le schermate.
- **Richiesta.** Una domanda che la pagina fa a un servizio: «dammi questa tabella», «salva questo».
- **Rotta.** La regola del Gateway che lega l'inizio di un indirizzo a un servizio.
- **Lasciapassare.** La prova che sei entrato: lo firma l'Accesso, dice chi sei e che livello hai, scade dopo poco.
- **Livello.** Quello che puoi fare: lettore, base, avanzato, staff.
- **Iscriversi.** Quello che fa un servizio quando si accende: dice al Monitor servizi «sono qui».
- **Ponte col COBOL.** L'altra strada: porta le maschere storiche nel browser, senza passare dal Gateway.

> **Per chi sviluppa.** Lo stato si controlla con `/actuator/health` su Gateway e servizi. Variabili e porte: `docs/devops/ENVIRONMENT.md` e `devops/deploy/README.md`.