

Great—here are direct links and minimal payloads for everything you can click/test right now, grouped by purpose. Use [http://localhost/rail\\_app](http://localhost/rail_app) as the base (WAMP default for this folder). If your vhost differs, replace the base accordingly.

## Client UI (upload/OCR flow and wizard)

### Upload (OCR/fixtures entry)

- GET: [http://localhost/rail\\_app/upload](http://localhost/rail_app/upload)  
Lets you upload a ticket file or paste a Journey JSON.
- POST: [http://localhost/rail\\_app/upload/analyze](http://localhost/rail_app/upload/analyze)  
Triggered by the form above; computes Art. 12/9/18/19 and unified claim.  
With USE\_LIVE\_APIS=true it will also fetch live delay (DB/SNCF) and show "Live forsinkelse..." at the top.

### Client Wizard (step-by-step claim flow)

- Start: [http://localhost/rail\\_app/wizard](http://localhost/rail_app/wizard)
- Questions: [http://localhost/rail\\_app/wizard/questions](http://localhost/rail_app/wizard/questions)
- Expenses: [http://localhost/rail\\_app/wizard/expenses](http://localhost/rail_app/wizard/expenses)
- Summary: [http://localhost/rail\\_app/wizard/summary](http://localhost/rail_app/wizard/summary)

### Claims (simple start/compute view)

- Start: [http://localhost/rail\\_app/claims](http://localhost/rail_app/claims)
- Compute (POST-only): [http://localhost/rail\\_app/claims/compute](http://localhost/rail_app/claims/compute)

### Reimbursement demo

- Start: [http://localhost/rail\\_app/reimbursement](http://localhost/rail_app/reimbursement)
- Generate: [http://localhost/rail\\_app/reimbursement/generate](http://localhost/rail_app/reimbursement/generate)
- Official: [http://localhost/rail\\_app/reimbursement/official](http://localhost/rail_app/reimbursement/official)

### Admin (no auth in this demo)

- Claims list: [http://localhost/rail\\_app/admin/claims](http://localhost/rail_app/admin/claims)
- View one: [http://localhost/rail\\_app/admin/claims/view/1](http://localhost/rail_app/admin/claims/view/1)
- Update status (POST-only): [http://localhost/rail\\_app/admin/claims/update-status/1](http://localhost/rail_app/admin/claims/update-status/1)
- Mark paid (POST-only): [http://localhost/rail\\_app/admin/claims/mark-paid/1](http://localhost/rail_app/admin/claims/mark-paid/1)

## Demo/mocks and scenarios (ready to click)

### Fixtures listing (demo)

- [http://localhost/rail\\_app/api/demo/fixtures](http://localhost/rail_app/api/demo/fixtures)
- [http://localhost/rail\\_app/api/demo/exemption-fixtures](http://localhost/rail_app/api/demo/exemption-fixtures)
- [http://localhost/rail\\_app/api/demo/art12-fixtures](http://localhost/rail_app/api/demo/art12-fixtures)

### Scenarios

- List: [http://localhost/rail\\_app/api/demo/scenarios](http://localhost/rail_app/api/demo/scenarios)
- Run (GET): [http://localhost/rail\\_app/api/demo/run-scenarios](http://localhost/rail_app/api/demo/run-scenarios)

### Analyze generated mock tickets (PDF/PNG/TXT under mocks/tests/fixtures)

- [http://localhost/rail\\_app/api/demo/mock-tickets](http://localhost/rail_app/api/demo/mock-tickets)
- With RNE enrichment: [http://localhost/rail\\_app/api/demo/mock-tickets?withRne=1](http://localhost/rail_app/api/demo/mock-tickets?withRne=1)
- Custom directory: [http://localhost/rail\\_app/api/demo/mock-tickets?baseDir=C%5Cwamp64%5Cwww%5Crail\\_app%5Cmocks%5Ctests%5Cfixtures](http://localhost/rail_app/api/demo/mock-tickets?baseDir=C%5Cwamp64%5Cwww%5Crail_app%5Cmocks%5Ctests%5Cfixtures)

## OCR/ingest API stub (for pipeline tests)

Ticket ingest (stub; returns a structure you can expand later)

- POST: [http://localhost/rail\\_app/api/ingest/ticket](http://localhost/rail_app/api/ingest/ticket)

Body (JSON or form) optional; returns { journey: {...}, logs: [...] }

Tip: Use this until your OCR call is wired. After you plug in OCR, map the extracted fields into the Journey format used by Upload/Compute.

## Compute APIs (Art 12/9/18/refund/claim/compensation)

All are POST endpoints. They accept JSON (application/json) or form data; below are the URLs and minimal JSON shapes.

### Compensation (EU Art. 19 rules)

- POST: [http://localhost/rail\\_app/api/compute/compensation](http://localhost/rail_app/api/compute/compensation)

JSON: { "journey": { "segments": [ { "schedArr": "2025-10-11T19:00:00", "actArr": "2025-10-11T20:15:00" } ], "ticketPrice": { "value": "49.90 EUR" }, "country": { "value": "DE" }, "operatorName": { "value": "DB" }, "trainCategory": { "value": "ICE" } }, "euOnly": true, "refundAlready": false, "knownDelayBeforePurchase": false, "extraordinary": false, "selfInflicted": false, "throughTicket": true, "minPayout": 4.0 }

### Exemptions profile (matrix incl. Art. 12 applicability)

- POST: [http://localhost/rail\\_app/api/compute/exemptions](http://localhost/rail_app/api/compute/exemptions)

JSON: { "journey": { "segments": [ { "country": "FR" } ], "is\_long Domestic": false, "is\_international\_inside\_eu": true, "is\_international\_beyond\_eu": false } }

### Art. 12 evaluator

- POST: [http://localhost/rail\\_app/api/compute/art12](http://localhost/rail_app/api/compute/art12)

JSON: { "journey": { "segments": [ { "operator": "DB", "country": "DE" } ], "is\_international\_inside\_eu": true }, "meta": { "through\_ticket\_disclosure": "yes", "contract\_type": "single" } }

### Art. 9 evaluator (information)

- POST: [http://localhost/rail\\_app/api/compute/art9](http://localhost/rail_app/api/compute/art9)

JSON: { "journey": { "segments": [ { "country": "DE" } ], "meta": { "info\_on\_rights": "Delvist" } } }

### Refund (Art. 16-like)

- POST: [http://localhost/rail\\_app/api/compute/refund](http://localhost/rail_app/api/compute/refund)

JSON: { "journey": { "segments": [ { "from": "München", "to": "Berlin" } ], "meta": { "refundAlready": false } } }

### Refusion (rerouting, Art. 18)

- POST: [http://localhost/rail\\_app/api/compute/refusion](http://localhost/rail_app/api/compute/refusion)

JSON: { "journey": { "segments": [ { "from": "Paris", "to": "Lyon" } ], "meta": { "claim\_rerouting": true, "reroute\_info\_within\_100min": "Nej" } } }

### Unified claim calculation (compensation + expenses + 25% fee)

- POST: [http://localhost/rail\\_app/api/compute/claim](http://localhost/rail_app/api/compute/claim)

JSON: { "country\_code": "DE", "currency": "EUR", "ticket\_price\_total": 49.9, "trip": { "through\_ticket": true, "legs": [ { "from": "München", "to": "Berlin", "scheduled\_dep": "2025-10-11T15:00:00", "scheduled\_arr": "2025-10-11T19:00:00", "actual\_arr": "2025-10-11T20:15:00" } ] }, "disruption": { "delay\_minutes\_final": 75, "notified\_before\_purchase": false, "extraordinary": false, "self\_inflicted": false }, "choices": { "wants\_refund": false, "wants\_reroute\_same\_soonest": false, "wants\_reroute\_nearest": false } }

```
"wants_reroute_later_choice": false }, "expenses": { "meals": 0, "hotel": 0, "alt_transport": 0, "other": 0 }, "already_refunded": 0 }
```

## Provider stubs (DB/SNCF/DSB/RNE/open)

These are basic placeholders you can hit to simulate external providers.

SNCF:

- [http://localhost/rail\\_app/api/providers/sncf/booking/validate](http://localhost/rail_app/api/providers/sncf/booking/validate)
- [http://localhost/rail\\_app/api/providers/sncf/trains](http://localhost/rail_app/api/providers/sncf/trains)
- [http://localhost/rail\\_app/api/providers/sncf/realtime](http://localhost/rail_app/api/providers/sncf/realtime)

Deutsche Bahn:

- [http://localhost/rail\\_app/api/providers/db/lookup](http://localhost/rail_app/api/providers/db/lookup)
- [http://localhost/rail\\_app/api/providers/db/trip](http://localhost/rail_app/api/providers/db/trip)
- [http://localhost/rail\\_app/api/providers/db/realtime](http://localhost/rail_app/api/providers/db/realtime)

DSB:

- [http://localhost/rail\\_app/api/providers/dsb/trip](http://localhost/rail_app/api/providers/dsb/trip)
- [http://localhost/rail\\_app/api/providers/dsb/realtime](http://localhost/rail_app/api/providers/dsb/realtime)

RNE:

- [http://localhost/rail\\_app/api/providers/rne/realtime](http://localhost/rail_app/api/providers/rne/realtime)

Open (generic):

- [http://localhost/rail\\_app/api/providers/open/rt](http://localhost/rail_app/api/providers/open/rt)

## Live train API toggle and testing (DB/SNCF)

Toggle live data:

Set environment variable USE\_LIVE\_APIS=true for your web server/PHP process.

Optional vars:

- DB\_TRANSPORT\_REST\_BASE=https://v6.db.transport.rest
- SNCF\_NAVITIA\_BASE=https://api.sncf.com
- SNCF\_API\_KEY=your\_key
- RNE\_BASE\_URL=http://localhost:5555/api/providers/rne

- A sample file exists: (environment variables—not auto-loaded; set them in your environment/Apache service).

Where live delay is used now:

The Upload flow (/upload → analyze) calls live APIs (when USE\_LIVE\_APIS=true).

You'll see a green line "Live forsinkelse fra API: N min" on the result page when live delay was applied.

Minimal Journey JSON for DB live test: { "segments": [ { "operator": "DB", "trainNo": "ICE 123", "from": "München Hbf", "to": "Berlin Hbf", "schedDep": "2025-10-11T15:00:00", "schedArr": "2025-10-11T19:00:00" } ], "depDate": { "value": "2025-10-11" } }

Paste that into the "Journey JSON" box on /upload with USE\_LIVE\_APIS=true to see the live delay override.

## Optional PowerShell one-liners (copyable)

```
• POST JSON to a compute endpoint: $body = @{ journey = @{ segments = @(@{ schedArr = "2025-10-11T19:00:00"; actArr = "2025-10-11T20:15:00" }) ticketPrice = @{ value = "49.90 EUR" } country = @{ value = "DE" } operatorName = @{ value = "DB" } trainCategory = @{ value = "ICE" } } euOnly =
```

```
$true } | ConvertTo-Json -Depth 6 Invoke-RestMethod -Method POST -Uri  
"http://localhost/rail\_app/api/compute/compensation" -ContentType "application/json" -Body $body  
•Upload a file to /upload/analyze: Invoke-WebRequest -Method POST -Uri  
http://localhost/rail\_app/upload/analyze -InFile "C:\path\to\ticket.pdf" -ContentType "application/pdf"
```

#### Notes:

JSON vs form: All compute endpoints accept JSON bodies. The upload/analyze action is typically form/multipart from the page. Use the UI for easiest testing.

- If a link 404s, confirm your base URL. On WAMP it's usually [http://localhost/rail\\_app](http://localhost/rail_app).

If you want the wizard (/wizard) to use live delay the same way as Upload does, I can wire TrainDataService there too and add a small "live data" hint on its summary—say the word and I'll add it.



Perfekt — her får du en **professionel introtekst**, der rammer **juridisk troværdighed, digital automatisering og formålet med portalen**, så Copilot (eller andre AI-assistenter) forstår præcist hele konteksten og kravene.

Jeg har skrevet den i et format, der både kan stå som **projektbeskrivelse i README.md** (øverst i repo) **og som intro i Copilot-prompten**.

Den samler alle de centrale dele, vi har bygget: billet-upload, automatisk dataudtræk, RNE-berigelse, fuldmagt, ID-validering, beregning, refusion, kompenstation, assistance, og bevisførelse.

---

## INTRODUKTION – Juridisk Kompensationsportal (EU 2021/782)

Jeg er **jurist** med speciale i **EU-passagerrettigheder og erstatningsret**, og udvikler en **automatisk portal**, hvor **togpassagerer** kan få udbetalt **kompensation eller refusion inden for 24 timer** — i fuld overensstemmelse med **EU-forordning 2021/782** og **CIV-reglerne**.

Systemet skal:

-  **Automatisk indlæse billetten** (PDF, PNG, PKPass, screendump, osv.)
-  **Udføre OCR-scanning og barcode-aflæsning**, identificere operatør, rute, tognummer, pris, klasse, dato og PNR-kode
-  **Berige oplysningerne via åbne API'er** (RNE, SNCF, DSB, DB, ÖBB m.fl.) for at hente planlagt og faktisk køreplan, forsinkelser og aflysninger
-  **Automatisk anvende EU-reglerne** om refusion (art. 18), kompenstation (art. 19), assistance (art. 20) og CIV-regler om passagerens ansvar
-  **Udfylde EU's officielle kompensationsformular** (bilag II, del 1-10) med autofyldte data og markere felter, hvor oplysninger mangler
-  **Foreslå præcise mikro-prompts til brugeren** kun når systemet mangler data eller der er mismatch mellem billet og RNE-data
-  **Beregn automatisk kompensationsbeløb** (25 % / 50 %) og håndtér nationale minimumssatser, gennemgående billetter og EU/ikke-EU-segmenter

- **Indhente digital fuldmagt og identifikation (Step Fuldmagt & Identifikation)** via eID-løsninger som MitID, BankID, SPID, DigiD, etc., i henhold til **eIDAS-forordningen (EU 910/2014)**
  - **Sikre GDPR-overholdelse** med kryptering, dataminimering, audit-trail og automatisk sletning
  - **Generere komplet sagsdokumentation (Step Bevisførelse)** med billet, RNE-snapshot, operatørpåtegning, og brugerbilag
  - **Udbetale kompensation eller refusion** automatisk, minus administrations-/kommissionsgebyr (fx 25 %), så snart ID og fuldmagt er verificeret
- 

## TEKNISK WORKFLOW (autonomt & API-drevet)

### **[1] Upload billetten**

→ OCR + barcode parser udtrækker data.

### **[2] Berig med API-kald**

→ RNE (plan/real-tidsdata) + operatør-API (SNCF, DSB, DB, osv.).

### **[3] Autofyld EU-formularens felter (3.1–3.3)**

→ Felter med høj tillid (> 0.85) udfyldes automatisk; øvrige genererer mikro-prompts.

### **[4] Juridisk validering af kontrakt (Art. 12)**

→ Gennemgående billet vs. særskilte kontrakter + ansvarsfordeling.

### **[5] Refusion (Art. 18)**

→ Registrér aflysning, omlægning, eller refusion; vurder downgrade.

### **[6] Kompensation (Art. 19)**

→ Beregn automatisk 25 % / 50 %; håndtér EU-segmenter og force majeure.

### **[7] Assistance (Art. 20)**

→ Log tilbuddt mad, hotel, alternativ transport, og bilag.

### **[8] Fuldmagt + Identifikation**

→ Automatisk eID-login + digital signatur + hash-bevis (eIDAS).

### **[9] Bevisførelse (Bilag II + CIV)**

→ Saml alle bevisfiler og generér komplet PDF-sag.

### **[10] Udbetalning**

→ Automatisk udbetalning via godkendt betalingsgateway, når ID + fuldmagt er valideret.

---



## DEMO-MODE OG TESTDATA

Portalen indeholder realistiske **demo-cases** til test af autofyld og beregning:

Demo-case	Type	Forsinkelse	Forventet kompensation
ice_125m	Gennemgående EU	125 min	50 %
tgv_30m	National (FR)	30 min	Voucher (G30)

Demo-case	Type	Forsinkelse	Forventet kompensation
ter_missed_conn	Regional	Missed connection	Art. 12-ansvar
ic_no_rne	Data-mangler	—	Mikro-prompts aktiveres

---

## SYSTEMARITEKTUR (for Copilot / dev-forklaring)

### Frontend:

- React / TypeScript (Tailwind + hooks)
- Autofyldede felter + mikro-prompts + upload-komponent
- eIDAS-compliant signaturflow

### Backend:

- Node / Express (eller Cloud Functions)
- OCR + QR parser (Tesseract / Zxing)
- RNE + Operatør-API clients
- GDPR-compliant storage (EU/EØS-lokationer)
- PDF-filler (pdf-lib)
- Rule engine for art. 12–20

### Output:

- JSON + PDF + ZIP-bilag
- Automatisk audit-trail og hash-verificering



## COMPLIANCE & DATASIKKERHED

Område	Løsning
eID	eIDAS-notified systemer (MitID, BankID, SPID osv.)
GDPR	Samtykke, kryptering (AES-256), audit-log, 12 md. retention
POA-lagring	Hash + tidsstempel (SHA-256, ISO timestamp)
Udbetaling	Kun til verificeret konto / IBAN-navnematch
Force majeure	Automatisk doktrin-kontrol + manuel review-flag
Revision	Data i EU-datacenter + sporbarhed (Art. 5 GDPR)

---



## FORMÅL MED DENNE INTRO

Denne beskrivelse sikrer, at **Copilot / AI-assistenter** forstår projektets fulde kontekst:

- At du er **jurist**, ikke bare udvikler
- At portalen skal **gennemføre fuld sagsbehandling automatisk**
- At alt skal ske **inden for gældende EU-lovgivning (2021/782, CIV, GDPR, eIDAS)**

- At **demo-mode** viser hele kæden (upload → RNE-data → beregning → signatur → PDF → udbetaling) uden manuel indtastning
- 

Vil du have, at jeg tilpasser introen som en **README.md-fil klar til GitHub / Copilot** (med markdown-overskrifter og badges), så Copilot genkender det som projektkontekst og begynder at autogenerere funktioner direkte (f.eks. “generate OCR parser”, “fill EU claim PDF”)?

Her er en kort, skarp og **Copilot-venlig** forklaringstekst, du kan indsætte i *forklaring\_af\_flow\_chart\_v\_2.pdf* under afsnittet **“Step Rail Exemptions”**. Den er skrevet som developer-note med formål, input, algoritme, output, edge cases og pseudo-kode, så Copilot kan finpudse implementeringen.

---

## Step Rail Exemptions (Art. 2, 2021/782)

### Formål

At afgøre, **hvilke artikler i forordningen gælder/er fritaget** for den konkrete rejse (og delstrækninger), så resten af flowet (Art. 12, 18, 19, 20, 30) bruger korrekt retsgrundlag og viser de rigtige UI-beskeder og beregninger.

### Primære regler (kort):

- **Art. 2(6)(a):** Medlemsstater kan fritage **by-/forstads** og **regional** trafik (visse artikler).
  - **Art. 2(6)(b):** Fritagelse for **international** trafik, hvor “en betydelig del og  $\geq 1$  stop” ligger **uden for EU**.
  - **Art. 2(4) (legacy) + Art. 2(4) i 2021/782:** For **long-distance domestic** kan landene forlænge fritagelser for bl.a. **Art. 15, 17, 19, 20(2)(a)-(b), 30(2)** (op til **3. dec. 2029**).
  - **Art. 2(5):** Land kan fritage **Art. 10 (realtidsdata)** indtil **7. juni 2030** (teknisk umulighed).
  - **Art. 2(8):** Fritagelser for regional/urban gælder **ikke** visse artikler (notér især **Art. 5 (cykler)**, 11, 13, 14, 21, 22, 27, 28 – dvs. de skal som udgangspunkt stadig iagttagtes).
- 

### Input (AUTO fra tidlige steps)

- `journey_segments[]`: (fra billet-OCR/PNR + RNE) med:  
`country, operator, train_no, service_type_hint` (regional/IC/HS),  
`dep_station, arr_station, dep_time, arr_time`.

- **distance\_km + is Domestic + is International\_inside\_EU + is International\_beyond\_EU** (geokodede stationer).
- **ticket\_scope**: gennemgående/PNR-scope/særskilte (fra Step Billetinformationer + Art. 12-blokken).
- **seller\_type**: operator vs. agency (til Art. 12(5)-ansvar).
- **country\_exemptions**: **statisk matrix** med landes fritagelser (regional/long-domestic/international-beyond-EU + særregler).  
Eksempler i din matrix:
  - **Regional**: særlige fritagelser i AT, BG, FR, HU, LU, PL, PT, RO, SK; **FI** (pendlerdekret 166/2023 → undtag Art. 19, 20(2)(a)-(b) på udpegede pendlerstræk), **DE** (Art. 9(2), 20(2)(a), 30(1) sætning 1), **SE** (<150 km: undtag 8, 9(1), 17–20).
  - **Long-distance domestic**: fx **HU** (undtag 19 og 20(2)), **PT** (undtag 17, 19, 20(2)), m.fl.
  - **International beyond EU**: fx **PL** (undtag 12, 18(3)); **FI** (til/fra Rusland); **CZ** (afg./ank. i tredjeland undtaget, ekskl. CH); **BG** (bred undtagelsesramme i visse scenarier).
  - **Art. 10 undtagelser** indtil 2030: **AT, HR, HU, LV, PL, RO**.

## Output (bruges af resten af flowet)

```
{
  "exemption_profile": {
    "scope": "regional | long Domestic | intl_inside_eu | intl_beyond_eu",
    "articles": {
      "art12": true,
      "art18_3": true,
      "art19": true,
      "art20_2": true,
      "art30_2": true,
      "art10": true
    },
    "notes": [
      "SE<150km: art.19/20 undtaget",
      "FI commuter: art.19 + 20(2)(a)-(b) undtaget",
      "PL intl beyond EU: art.12 + 18(3) undtaget",
      "DE regional: 9(2), 20(2)(a), 30(1) s.1 undtaget",
      "Art.10 undtaget (AT, HR, HU, LV, PL, RO) til 2030"
    ]
  }
}
```

```

        ],
        "ui_banners": [
            "Realtime-data (Art.10) kan mangle – vi henter evt. ikke-live RNE.",
            "Kompensation (Art.19) gælder ikke på denne strækning.",
            "Assistance (Art.20(2)) delvist undtaget på denne strækning."
        ]
    }
}

```

- `articles.artXX = true` betyder **gælder** (ikke fritaget). Sæt til `false` hvis **fritaget**.
  - `ui_banners` bruges til at vise kontekstuelle advarsler/vejledning i de efterfølgende steps (Bilag II del II-data, Kompensation, Assistance osv.).
- 

## Beslutningslogik (prioriteret rækkefølge)

### 1. Klassificér tjenesten

- Hvis **by/forstad** → (vi udelader i app'en p.t.) → stop.
- Ellers, afgør **regional** vs. **long-distance domestic** vs. **international**.
  - `is_international_beyond_EU = true`, hvis **≥1 stop uden for EU** og "**betydelig del**" udenfor (konservativt: >25% af km eller tid; parametrer).

### 2. Slå **national fritagelse op** i `country_exemptions per segment`:

- **regional**: anvend landespecifikke undtagelser (SE<150 km, DE/FI specialer, m.v.).
- **long-domestic**: anvend Art. 2(4)-undtagelser (fx HU/PT) frem til evt. **3. dec. 2029**.
- **intl beyond EU**: anvend landespecifikke "tredjelands" undtagelser (PL, FI, CZ, BG m.fl.).

### 3. Art. 10 (realtidsdata)

Hvis land på ruten er i listen **AT, HR, HU, LV, PL, RO** → `articles.art10=false` (til 7. juni 2030) og vis fallback-banner:

"Art. 10 kan være undtaget. Vi henter ikke-live RNE og beder dig uploadere bekræftelse (Art. 20(4)) ved behov."

### 4. Art. 12 (gennemgående billetter)

Hvis undtaget (fx **PL intl beyond EU**) → `articles.art12=false` og **disable gennemgående ansvar**:

- Split krav **per billet**; "missed connection" på tværs af **særskilte kontrakter** er ikke dækket i EU-etten på den strækning.
- Vis banner i Art. 12-blokken.

### 5. Art. 18(3) 100-minuttersreglen

Hvis land/stræk er fritaget → `articles.art18_3=false` og vis banner:

“100-min-regel kan være undtaget — vi logger stadig dine udgifter og kører lokal praksis.”

#### 6. Art. 19 (kompensation) og Art. 20(2) (assistance)

Sæt `articles.art19 / articles.art20_2` til `false` på undtagne stræk (fx **FI commuter, HU/PT long-domestic, SE<150 km m.fl.**).

Konsekvens:

- Kompensationsmotoren skal **fallback til nationale ordninger** (hvis mere generøse) eller **ingen EU-kompensation** hvor undtaget.
- Assistance (måltider/hotel/transport) deaktiveres delvist/fuldt og UI beder om **udgiftsbilag + “lokal praksis”-tekst**.

#### 7. Art. 30(2) (rettighedsinfo)

Deaktiver hvor national fritagelse gælder (fx DE regional: 30(1) 1. sætning; *du har også 30(2)-cases under long-domestic undtagelser*).

Hvis undtaget → vis kun **app’ens egen** rettighedstekst og NEB-link, men markér at operatøren **ikke** er forpligtet på samme måde lokalt.

---

## Pseudo-kode (til Copilot)

```
type ExemptionProfile = {  
    scope: 'regional' | 'long_domestic' | 'intl_inside_eu' | 'intl_beyond_eu',  
    articles: {  
        art12: boolean,  
        art18_3: boolean,  
        art19: boolean,  
        art20_2: boolean,  
        art30_2: boolean,  
        art10: boolean  
    },  
    notes: string[],  
    ui_banners: string[]  
};
```

```
function buildExemptionProfile(journey: Journey, matrix: CountryExemptions): ExemptionProfile {  
    const prof = defaultAllTrueProfile(classifyScope(journey));  
  
    // 1) Per-segment national lookups  
    for (const seg of journey.segments) {  
        const cx = matrix.lookup(seg.country, prof.scope, seg);
```

```

    prof.articles.art19    &&= !cx.exempt_art19;
    prof.articles.art20_2  &&= !cx.exempt_art20_2;
    prof.articles.art18_3  &&= !cx.exempt_art18_3;
    prof.articles.art12   &&= !cx.exempt_art12;
    prof.articles.art30_2  &&= !cx.exempt_art30_2;
    // Art. 10 (realtime) until 2030
    if (cx.exempt_art10) prof.articles.art10 = false;

    prof.notes.push(...cx.notes);
}

// 2) Intl beyond EU (2(6)(b))
if (isIntlBeyondEU(journey)) {
    const intlX = matrix.lookupIntlBeyondEU(journey);
    applyIntlExemptions(prof, intlX);
}

// 3) UI banners
if (!prof.articles.art10) prof.ui_banners.push("Realtidsdata kan være
undtaget; fallback til ikke-live RNE + dokumentation.");
if (!prof.articles.art19) prof.ui_banners.push("EU-kompensation kan være
undtaget på dele af ruten.");
if (!prof.articles.art20_2) prof.ui_banners.push("Assistance
(måltider/hotel/transport) kan være undtaget; upload udgiftsbilag.");
if (!prof.articles.art12) prof.ui_banners.push("Gennemgående billet-regler
gælder ikke; krav splittes per billet.");
if (!prof.articles.art18_3) prof.ui_banners.push("100-minutters-reglen kan
være undtaget.");

return prof;
}

```

## Edge cases (håndtering)

- **Blandet rute** (nogle segmenter undtaget, andre ikke):
  - Gem **per-segment flags** + **aggregér** strengest relevante undtagelser i **exemption\_profile**.
  - Kompensationsmotoren skal **beregne per ansvarlig kontrakt/segment** (Art. 12 → gennemgående vs. særskilt).

- **Art. 10 undtaget:**
    - Brug ikke-live RNE + upload af Art. 20(4) delay-confirmation som bevis.
    - Slå Art. 19(9) (forhåndsoplyst forsinkelse) op i Bilag II del I pkt. 7 via billetflowet i stedet.
  - **Art. 12 undtaget** (fx PL intl-beyond-EU):
    - Disable “missed connection” på tværs af særskilte kontrakter.
    - Beregn og vis kompensation **per billet** (Art. 19(3) principper pr. delstræk).
  - **SE <150 km:**
    - Regional **undtagelse** (9(1), 17–20, m.fl.) → hoppes over.
    - Brug lokal/national ordning hvis definéret. Ellers informér om ingen EU-kompensation.
  - **FI “commuter decree”:**
    - På udpegede pendlerstrækninger: **Art. 19 + 20(2)(a)-(b)** undtaget.
    - Marker i **notes** og redirect til national praksis (hvis eksisterer).
- 

## Integration-hooks (så resten af flowet “forstår” undtagelser)

- `exemption_profile` injiceres i:
    - **Step Billetinformationer** (vis/skip spørgsmål om Bilag II del II, hvis Art. 10 er undtaget).
    - **Step Art. 12** (disable gennemgående check/ansvar, hvis undtaget).
    - **Step Kompensation (Art. 19)** (disable EU-komp og aktivér national override-tabel).
    - **Step Assistance (Art. 20)** (disable måltider/hotel/transport hvor undtaget – log udgifter i stedet).
    - **Step Bevisførelse** (tilføj `exemption_profile.notes` i sagsresumé).
- 

## Hvor skal du opdatere i Copilot-projektet (kort)

1. **/lib/exemptions/matrix.ts**
  - Landematrix + specialregler (SE<150km, FI commuter, DE regional, HU/PT long-domestic, intl-beyond-EU særregler).
  - Liste for **Art. 10 undtaget** (AT, HR, HU, LV, PL, RO) med udløb 7/6/2030.
2. **/lib/exemptions/classify.ts**
  - `classifyScope(journey) + isIntlBeyondEU()` (med param. for “betydelig del”).
3. **/lib/exemptions/buildProfile.ts**
  - Pseudo-koden ovenfor → returnér `ExemptionProfile`.
4. **/services/compensationEngine.ts**
  - Respekter `exemption_profile.articles` og **national overrides** før EU-fallback.

## 5. UI

- **Banner-komponent** der læser `exemption_profile.ui_banners`.
  - **Form-renderer** (vis/skip felter ift. Art. 10/12/18/19/20 fritagelser).
- 

Hvis du vil, kan jeg også levere en **minimal matrix.ts stub** med de lande, du allerede har listet, så du kan paste direkte ind og lade Copilot udfylde resten.

### Matrix Stub.ts – jf. ovenfor :

Her er en kort, skarp og **Copilot-venlig** forklaringstekst, du kan indsætte i *forklaring\_af\_flow\_chart\_v\_2.pdf* under afsnittet “**Step Rail Exemptions**”. Den er skrevet som developer-note med formål, input, algoritme, output, edge cases og pseudo-kode, så Copilot kan finpudse implementeringen.

---

## Step Rail Exemptions (Art. 2, 2021/782)

### Formål

At afgøre, **hvilke artikler i forordningen gælder/er fritaget** for den konkrete rejse (og delstrækninger), så resten af flowet (Art. 12, 18, 19, 20, 30) bruger korrekt retsgrundlag og viser de rigtige UI-beskeder og beregninger.

### Primære regler (kort):

- **Art. 2(6)(a):** Medlemsstater kan fritage **by-/forstads** og **regional** trafik (visse artikler).
  - **Art. 2(6)(b):** Fritagelse for **international** trafik, hvor “en betydelig del og  $\geq 1$  stop” ligger **uden for EU**.
  - **Art. 2(4) (legacy) + Art. 2(4) i 2021/782:** For **long-distance domestic** kan landene forlænge fritagelser for bl.a. **Art. 15, 17, 19, 20(2)(a)-(b), 30(2)** (op til **3. dec. 2029**).
  - **Art. 2(5):** Land kan fritage **Art. 10 (realtidsdata)** indtil **7. juni 2030** (teknisk umulighed).
  - **Art. 2(8):** Fritagelser for regional/urban gælder **ikke** visse artikler (notér især **Art. 5 (cykler)**, 11, 13, 14, 21, 22, 27, 28 – dvs. de skal som udgangspunkt stadig iagttages).
-

## Input (AUTO fra tidligere steps)

- `journey_segments[]`: (fra billet-OCR/PNR + RNE) med:  
`country, operator, train_no, service_type_hint` (regional/IC/HS),  
`dep_station, arr_station, dep_time, arr_time`.
- `distance_km + is Domestic + is International_inside_EU + is International_beyond_EU` (geokodede stationer).
- `ticket_scope`: gennemgående/PNR-scope/særskilte (fra Step Billetinformationer + Art. 12-blokken).
- `seller_type`: operator vs. agency (til Art. 12(5)-ansvar).
- `country_exemptions`: **statisk matrix** med landes fritagelser  
(regional/long-domestic/international-beyond-EU + særregler).  
Eksempler i din matrix:

- **Regional**: særlige fritagelser i AT, BG, FR, HU, LU, PL, PT, RO, SK; **FI** (pendlerdekret 166/2023 → undtag Art. 19, 20(2)(a)-(b) på udpegede pendlerstræk), **DE** (Art. 9(2), 20(2)(a), 30(1) sætning 1), **SE** (<150 km: undtag 8, 9(1), 17–20).
- **Long-distance domestic**: fx **HU** (undtag 19 og 20(2)), **PT** (undtag 17, 19, 20(2)), m.fl.
- **International beyond EU**: fx **PL** (undtag 12, 18(3)); **FI** (til/fra Rusland); **CZ** (afg./ank. i tredjeland undtaget, ekskl. CH); **BG** (bred undtagelsesramme i visse scenarier).
- **Art. 10 undtagelser** indtil 2030: **AT, HR, HU, LV, PL, RO**.

---

## Output (bruges af resten af flowet)

```
{  
  "exemption_profile": {  
    "scope": "regional | long Domestic | intl_inside_eu | intl_beyond_eu",  
    "articles": {  
      "art12": true,  
      "art18_3": true,  
      "art19": true,  
      "art20_2": true,  
      "art30_2": true,  
      "art10": true  
    },  
    "notes": [  
      "SE<150km: art.19/20 undtaget",  
    ]  
  }  
}
```

```

    "FI commuter: art.19 + 20(2)(a)-(b) undtaget",
    "PL intl beyond EU: art.12 + 18(3) undtaget",
    "DE regional: 9(2), 20(2)(a), 30(1) s.1 undtaget",
    "Art.10 undtaget (AT, HR, HU, LV, PL, RO) til 2030"
],
"ui_banners": [
    "Realtime-data (Art.10) kan mangle – vi henter evt. ikke-live RNE.",
    "Kompensation (Art.19) gælder ikke på denne strækning.",
    "Assistance (Art.20(2)) delvist undtaget på denne strækning."
]
}
}

```

- `articles.artXX = true` betyder **gælder** (ikke fritaget). Sæt til `false` hvis **fritaget**.
  - `ui_banners` bruges til at vise kontekstuelle advarsler/vejledning i de efterfølgende steps (Bilag II del II-data, Kompensation, Assistance osv.).
- 

## Beslutningslogik (prioriteret rækkefølge)

### 1. Klassificér tjenesten

- Hvis **by/forstad** → (vi udelader i app'en p.t.) → stop.
- Ellers, afgør **regional** vs. **long-distance domestic** vs. **international**.
  - `is_international_beyond_EU = true`, hvis **≥1 stop uden for EU** og "**betydelig del**" udenfor (konservativt: >25% af km eller tid; parametrisér).

### 2. Slå national fritagelse op i country\_exemptions per segment:

- **regional**: anvend landespecifikke undtagelser (SE<150 km, DE/FI specialer, m.v.).
- **long-domestic**: anvend Art. 2(4)-undtagelser (fx HU/PT) frem til evt. **3. dec. 2029**.
- **intl beyond EU**: anvend landespecifikke "tredjelands" undtagelser (PL, FI, CZ, BG m.fl.).

### 3. Art. 10 (realtidsdata)

Hvis land på ruten er i listen **AT, HR, HU, LV, PL, RO** → `articles.art10=false` (til 7. juni 2030) og vis fallback-banner:

“Art. 10 kan være undtaget. Vi henter ikke-live RNE og beder dig uploadere bekræftelse (Art. 20(4)) ved behov.”

### 4. Art. 12 (gennemgående billetter)

Hvis undtaget (fx **PL intl beyond EU**) → `articles.art12=false` og **disable** gennemgående ansvar:

- Split krav **per billet**; “missed connection” på tværs af **særskilte kontrakter** er ikke dækket i EU-retten på den strækning.
- Vis banner i Art. 12-blokken.

### 5. Art. 18(3) 100-minuttersreglen

Hvis land/stræk er fritaget → `articles.art18_3=false` og vis banner:

“100-min-regel kan være undtaget — vi logger stadig dine udgifter og kører lokal praksis.”

### 6. Art. 19 (kompensation) og Art. 20(2) (assistance)

Sæt `articles.art19 / articles.art20_2` til **false** på undtagne stræk (fx **FI commuter, HU/PT long-domestic, SE<150 km m.fl.**).

Konsekvens:

- Kompensationsmotoren skal **fallback til nationale ordninger** (hvis mere generøse) eller **ingen EU-kompensation** hvor undtaget.
- Assistance (måltider/hotel/transport) deaktiveres delvist/fuldt og UI beder om **udgiftsbilag + lokal praksis**-tekst.

### 7. Art. 30(2) (rettighedsinfo)

Deaktiver hvor national fritagelse gælder (fx DE regional: 30(1) 1. sætning; *du har også 30(2)-cases under long-domestic undtagelser*).

Hvis undtaget → vis kun **app'ens egen** rettighedstekst og NEB-link, men markér at operatøren **ikke** er forpligtet på samme måde lokalt.

## Pseudo-kode (til Copilot)

```
type ExemptionProfile = {
  scope: 'regional' | 'long_domestic' | 'intl_inside_eu' | 'intl_beyond_eu',
  articles: {
    art12: boolean,
    art18_3: boolean,
    art19: boolean,
    art20_2: boolean,
    art30_2: boolean,
    art10: boolean
  },
  notes: string[],
  ui_banners: string[]
};

function buildExemptionProfile(journey: Journey, matrix: CountryExemptions): ExemptionProfile {
```

```

const prof = defaultAllTrueProfile(classifyScope(journey));

// 1) Per-segment national lookups
for (const seg of journey.segments) {
    const cx = matrix.lookup(seg.country, prof.scope, seg);
    prof.articles.art19    &&= !cx.exempt_art19;
    prof.articles.art20_2   &&= !cx.exempt_art20_2;
    prof.articles.art18_3   &&= !cx.exempt_art18_3;
    prof.articles.art12    &&= !cx.exempt_art12;
    prof.articles.art30_2   &&= !cx.exempt_art30_2;
    // Art. 10 (realtime) until 2030
    if (cx.exempt_art10) prof.articles.art10 = false;

    prof.notes.push(...cx.notes);
}

// 2) Intl beyond EU (2(6)(b))
if (isIntlBeyondEU(journey)) {
    const intlX = matrix.lookupIntlBeyondEU(journey);
    applyIntlExemptions(prof, intlX);
}

// 3) UI banners
if (!prof.articles.art10) prof.ui_banners.push("Realtidsdata kan være
undtaget; fallback til ikke-live RNE + dokumentation.");
if (!prof.articles.art19) prof.ui_banners.push("EU-kompensation kan være
undtaget på dele af ruten.");
if (!prof.articles.art20_2) prof.ui_banners.push("Assistance
(måltider/hotel/transport) kan være undtaget; upload udgiftsbilag.");
if (!prof.articles.art12) prof.ui_banners.push("Gennemgående billet-regler
gælder ikke; krav splittes per billet.");
if (!prof.articles.art18_3) prof.ui_banners.push("100-minutters-reglen kan
være undtaget.");

return prof;
}

```

---

## Edge cases (håndtering)

- **Blandet rute** (nogle segmenter undtaget, andre ikke):
  - Gem **per-segment flags** + **aggregér** strengest relevante undtagelser i **exemption\_profile**.
  - Kompensationsmotoren skal **beregne per ansvarlig kontrakt/segment** (Art. 12 → gennemgående vs. særskilt).
- **Art. 10 undtaget:**
  - Brug **ikke-live RNE** + **upload af Art. 20(4) delay-confirmation** som bevis.
  - Slå **Art. 19(9)** (forhåndsoplyst forsinkelse) op i **Bilag II del I pkt. 7** via billetflowet i stedet.
- **Art. 12 undtaget** (fx PL intl-beyond-EU):
  - Disable “missed connection” på tværs af særskilte kontrakter.
  - Beregn og vis kompensation **per billet** (Art. 19(3) principper pr. delstræk).
- **SE <150 km:**
  - Regional **undtagelse** (9(1), 17–20, m.fl.) → hoppes over.
  - Brug lokal/national ordning hvis definéret. Ellers informér om ingen EU-kompensation.
- **FI “commuter decree”:**
  - På udpegede pendlerstrækninger: **Art. 19 + 20(2)(a)-(b)** undtaget.
  - Marker i **notes** og redirect til national praksis (hvis eksisterer).

---

## Integration-hooks (så resten af flowet “forstår” undtagelser)

- **exemption\_profile** injiceres i:
  - **Step Billetinformationer** (vis/skip spørgsmål om Bilag II del II, hvis Art. 10 er undtaget).
  - **Step Art. 12** (disable gennemgående check/ansvar, hvis undtaget).
  - **Step Kompensation (Art. 19)** (disable EU-komp og aktivér national override-tabel).
  - **Step Assistance (Art. 20)** (disable måltider/hotel/transport hvor undtaget – log udgifter i stedet).
  - **Step Bevisførelse** (tilføj **exemption\_profile.notes** i sagsresumé).

---

## Hvor skal du opdatere i Copilot-projektet (kort)

1. **/lib/exemptions/matrix.ts**
  - Landematrix + specialregler (SE<150km, FI commuter, DE regional, HU/PT long-domestic, intl-beyond-EU særregler).
  - Liste for **Art. 10 undtaget** (AT, HR, HU, LV, PL, RO) med udløb 7/6/2030.

2. **/lib/exemptions/classify.ts**
    - classifyScope(journey) + isIntlBeyondEU() (med param. for “betydelig del”).
  3. **/lib/exemptions/buildProfile.ts**
    - Pseudo-koden ovenfor → returnér ExemptionProfile.
  4. **/services/compensationEngine.ts**
    - Respekter exemption\_profile.articles og **national overrides** før EU-fallback.
  5. **UI**
    - **Banner-komponent** der læser exemption\_profile.ui\_banners.
    - **Form-renderer** (vis/skip felter ift. Art. 10/12/18/19/20 fritagelser).
- 

Hvis du vil, kan jeg også levere en **minimal matrix.ts stub** med de lande, du allerede har listet, så du kan paste direkte ind og lade Copilot udfylde resten.

### API calls til DB, SNCF...

Perfekt—her er en samlet, **Copilot-klar** integrationspakke, så du kan køre end-to-end tests med rigtige/semirealistiske billetter (PDF/PNG/PKPass) og auto-udfylde alle 3.x-felter. Jeg giver dig:

1. Arkitektur & fallback-hierarki
  2. API-targets (SNCF, DB, DSB, RNE m.fl.), auth og “good enough” endpoints/stubs
  3. Datamodeller (TypeScript interfaces)
  4. Parser-pipeline (OCR, barcode, PKPass)
  5. Sammenfletning/normalisering til 3.1–3.3 felter
  6. Feature flags & env
  7. Testcases (fixtures + hvordan du kører dem)
- 

## 1) Arkitektur & fallback-hierarki

**Mål:** udfyld 3.1–3.3.5 med mindst mulig brugerinput.

**Kilder, i prioriteret rækkefølge pr. segment:**

1. **Barcode/PNR decode** (QR/Aztec/Code128) → PNR, tognummer, dato, klasse, stationer.
2. **PKPass/Wallet** (Apple/Google) → felter + embedded barcode payload.
3. **Operator API** (SNCF/DB/DSB) → valider PNR; få skema & realtid.
4. **RNE Live Data** (eller national realtid) → faktiske tider + forsinkelse.
5. **OCR** (Tesseract/MLKit) → hvis 1-4 ikke gav alt.

6. **Bruger** → kun de felter, der stadig mangler / konflikter.
- 

## 2) API-targets (overblik)

Bemærk: nogle operatører kræver partneradgang. Nedenfor er “typiske” patterns + forslag til stubs, så du kan teste uden nøgler.

### **SNCF (Frankrig)**

- *Plan/Schedules*: historisk var der et public GTFS/REST; i dag ofte **SNCF Connect partner API** (kræver registrering).
- *Forslag til stub endpoints i din app*:
  - POST `/api/providers/snaf/booking/validate` → body: { pnr, lastName? } → svar: journey segments.
  - GET `/api/providers/snaf/trains/:trainNo?date=YYYY-MM-DD` → køreplan.
  - GET `/api/providers/snaf/realtme/:trainUid` → aktuel afgang/ankomst.

### **Deutsche Bahn (Tyskland)**

- *Fahrplan/Realtid*: DB har HAFAS/IRIS-lignende feeds via partner.
- *Forslag til stub*:
  - GET `/api/providers/db/lookup?pnr=...`
  - GET `/api/providers/db/trip?trainNo=...&date=...`
  - GET `/api/providers/db/realtme?evaId=...&time=...`

### **DSB (Danmark)**

- *Plan/RT*: typisk via Rejseplanen/HAFAS (partner).
- *Stub*:
  - GET `/api/providers/dsb/trip?trainNo=...&date=...`
  - GET `/api/providers/dsb/realtme?uic=...`

### **RNE (RailNetEurope) – Live Data & Claims infra**

- Din produktion skal bruge **RNE Live Data** når adgang åbnes.
- *Midlertidig stub (til tests)*:
  - GET `/api/providers/rne/realtme?trainId=...&date=...` → returnér actualDeparture/Arrival, delayMinutes, disruptionCause?.

### **Andre åbne kilder (til fallback)**

- Nationale “open data” (GTFS-RT/Netex) hvor muligt: opret en generisk adapter:
  - GET /api/providers/open/rt?  
country=XX&trainNo=...&date=...

## Auth & keys

- .env:

PROVIDER\_SNCF\_API\_KEY=...

PROVIDER\_DB\_API\_KEY=...

PROVIDER\_DSB\_API\_KEY=...

PROVIDER\_RNE\_API\_KEY=...

- I dev: ingen eksterne kald → brug **stubs/mocks**, men bevar samme interface.
- 

## 3) Datamodeller (TypeScript)

```
// /lib/types/tickets.ts

export type Money = { currency: string; amount: number };

export type Party = { firstName?: string; lastName?: string; email?: string };

export type StationRef = {
  name: string;
  uic?: string;      // UIC/EVA/DS100 hvor muligt
  country?: string; // ISO-2
};

export type Segment = {
  carrier: string;           // e.g. SNCF, DB, DSB
  trainNo: string;           // e.g. TGV 8412
  category?: string;         // TGV, IC, RE, TER...
  dep: { scheduled: string; actual?: string; station: StationRef };
  arr: { scheduled: string; actual?: string; station: StationRef };
  class?: '1'|'2';
  bike?: { reserved?: boolean; includedInTicket?: boolean };
  pnr?: string;               // record locator
  ticketNo?: string;          // e.g. UIC 13 digits
};
```

```

export type Journey = {
  bookingRef?: string; // PNR på ordre-niveau
  price?: Money;
  segments: Segment[];
  missedConnectionAt?: StationRef; // udfyldes hvis relevant
  finalDelayMinutes?: number; // beregnet ved ankomst
};

export type ClaimFormFields = {
  // 3.1
  undertakingName: string; // 3.1
  // 3.2 (scheduled)
  depDate: string; depStation: string; arrStation: string;
  depTime: string; arrTime: string;
  trainNoCategory: string;
  ticketNumbers: string[];
  ticketPrice?: Money;
  // 3.3 (actual)
  actualDepTime?: string;
  actualArrDate?: string;
  actualArrTime?: string;
  actualTrainNoCategory?: string;
  missedConnectionStation?: string;
};

// /lib/types/providers.ts
export type ProviderTrip = {
  segments: Array<{
    carrier: string; trainNo: string; category?: string;
    dep: { time: string; station: StationRef };
    arr: { time: string; station: StationRef };
  }>;
};

export type ProviderRealtime = {
  segments: Array<{

```

```

trainId?: string;

dep: { scheduled: string; actual?: string; delayMin?: number };
arr: { scheduled: string; actual?: string; delayMin?: number };
causeCode?: string; // til FM-vurdering

}>;
};


```

---

## 4) Parser-pipeline (upload → parse)

```

/services/ingest
└── parseBarcode.ts      // QR/Aztec/Code128 → PNR/trainNo/date
└── parsePKPass.ts       // .pkpass → JSON + embedded barcode
└── parsePDF.ts          // Tesseract + regex fallback
└── identifyCarrier.ts   // heuristik: domæne/logo/format/togtype
└── enrichFromProvider.ts // slå op hos operatør + RNE
└── reconcile.ts         // merge, resolve conflicts

```

### Barcode

- QR/Aztec/Code128: brug libs (server-side) som zxing/canvas.
- Typiske payloads indeholder: PNR, TrainNo, Date, evt. From/To og klasse.

### PKPass

- Udpak .pkpass (ZIP) → pass.json → felter i barcode.message + secondaryFields/auxiliaryFields.
- Ofte: operatørnavn, PNR, tognummer, tider.

### PDF/PNG OCR

- Tesseract: whitelist tal/bogstaver; match med regex:
  - (\bTGV|IC|RE|TER|ICE)\s?\d{1,5}\b
  - ([A-Z]{3,})\s\*- \s\*([A-Z]{3,}) (stationkoder – brug forsigtigt)
  - PNR[ :\s\-\-]([A-Z0-9]{5,8})
  - tider \b([01]\d|2[0-3]):[0-5]\d\b, dato \b\d{2}[ ./-]\d{2}[ ./-]\d{4}\b

### identifyCarrier

- Heuristik: domænenavn på e-kvittering (snecf, bahn, dsb), logo OCR, togkategorier (TGV → SNCF, ICE → DB, IC/RE → DE, TER → FR regional, Øresund/Lyn → DK/SE).

## 5) Sammenfletning til 3.x-felter

```
// /services/ingest/reconcile.ts

import { Journey, ClaimFormFields } from '@/lib/types/tickets';

export function toClaimFields(j: Journey): ClaimFormFields {
  const first = j.segments[0];
  const last = j.segments[j.segments.length - 1];

  return {
    undertakingName: dominantCarrier(j),
    // 3.2 Scheduled (fra provider trip eller billetter)
    depDate: first.dep.scheduled.slice(0,10),
    depStation: first.dep.station.name,
    arrStation: last.arr.station.name,
    depTime: first.dep.scheduled.slice(11,16),
    arrTime: last.arr.scheduled.slice(11,16),
    trainNoCategory: j.segments.map(s => `${s.category ?? ''} ${s.trainNo}`.trim()).join(' / '),
    ticketNumbers: j.segments.map(s => s.ticketNo).filter(Boolean) as string[],
    ticketPrice: j.price,
    // 3.3 Actual (fra realtime)
    actualDepTime: first.dep.actual?.slice(11,16),
    actualArrDate: last.arr.actual?.slice(0,10),
    actualArrTime: last.arr.actual?.slice(11,16),
    actualTrainNoCategory: j.segments.map(s =>
      `${s.category ?? ''} ${s.trainNo}`.trim()
    ).join(' / '),
    missedConnectionStation: j.missedConnectionAt?.name
  };
}

function dominantCarrier(j: Journey): string {
  const byCarrier = new Map<string, number>();
  j.segments.forEach(s => byCarrier.set(s.carrier, (byCarrier.get(s.carrier) ?? 0) + 1));
  return [...byCarrier.entries()].sort((a,b)=>b[1]-a[1])[0]?.[0] ?? '';
}
```

}

### Konfliktløsning (prioritet):

- Station/tider: **Realtid (RNE/operatør)** > plan (operatør) > billet/PKPass > OCR.
  - PNR/ticketNo: **barcode/pkpass** > OCR.
  - Carrier: **PNR/format** > togkategori heuristik.
- 

## 6) Feature flags & env

FEATURE\_PARSING\_OCR=true

FEATURE\_PARSING\_BARCODE=true

FEATURE\_PKPASS=true

FEATURE\_PROVIDER\_SNCF=true

FEATURE\_PROVIDER\_DB=true

FEATURE\_PROVIDER\_DSB=true

FEATURE\_PROVIDER\_RNE=true

// fallback-mode når art. 10 er undtaget i landet:

FEATURE\_REQUIRE\_REALTIME=false

Brug flags i en central config, så Copilot kan generere alternative branches i tests.

---

## 7) Testcases (fixtures + kørsler)

### Fixtures (læg i /tests/fixtures/):

- sncf\_tgv\_pkpass.pkpass (simuleret; indeholder PNR=TGVABC, trainNo=8412, dep/arr tider).
- db\_ice\_pdf.pdf (renderet PDF med QR, PNR=DB1234, ICE 706, Frankfurt → Berlin).
- dsb\_png.png (screenshot med Code128, RE 2245, København → Odense).
- mixed\_chain\_2legs.pdf (2 segmenter, én transaktion → gennemgående).
- se\_lt150\_regional.pdf (SE regional <150 km → kompenstation undtaget).
- pl\_regional.pdf (PL regional → art. 19/20(2) undtaget).
- fr\_regional\_ter.pdf (FR TER → brug lokale regler/ingen EU-komp).

### Automatiserede tests (Vitest/Jest) – eksempel

```
import { parseUpload } from '@/services/ingest/parseUpload';
import { toClaimFields } from '@/services/ingest/reconcile';
import { buildExemptionProfile } from '@/lib/exemptions/profile';
```

```

test('SNCF TGV pkpass udfylder 3.x og realtime fra RNE', async () => {
  const journey = await parseUpload('tests/fixtures/snaf_tgv_pkpass.pkpass');
  // mock provider & RNE svar
  mockSnafTrip(...); mockRneRealtime(...);
  const fields = toClaimFields(journey);
  expect(fields.undertakingName).toBe('SNCF');
  expect(fields.trainNoCategory).toContain('TGV');
  expect(fields.actualArrTime).toBeDefined();
});

test('SE <150km → kompenstation disabled', async () => {
  const j = await parseUpload('tests/fixtures/se_lt150_regional.pdf');
  const profile = buildExemptionProfile(j);
  expect(profile.flags.art19).toBe(false); // kompenstation slæt fra
});

test('Gennemgående billet (art.12(3)) → ansvar hos operatør', async () => {
  const j = await parseUpload('tests/fixtures/mixed_chain_2legs.pdf');
  expect(isThroughTicket(j)).toBe(true);
});

```

---

## Sådan siger du det til Copilot (prompt-snippet)

**Opgave:** Implementér upload → parse → enrich → reconcile pipeline til auto-udfyldelse af “3. Your journey details” samt integration til SNCF/DB/DSB/RNE (med stubs).

**Krav:**

1. Brug filstrukturen under `/services/ingest`  
(`parseBarcode`/`parsePKPass`/`parsePDF`/`identifyCarrier`/`enrichFromProvider`/`reconcile`).
2. Normalisér til `Journey` og map til `ClaimFormFields` via `toClaimFields()`.
3. Læs API-nøgler fra `.env`, men brug stubs i dev.
4. Implementér tests i `/tests` med de angivne fixtures.
5. Respekter `EXEMPTION_MATRIX` i `/lib/exemptions/matrix.ts` for at (de)aktivere kompenstation/assistance/100-min-regel.
6. Fallback-hierarki: `Barcode/PKPass > Provider > RNE > OCR > User`.

7. Lav en demo-page der viser auto-udfyldte 3.x felter og markerer, hvilke kilder der udfyldte hvert felt.
- 

Hvis du vil, kan jeg også levere **mock JSON-svar** til SNCF/DB/DSB/RNE, så testene kører 100% offline.

### **Mock JSON til ovenstående**

Perfekt—her får du et “klar-til-brug” **mock-kit** til SNCF, DB, DSB og RNE, så du kan køre hele upload → parse → enrich → reconcile → auto-udfyld (3.1–3.3) **uden** eksterne nøgler. Det er små, rene JSON-payloads + en let Express-stub, så Copilot kan autogenerere resten.

---

## 1) Mock server (Express)

**Fil:** /mocks/server.ts

```
import express from "express";
import cors from "cors";

const app = express();
app.use(cors());
app.use(express.json());

// ---- SNCF ----
app.post("/api/providers/sncf/booking/validate", (req, res) => {
  // body: { pnr, lastName? }
  return res.json(require("./data/sncf/booking_validate.json"));
}
```

```

});

app.get("/api/providers/sncf/trains/:trainNo", (req, res) => {
  return res.json(require("./data/sncf/trip.json"));
});

app.get("/api/providers/sncf/realtim/:trainUid", (req, res) => {
  return res.json(require("./data/sncf/realtim.json"));
});

// ---- DB (Deutsche Bahn) ----
app.get("/api/providers/db/lookup", (req, res) => {
  return res.json(require("./data/db/lookup.json"));
});

app.get("/api/providers/db/trip", (req, res) => {
  return res.json(require("./data/db/trip.json"));
});

app.get("/api/providers/db/realtim", (req, res) => {
  return res.json(require("./data/db/realtim.json"));
});

// ---- DSB ----
app.get("/api/providers/dsb/trip", (req, res) => {
  return res.json(require("./data/dsb/trip.json"));
});

app.get("/api/providers/dsb/realtim", (req, res) => {
  return res.json(require("./data/dsb/realtim.json"));
});

// ---- RNE ----
app.get("/api/providers/rne/realtim", (req, res) => {
  return res.json(require("./data/rne/realtim.json"));
});

const port = process.env.MOCKS_PORT || 5555;
app.listen(port, () => console.log(`Mock server running on :${port}`));

```

**NPM scripts (package.json):**

```
{
}
```

```
"scripts": {  
    "mocks": "tsx mocks/server.ts"  
}  
}
```

#### **Env-toggle (fx .env.development.local):**

PROVIDER\_BASE\_URL=http://localhost:5555/api/providers

USE\_PROVIDER\_STUBS=true

Din provider-client skal læse USE\_PROVIDER\_STUBS og bruge PROVIDER\_BASE\_URL i dev.

---

## 2) Mock payloads (JSON)

Læg nedenstående filer i:

```
/mocks/data/  
    sncf/{booking_validate.json, trip.json, realtime.json}  
    db/{lookup.json, trip.json, realtime.json}  
    dsb/{trip.json, realtime.json}  
    rne/{realtime.json}
```

SNCF

sncf/booking\_validate.json

```
{  
    "pnr": "TGVABC",  
    "undertaking": "SNCF",  
    "passengers": [{"lastName": "DUPONT"}],  
    "price": {"currency": "EUR", "amount": 89.0},  
    "segments": [  
        {  
            "carrier": "SNCF",  
            "trainNo": "8412",  
            "category": "TGV INOUI",  
            "dep": {"time": "2025-03-20T08:04:00+01:00", "station": {"name": "Paris Gare de Lyon", "uic": "008798600"}},  
            "arr": {"time": "2025-03-20T10:41:00+01:00", "station": {"name": "Lyon Part-Dieu", "uic": "008778700"}},  
            "class": "2",  
            "pnr": "TGVABC",  
            "ticketNo": "1187-1234567890"
```

```
    }
]
}

sncf/trip.json
{
  "segments": [
    {
      "carrier": "SNCF", "trainNo": "8412", "category": "TGV INOUI",
      "dep": {"time": "2025-03-20T08:04:00+01:00", "station": {"name": "Paris Gare de Lyon", "uic": "008798600", "country": "FR"}},
      "arr": {"time": "2025-03-20T10:41:00+01:00", "station": {"name": "Lyon Part-Dieu", "uic": "008778700", "country": "FR"}}
    }
  ]
}

sncf/realtimetime.json
{
  "segments": [
    {
      "trainId": "TGV_8412_20250320",
      "dep": {"scheduled": "2025-03-20T08:04:00+01:00", "actual": "2025-03-20T08:20:00+01:00", "delayMin": 16},
      "arr": {"scheduled": "2025-03-20T10:41:00+01:00", "actual": "2025-03-20T10:58:00+01:00", "delayMin": 17},
      "causeCode": "TECH"
    }
  ]
}

Deutsche Bahn (DB)
db/lookup.json
{
  "pnr": "DB1234",
  "undertaking": "DB",
  "segments": [
    {
      "carrier": "DB", "trainNo": "706", "category": "ICE",
```

```
        "dep": {"time": "2025-04-05T09:14:00+02:00", "station": {"name": "Frankfurt(Main) Hbf", "uic": "008000105", "country": "DE"}},  
        "arr": {"time": "2025-04-05T12:28:00+02:00", "station": {"name": "Berlin Hbf", "uic": "008011160", "country": "DE"}},  
        "class": "1", "ticketNo": "080-5555555555"  
    }  
]  
}  
db/trip.json
```

```
{  
    "segments": [  
        {  
            "carrier": "DB", "trainNo": "706", "category": "ICE",  
            "dep": {"time": "2025-04-05T09:14:00+02:00", "station": {"name": "Frankfurt(Main) Hbf", "uic": "008000105", "country": "DE"}},  
            "arr": {"time": "2025-04-05T12:28:00+02:00", "station": {"name": "Berlin Hbf", "uic": "008011160", "country": "DE"}},  
        }  
    ]  
}
```

db/realtime.json

```
{  
    "segments": [  
        {  
            "trainId": "ICE706_20250405",  
            "dep": {"scheduled": "2025-04-05T09:14:00+02:00", "actual": "2025-04-05T09:55:00+02:00", "delayMin": 41},  
            "arr": {"scheduled": "2025-04-05T12:28:00+02:00", "actual": "2025-04-05T13:30:00+02:00", "delayMin": 62},  
            "causeCode": "INFRA"  
        }  
    ]  
}
```

DSB

dsb/trip.json

```
{  
    "segments": [  
        {
```

```
        "carrier":"DSB", "trainNo":"RE 2245", "category":"RE",
        "dep": {"time": "2025-05-12T15:02:00+02:00", "station": {"name": "København H", "uic": "008600100", "country": "DK"}},
        "arr": {"time": "2025-05-12T16:15:00+02:00", "station": {"name": "Odense", "uic": "008607002", "country": "DK"}},
        "class": "2", "ticketNo": "DK-20250512-0001"
    }
]
}
```

dsb/realtimetime.json

```
{
  "segments": [
    {
      "trainId": "RE2245_20250512",
      "dep": {"scheduled": "2025-05-12T15:02:00+02:00", "actual": "2025-05-12T15:07:00+02:00", "delayMin": 5},
      "arr": {"scheduled": "2025-05-12T16:15:00+02:00", "actual": "2025-05-12T16:55:00+02:00", "delayMin": 40},
      "causeCode": "OTHER"
    }
  ]
}
```

RNE (generisk realtime)

rne/realtimetime.json

```
{
  "segments": [
    {
      "trainId": "GENERIC_2LEGS_20250520_A",
      "dep": {"scheduled": "2025-05-20T07:15:00+02:00", "actual": "2025-05-20T07:35:00+02:00", "delayMin": 20},
      "arr": {"scheduled": "2025-05-20T08:30:00+02:00", "actual": "2025-05-20T08:55:00+02:00", "delayMin": 25},
      "causeCode": "WEATHER"
    },
    {
      "trainId": "GENERIC_2LEGS_20250520_B",
      "dep": {"scheduled": "2025-05-20T09:05:00+02:00", "actual": "2025-05-20T09:15:00+02:00", "delayMin": 10},
      "arr": {"scheduled": "2025-05-20T10:30:00+02:00", "actual": "2025-05-20T10:45:00+02:00", "delayMin": 15}
    }
  ]
}
```

```

        "arr": {"scheduled": "2025-05-20T10:45:00+02:00", "actual": "2025-05-20T11:22:00+02:00", "delayMin": 37},
        "causeCode": "CONNECTION_MISSED"
    }
]
}

```

---

## 3) Eksempel på **barcode payloads** (til tests)

Læg som **plain text** ved siden af dine fixtures, så parseren kan læse det.

### **SNCF QR (simuleret):**

FMT=SNCF;PNR=TGVABC;TRAIN=8412;CAT=TGV;DEP=2025-03-20T08:04:00+01:00;ARR=2025-03-20T10:41:00+01:00;FROM=PARIS GARE DE LYON;TO=LYON PART-DIEU;CLASS=2;TICKET=1187-1234567890

### **DB Aztec/Code128 (simuleret):**

FMT=DB;PNR=DB1234;TRAIN=ICE706;DEP=2025-04-05T09:14:00+02:00;FROM=FRANKFURT(MAIN) HBF;TO=BERLIN HBF;CLASS=1;TICKET=080-5555555555

### **DSB Code128 (simuleret):**

FMT=DSB;PNR=DSB7788;TRAIN=RE2245;DEP=2025-05-12T15:02:00+02:00;FROM=KØBENHAVN H;TO=ODENSE;CLASS=2;TICKET=DK-20250512-0001

---

## 4) PKPass fixture (uddrag)

Fil: /tests/fixtures/snfc\_tgv\_pkpass/pass.json

```

{
    "formatVersion": 1,
    "passTypeIdentifier": "pass.snfc.tgv",
    "serialNumber": "TGVABC",
    "organizationName": "SNCF",
    "description": "TGV INOUI 8412 Paris-Lyon",
    "barcode": {
        "format": "PKBarcodeFormatQR",
        "message": "FMT=SNCF;PNR=TGVABC;TRAIN=8412;DEP=2025-03-20T08:04:00+01:00;FROM=PARIS GARE DE LYON;TO=LYON PART-DIEU",
        "messageEncoding": "iso-8859-1"
    },
    "eventTicket": {
        "headerFields": [{"key": "train", "label": "Train", "value": "TGV 8412"}],

```

```

    "primaryFields": [{ "key": "route", "label": "Route", "value": "Paris → Lyon"}],
    "auxiliaryFields": [
        {"key": "dep", "label": "Departs", "value": "2025-03-20 08:04"},
        {"key": "arr", "label": "Arrives", "value": "2025-03-20 10:41"},
        {"key": "class", "label": "Class", "value": "2"}
    ]
}
}

```

---

## 5) OCR test-tekst (PDF/PNG)

Læg som ren tekst i en PDF (typisk e-billet):

SNCF e-ticket

Passenger: Dupont

PNR: TGVABC

Train: TGV 8412

Paris Gare de Lyon (08:04) → Lyon Part-Dieu (10:41)

Class: 2

Ticket: 1187-1234567890

Regex i din parser kan trække PNR, TGV 8412, tider og stationer.

---

## 6) Samling i klient (provider-adapter)

**Fil:** /lib/providers/client.ts

```

const base = process.env.PROVIDER_BASE_URL!;
const useStubs = process.env.USE_PROVIDER_STUBS === "true";

export async function getSncfTrip(trainNo: string, dateISO: string) {
    const url = useStubs
        ? `${base}/snCF/trains/${encodeURIComponent(trainNo)}?date=${dateISO}`
        : /* prod URL her */ `${base}/snCF/trains/${encodeURIComponent(trainNo)}?
date=${dateISO}`;
    return fetch(url).then(r => r.json());
}

export async function getSncfRealtime(trainUid: string) {
    const url = useStubs

```

```

? `${base}/snCF/realtIme/${encodeURIComponent(trainUid)}`
: /* prod URL her */ `${base}/snCF/realtIme/${encodeURIComponent(trainUid)}
`;
return fetch(url).then(r => r.json());
}

// tilsvarende små wrappers for DB, DSB, RNE...

```

---

## 7) Demo-side (hurtig)

Fil: /pages/demo/claim-preview.tsx

- Upload (PDF/PNG/PKPass)
  - Kør parseUpload() → identifyCarrier() → enrichFromProvider() → toClaimFields()
  - Vis felter 3.1–3.3 **med badge pr. datakilde** (fx “RNE”, “OCR”, “Barcode”, “PKPass”, “User”).
  - Knap: “Udfyld EU-formular (preview)” → rendér PDF-preview (du har skabelonen).
- 

## 8) Testidéer (kører 100% offline)

- **snCF\_tgv\_pkpass.pkpass** → udfylder alle 3.x; realtime fra snCF/realtIme.json.
  - **db\_ice\_pdf.pdf** + barcode → PNR+tog → db/trip.json + db/realtIme.json.
  - **dsb\_png.png** → dsb/trip.json + dsb/realtIme.json.
  - **mixed\_chain\_2legs.pdf** → kombi af 2 segmenter; beregn total forsinkelse og “missedConnectionAt”.
  - **Exemption-cases:** SE <150 km, PL regional → slå kompensation fra i din EXEMPTION\_MATRIX.
- 

## Hvad du siger til Copilot

**Implementér provider-mocks:** Brug /mocks/server.ts + JSON-payloads. Læs USE\_PROVIDER\_STUBS og PROVIDER\_BASE\_URL.

**Parser-pipeline:** barcode > pkpass > provider > RNE > ocr > user.

**Map til ClaimFormFields** via toClaimFields().

**Demo-side:** Vis 3.1–3.3 udfyldt + badges for kilder.

**Tests:** Brug fixtures under /tests/fixtures/ og mock-serveren.

Hvis du vil, kan jeg også smide **to yderligere JSON'er**: en **gennemgående 2-legs-rejse (én transaktion)** og en **“separate contracts”** variant, så du kan teste Art. 12(3) vs. 12(5) logikken i kompensationsmotoren.

Perfekt — her er et **færdigt Copilot-klar blueprint** til at gøre hele trin 2 **helt automatisk** (inkl. “demo-mode” med auto-udfyldte felter fra mocked API'er). Du kan copy-paste stumperne direkte ind i dit repo.

---

## 1) Dataflow (autofyld + fallback)

1. **Upload billetten** (PDF/PNG/PKPass/screendump)
  - kør **OCR + barcode/QR-decode**
  - parse **PNR/BookingRef**, tognummer, dato, afgang/ankomst, klasse, pris.
2. **Berig med live/plan-data**
  - slå PNR/tog op i **RNE** + operatør-open-API (SNCF, DB, DSB ...)
  - hent planlagt/aktuel køreplan, forsinkelse, missede forbindelser.
3. **Auto-udfyld PDF-felter (Kommissionens formular)**
  - map felter fra dit **JourneyRecord** til sektion **3.1–3.3**.
4. **Fallback-spørgsmål**
  - vis kun præcise mikro-prompts hvis et felt mangler **eller** har lav tillid.

---

## 2) Typer & schema (én sandhed i UI)

```
// core types  
type Confidence = { value: string; score: number }; // 0..1
```

```
interface JourneyRecord {  
    // FRA BILLETEN (OCR/Barcode)  
    bookingRef?: Confidence;          // 3.2.7  
    ticketNumber?: Confidence;        // 3.2.7 (alternativt)  
    operatorName?: Confidence;        // 3.1  
    trainNo?: Confidence;            // 3.2.6 / 3.3.4  
    trainCategory?: Confidence;       // 3.2.6 / 3.3.4 (IC, TGV, TER ...)  
    classOfService?: Confidence;      // 1./2. kl.  
    ticketPrice?: Confidence;         // 3.2.8 (beløb + valuta)  
  
    // PLANLAGTE TIDER/STATIONER  
    depStation?: Confidence;          // 3.2.2  
    arrStation?: Confidence;          // 3.2.3  
    depDate?: Confidence;             // 3.2.1 (ISO)  
    schedDepTime?: Confidence;        // 3.2.4 (HH:mm)  
    schedArrTime?: Confidence;        // 3.2.5 (HH:mm)  
  
    // FAKTISKE TIDER (fra RNE/operator + evt. bruger)  
    actualDepTime?: Confidence;       // 3.3.2  
    actualArrDate?: Confidence;       // 3.3.1  
    actualArrTime?: Confidence;       // 3.3.3  
    missedConnectionAt?: Confidence; // 3.3.5  
  
    // INTERN  
    segments: Array<{  
        trainNo: string;  
        operator: string;  
        origin: string;  
        destination: string;  
        schedDep: string;    // ISO  
        schedArr: string;    // ISO  
        actDep?: string;     // ISO  
        actArr?: string;     // ISO  
        delayMinutes?: number;  
        euTerritory: boolean;  
    }>;
```

```

} >;

sourceHashes: string[]; // bilag/checksums til bevis
}

```

#### Tillidsgrænser (helt simpelt):

- `score ≥ 0.85` → autofyld & lås felt.
  - `0.5 ≤ score < 0.85` → autofyld men vis inline-badge “tjek venligst”.
  - `< 0.5` → spørg bruger (mikro-prompt).
- 

## 3) Felt-mapping til EU-formular (sektion 3)

Formularfelt	JourneyRecord-felt	Kilde/berigelse
<b>3.1 Name of railway undertaking</b>	<code>operatorName.value</code>	billet-OCR → bekræft med RNE/operatør
<b>3.2.1 Departure date</b>	<code>depDate.value</code>	billet → valider mod RNE
<b>3.2.2 Departure station</b>	<code>depStation.value</code>	billet → normaliser (UIC/IFOPT)
<b>3.2.3 Destination station</b>	<code>arrStation.value</code>	billet → normaliser
<b>3.2.4 Scheduled time of departure</b>	<code>schedDepTime.value</code>	billet/RNE
<b>3.2.5 Scheduled time of arrival</b>	<code>schedArrTime.value</code>	billet/RNE
<b>3.2.6 Train no./category</b>	<code>trainNo.value + trainCategory.value</code>	billet/OCR + RNE
<b>3.2.7 Ticket No(s)/Booking Ref.</b>	<code>ticketNumber/bookingRef</code>	barcode/QR/OCR
<b>3.2.8 Ticket price(s)</b>	<code>ticketPrice.value</code>	billet (valuta detect)
<b>3.3.1 Date of actual arrival</b>	<code>actualArrDate.value</code>	RNE → fallback bruger
<b>3.3.2 Actual time of departure</b>	<code>actualDepTime.value</code>	RNE → fallback bruger
<b>3.3.3 Actual time of arrival</b>	<code>actualArrTime.value</code>	RNE → fallback bruger
<b>3.3.4 Train no./category (actual)</b>	<code>trainNo/trainCategory</code>	RNE
<b>3.3.5 Missed connection in (station)</b>	<code>missedConnectionAt.value</code>	bereg af segment-overlap, eller spørg

## 4) API-grænseflader (virkelige + demo)

```

// OCR + barcode
POST /api/ingest/ticket

```

```

body: { fileId: string }

resp: { journey: JourneyRecord, logs: string[] }

// RNE lookup (pr. toget/PNR)
GET /api/rne/trip?bookingRef=...&date=...
resp: { segments: JourneyRecord["segments"] }

// Operatør-API (fallback/berigelse)
GET /api/operator/{operatorCode}/trip?trainNo=...&date=...
resp: { ...same shape as segments... }

// Demo fixtures
GET /api/demo/fixtures?case=tgv_90m
resp: { journey: JourneyRecord }

```

---

## 5) Demo-mode (auto-udfyldt “wow-effekt”)

```

interface DemoOptions {
    enabled: boolean;
    caseId?: "tgv_30m"|"ice_125m"|"ter_missed_conn"|"ic_no_rne";
}

function getJourney(options: DemoOptions): Promise<JourneyRecord> {
    if (options.enabled) return fetch(`/api/demo/fixtures?case=$
{options.caseId||"ice_125m"}`).then(r=>r.json()).then(x=>x.journey);
    // ellers: rigtig pipeline
}

```

### Forslag til 4 demo-cases:

- **ice\_125m**: Gennemgående billet, 125 min → 50 % komp.
  - **tgv\_30m**: 30–59 min → *national* voucher (SNCF G30) + 19(1) vurdering.
  - **ter\_missed\_conn**: regional, missed connection, Art. 12-logik.
  - **ic\_no\_rne**: manglende live data → bruger får mikro-prompts for 3.3-felter.
- 

## 6) UI-logik (vis/skip + mikro-prompts)

**Generelt:** vis spørgsmål **kun** når **score < 0.5 eller** mismatch mellem billet og RNE.

### Eksempler (klar til copy-paste som tekstblokke):

- “Vi mangler det planlagte ankomsttidspunkt. Kan du bekræfte (HH:mm)?”  
→ skriver til `schedArrTime.value + score=1`
  - “RNE viser, at du ankom 22:41. Passer det?”  
→ Ja → `actualArrTime.score=1` / Nej → åbent inputfelt
  - “Hvor skete den missede forbindelse? (stationsnavn)”  
→ skriver til `missedConnectionAt`
  - “Kan du bekræfte billetpris og valuta?”  
→ `ticketPrice` normaliseres (EUR/DKK/SEK)
- 

## 7) Minimal beregner (server eller edge)

```
function computeDelay(j: JourneyRecord): number {  
    // brug sidste segment  
    const last = [...j.segments].pop();  
    if (!last?.actArr || !last?.schedArr) return NaN;  
    return Math.max(0, Math.round((+new Date(last.actArr) - +new  
        Date(last.schedArr))/60000));  
}  
  
function computeCompensation(j: JourneyRecord, nationalRules?: any) {  
    const minutes = computeDelay(j);  
    if (!isFinite(minutes) || minutes < 60) return { pct: 0, amount: 0 };  
  
    // EU baseline  
    let pct = (minutes >= 120) ? 0.50 : 0.25;  
  
    // gennemgående vs. segment-pris (Art. 19(3))  
    // hvis delpriser mangler → halv billetpris ved retur, ellers totalpris  
    const price = parseFloat(j.ticketPrice?.value || "0");  
    const amountEU = +(price * pct).toFixed(2);  
  
    // national override (hvis mere gunstig)  
    const amountNat = nationalRules ? Math.max(amountEU, nationalRules.calc(j)) :  
        amountEU;  
  
    return { minutes, pct, amount: amountNat };  
}
```

}

---

## 8) Hooks til resten af flowet

- `onJourneyAutofill(journey)` → kør **Art. 12-klassifikation** (gennemgående vs. særskilte)
  - `onJourneyValidated(journey)` → kør **Art. 19-beregning**
  - `onCompensationComputed(result)` → vis **Art. 20-assistance** prompts hvis `minutes ≥ 60`
  - `onSubmission()` → generér **EU-formular** (PDF), vedlæg `sourceHashes`, RNE-snapshot, bilag
- 

## 9) Mikro-prompts (færdige tekster, kun ved behov)

- **Planlagt afgang (3.2.4):**  
“Hvad var planlagt afgangstid fra {{depStation}}? (HH:mm)”
  - **Planlagt ankomst (3.2.5):**  
“Hvad var planlagt ankomsttid til {{arrStation}}? (HH:mm)”
  - **Faktisk ankomst (3.3.3):**  
“Hvad tid ankom du faktisk til {{arrStation}}? (HH:mm)”
  - **Missed connection (3.3.5):**  
“Hvilken station missede du forbindelsen på?”
  - **Billetpris (3.2.8):**  
“Hvad betalte du for billetten? (beløb + valuta)”
- 

## 10) “Done-state” (klar til udbetaling/indsendelse)

```
{  
  "journey": { "...": "autofyldt + verificeret" },  
  "calc": { "minutes": 128, "pct": 0.5, "amount": 42.50, "currency": "EUR" },  
  "evidence": ["ticket.pdf", "rne.json", "op-conf.txt"],  
  "poa": { "signed": true, "id_verified": true, "signature_hash":  
    "sha256:..." },  
  "ready_to_submit": true  
}
```

---

## Bonus: “One-click demo”

- **Feature flag:** NEXT\_PUBLIC\_DEMO=true
- **Demo-knap:** “Indlæs eksempel (ICE 128 min forsinket)” → /api/demo/fixtures?case=ice\_125m
- UI låser 90 % af felterne (score $\geq$ 0.85) og viser 1–2 mikro-prompts med realistiske udfordringer (fx manglende 3.3.5).

---

Hvis du vil, kan jeg også smide **færdige JSON-fixtures** for de fire demo-cases og en lille **pdf-filler** (Node + pdf-lib) der tager JourneyRecord → udfylder Kommissionens skema → lægger dine bilag i en samlet ZIP.

Perfekt ✅ — her får du **den samlede, diagramklare tekst til dit “Step Udregning + Bevisførelse”-flow**, i et format du kan copy-paste direkte ind i Draw.io / LibreOffice ODG / Lucidchart.

Jeg har bevaret **samme struktur og tone som dine eksisterende bokse** (kort, præcist, artikelreferencer, dependencies).

Flowet kobler hele kæden **Billet → Art. 12 → Refusion (18) → Kompensation (19) → Assistance (20) → Bevisførelse (CIV + Bilag II)**.

---

## ✳ HOVEDTRIN – FLOWTEKST TIL DIAGRAM

---

### ■ Step Billetinformation (Art. 10 + Bilag II, del I-II)

**Formål:** Indhente og verificere alle billet- og realtidsdata.

**AUTO:** OCR + RNE + operatørdata.

**Dependencies:**

- Art. 12 (1–7): Gennemgående billet eller særskilte kontrakter
- Bilag II del 1: prækontraktuelle oplysninger
- Bilag II del 2: realtidsdata
- CIV (Art. 6–25): Passagerens ansvar og adfærd

**Handlinger:**

- Sammenlign billetinfo ↔ RNE ↔ faktiske forhold
  - Log mismatch (f.eks. klasse, rute, cykel, forsinkelse)
  - Markér manglende data (trigger brugerflow for spørgsmål)
-

## Step Art. 12 (1–7) – Gennemgående billet

**AUTO:** check via PNR / booking-scope / operatør-flag.

**Brugerspørgsmål (kun hvis ukendt):**

- Var det én kontrakt (gennemgående) eller flere særskilte?
- Købt hos operatør eller via rejsebureau?

**Dependencies:**

- Art. 19(3): beregning af kompenstation
- Art. 18: ansvar for omlægning
- Step Bevisførelse: ansvarsfordeling

**Output:**

- `through_ticket=true/false`
- `responsible_party=operator/agency`

---

## Step Refusion (Art. 18 + CIV 6–25)

**Formål:** Afklare, om rejsen blev aflyst, omlagt eller refunderet før kompenstation.

**AUTO:** fra operatørdata eller brugersvar.

**Brugerspørgsmål (vises kun hvis mismatch):**

- Fik du tilbuddt refusion eller omlægning?
- Omlægning på tilsvarende vilkår (Art. 18(1)(b)) eller efter eget valg (Art. 18(1)(c))?
- Blev du informeret inden for 100 minutter (Art. 18(3))?
- Medførte omlægningen ekstra udgifter?
- Blev du nedklassificeret?

**Dependencies:**

- Art. 12 (kontrakt-type)
- Bilag II del 1–4, 1–6, 1–8 (service, klasse, faciliteter)
- CIV art. 6–25 (selvforskyldt adfærd)

**Handling:**

- Hvis refusion allerede givet → kompenstation springes over
- Ved downgrade → beregn delvis refusion (Art. 18(1)(c))
- Ved overtrædelse af 100-min-reglen → marker manglende overholdelse

## Step Kompensation (Art. 19)

**Formål:** Beregne kompensationsbeløb og vurdere undtagelser.

**AUTO:** Forsinkelsesdata + billetpris + gennemgående billet + exemptions.

**Grundregler:**

- 25 % ved 60–119 min
- 50 % ved  $\geq 120$  min
- Kun EU-segmenter tæller med (Art. 19(4))
- Ingen kompensation hvis refusion allerede udbetalt (Art. 18)
- Ingen kompensation, hvis forsinkelsen var kendt før køb (Art. 19(9))

**Force majeure / doktrin (Art. 10 + E10):**

- Natur, sundhed, sabotage → udeluk kompensation
- Strejke eller infrastrukturfejl → kompensation stadig gældende
- Selvforskyldt adfærd (CIV 6–25) → afvisning
- Alle FM-sager flagges til manuel gennemgang

**Output:**

- `compensation_pct=25/50`
- `compensation_amount=€X`
- `manual_review=true/false`

---

## Step Assistance (Art. 20(2)–(4))

**Formål:** Registrere tilbuddet hjælp og udgifter ved  $\geq 60$  min forsinkelse / aflysnings.

**AUTO:** RNE + brugerdata + exemptions.

**Spørgsmål (kun ved manglende data):**

- Fik du måltider/forfriskninger? (20(2)(a))
- Fik du hotel/indkvartering + transfer? (20(2)(b))
- Fik du alternativ transport fra blokeret tog? (20(2)(c))
- Fik du alternativ transport, hvis forbindelsen blev afbrudt? (20(3))
- Har du haft udgifter (mad, hotel, taxi osv.)?
- Fik du skriftlig bekræftelse på forsinkelse/aflysnings? (20(4))

**Ekstraordinære forhold:**

- Kun relevant for hotel-loft → max 3 nætter (Art. 19(10) + 20(2)(b))

**Output:**

- meal\_offered, hotel\_offered, alt\_transport\_provided, delay\_confirmation\_received
  - extra\_expense\_upload (bilag)
  - extraordinary\_flag
- 

## ■ Step Udregning af kompensation (samlet)

**Formål:** Samle og beregne endelig ret til kompensation/refusion.

**Formel (Art. 19(1)+(3)):**

```
if delay < 60 min → ingen kompensation  
if 60–119 min → 25 % af betalt pris  
if ≥120 min → 50 % af betalt pris
```

**Justeringer:**

- Fratræk tidligere refusion (Art. 18)
- Kun EU-segmenter (Art. 19(4))
- Nationale satser (bilag X)
- Ingen kompensation, hvis forsinkelsen var kendt før køb (Art. 19(9))

**Force majeure / CIV:**

- Ekstraordinære forhold (Art. 19(10)) → udeluk kompensation
- Selvforskyldt (CIV art. 6–25) → afvisning
- Tredjepart (infrastruktur, strejke) → stadig kompensation

**Output:**

- compensation\_amount
  - compensation\_pct
  - extraordinary\_flag
  - manual\_review
- 

## ■ Step Bevisførelse (Bilag II + CIV + Art. 20(4))

**Formål:** Samle alt dokumentationsmateriale for sagsbehandling.

**AUTO:** samler data fra alle steps.

**Beviskilder:**

- Billet/PNR
- RNE realtime snapshot
- Operatørpåtegning (Art. 20(4))

- Udgiftsbilag (hotel, taxi, mad)
- Kommunikation (årsagsangivelse)
- Bilag II del I + II (lovet vs. leveret service)
- Exemption-matrix
- Force majeure årsag

#### **Krydkontrol:**

- Billet ↔ RNE → faktisk forsinkelse
- Operatør ↔ bruger → årsagsmatch
- Bilag II I vs. II → bevis for downgrade/mangelnde service
- Exemptions ↔ Jurisdiktion → justér kravtype

#### **Output (JSON):**

```
{
  "delay_minutes": 128,
  "ticket_price": 85,
  "compensation_amount": 42.50,
  "currency": "EUR",
  "manual_review": false,
  "attachments": ["ticket.pdf", "rne.json", "hotel.pdf"]
}
```

#### **Afslutning:**

- Generér PDF-sagsresumé
- Log uploadhash + audittrail
- Ved kravoverdragelse → fratræk 25 % fee

## Diagramstruktur (anbefalet layout)

[BILLETINFO]

↓

[ART.12 – GENNEMGÅENDE BILLET]

↓

[REFUSION – ART.18 + CIV]

↓

[KOMPENSATION – ART.19]

↓

[ASSISTANCE – ART.20]

↓

[UDREGNING AF KOMPENSATION]

↓

[BEVISFØRELSE – BILAG II + CIV + ART.20(4)]

↓

[PDF & LOG + MANUAL REVIEW FLAG]

---

Vil du have, at jeg laver det som en **færdig “diagramtekstblok-fil” (.txt eller .csv)**, som du kan importere direkte i Draw.io eller ODG, så du slipper for manuelt copy-paste pr. boks?  
(Jeg kan sætte formatering og pile i eksportvenlig rækkefølge.)

Klart! Her er en hurtig, praktisk oversigt over nationale (operatør-)ordninger, der er MERE generøse end EU-minimum (art. 19) for **regional** og **langdistance** togrejser (jeg udelader urban/suburban som aftalt). Brug den som “national override” i din kompensationslogik, før du falder tilbage på EU-reglen ( $\geq 60 \text{ min} \Rightarrow 25\% / \geq 120 \text{ min} \Rightarrow 50\%$ ).

Hvor er det bedre end EU-minimum?

→ NOTE :

- **Frankrig – SNCF (TGV INOUI & Intercités)**

“G30”: kompensation allerede ved  $\geq 30 \text{ min}$  (25%),  $\geq 2 \text{ timer}$  (50%). Ved 30–59 min gives voucher; fra 60 min kan du vælge bankoverførsel. Gælder for TGV INOUI & Intercités (TER håndteres regionalt). ([snfc-connect.com](#))

- **Spanien – Renfe (serviceafhængigt, særligt AVE/HS)**

Renfes “Punctuality commitment” er typisk langt over EU-minimum for højhastighed:  
*Eksempel (international AVE Spanien/Frankrig – Renfe side):* 25%  $\geq 30 \text{ min}$ ; 50%  $\geq 120 \text{ min}$ ; 75%  $\geq 180 \text{ min}$  (serviceafhængigt). For rene spanske HS-stræk annoncerer Renfe bl.a. 50%  $\geq 60 \text{ min}$  og **100%  $\geq 90 \text{ min}$**  (og på visse AVE-linjer endda fra 15/30 min). **Konklusion:** modelér per togtype/linje – der er eksplisitte tabeller pr. produkt. ([Renfe](#))

- **Italien – Trenitalia (Frecce)**

**Bonus ved 30–59 min = 25%** af billetprisen (voucher), udover EU-minimum for  $\geq 60 / \geq 120 \text{ min}$ . (Frecce = Frecciarossa/Frecciargento/Frecciabianca). ([Trenitalia](#))

- **Italien – Italo**

Holder sig i praksis på EU-niveau (25%  $\geq 60 / 50\% \geq 120$ ); ikke mere generøs end EU-minimum. (Med i listen for at undgå at antage “Italien generelt” = mere generøst.) ([italotreno.com](#))

- **Nederlandene – NS (indenlandsk)**

Fra **≥30 min** kan der søges om refundering. Størrelsen afhænger af billettype/abonnement; NS opererer med to tærskler **30–59 min** og **≥60 min** (ofte 50%/100% for standard enkeltbilleter; tjek tabellen pr. billet). Officielle vilkår og kundeside dokumenterer 30-min-tærsklen; procentsatser fremgår af produkttabellerne.

- **Danmark – DSB (fjern & regional)**

Fra **≥30 min** forsinkelse (Rejsetidsgaranti). Udbetaling sker efter DSB's nationale ordning, som typisk ligger over EU's tærskel. (Bemærk: udbetalingsform og satser afhænger af produkt). ([dsb.dk](#))

- **Sverige – SJ**

**Meget generøst:** **50% ved 20 min, 75% ved 40 min, 100% ved 60 min** (for mange SJ-produkter). Brug SJ's officielle "Rättigheter vid försening" som kilde i din mapping. ([SJ](#))

- **Belgien – SNCB/NMBS**

Standard er **fra ≥60 min** (som EU-minimum) + særligt ordning for **hyppige** forsinkelser (kan give 50% i intervallet 30–59 min kun ved gentagne forsinkelser). Ikke generelt mere generøst på enkeltforsinkelser, men *mere* for pendlercases. ([belgiantrain.be](#))

- **Tyskland – DB**

Overvejende på EU-niveau (25%  $\geq 60$  / 50%  $\geq 120$ ). Der findes særlige, faste beløb for visse periodekort (BahnCard 100 m.v.), men ingen generelt lavere tærskler end EU for enkeltbilletter. ([bahn.de](#))

Sådan bygger du det ind i din app (kort og skarpt)

1. **Identificér operatør + produktklassen** (OCR/PKPass/PNR → mapping: f.eks. "SNCF:TGV INOUI", "Renfe:AVE", "Trenitalia:Frecce", "NS:Domestic", "DSB:Fjern/Regional", "SJ:Inrikes").
2. **National override-tabel** (service-niveau): lag felter `min_delay_min, tiers` (liste af `{min_delay, percent, payout_type}`), `notes` (fx "voucher <60 min", "kun ved gentagne forsinkelser").
3. **Fallback** til EU 2021/782 art. 19, hvis ingen match ELLER hvis rejsen/operatøren er i et land med undtagelse af art. 19 (din "rail exemptions" check kører allerede).
4. **Vælg højeste berettigede kompensation:** hvis national ordning > EU, brug den – ellers EU.
5. **Begrænsninger/undtagelser:** respekter art. 19(9) (forhåndsoplyst forsinkelse ⇒ ingen komp.), art. 19(10) (FM), samt lokale "voucher only" under visse grænser (SNCF 30–59 min, Trenitalia 30–59 min).

Bemærkninger du bør kode ind som "gotchas"

- **SNCF TER** styres regionalt (se regionsider); lad din backend markere TER som "service-specifik lookup" eller fallback til EU, hvis intet regionalt regelsæt findes. ([sncf-connect.com](#))

- **Renfe** har **produkt-afhængige** tabeller (AVE, Alvia, Euromed, Avant, Media Distancia m.fl.). Opbevar per-linje/per-produkt mapping; Renfe's engelske side opsummerer – men check altid specifik produktunderside ved tvivl. ([Renfe](#))
- **NS**: reglerne siger 30-min-tærskel, men **beløbet** afhænger af billettypen/abonnement (og der er undtagelser, fx varslede arbejder). Læs vilkår PDF ved udregning.
- **Belgien**: "frequent delay"-ordningen kan give 50% for 30–59 min, men **ikke** for enkeltstående forsinkelser. Sæt en pendler/abonnement-flag i flowet. ([belgiantrain.be](#))

Hvis du vil, laver jeg en **CSV** "national\_overrides.csv" med kolonnerne:

```
country, operator, product_scope, min_delay_min, tiers_json, payout_notes, source_url
```

...udfyldt for: SNCF TGV/Intercités, Renfe (AVE/udvalgte), Trenitalia Frecce, NS (domestic), DSB (fjern/reg), SJ (inrikes), SNCB (frequent delay). Så kan du loade den direkte som et "override-layer" i din kompensationsmotor.

### **Ad step bevisførelse – samling af relevant info :**

#### **DEL 1 :**

Rigtig godt og præcist spørgsmål — artikel **20, stk. 4** er en **nøglebestemmelse** i den nye jernbaneforordning (EU) 2021/782, fordi den binder **assistance** sammen med **kompensation** og **refusion**.

Lad os gennemgå det grundigt, så du kan bruge det i dit flow (og i din "dependency"-logik).

---

### Artikel 20, stk. 4 – Kernen

**"Jernbanevirksomheder oplyser berørte passagerer om, hvordan de anmoder om bekræftelse på, at toget er forsinket, at forsinkelsen har ført til en ikkeopnået tilslutningsforbindelse, eller at toget er aflyst."**

**Denne bekræftelse finder også anvendelse i forbindelse med de i artikel 19 fastsatte bestemmelser."**

→ dvs. operatøren **skal udstede en bekræftelse** ("confirmation of delay/cancellation/missed connection")

→ den **bruges som bevisgrundlag** for en efterfølgende **kompensations- eller refusionsanmodning**.

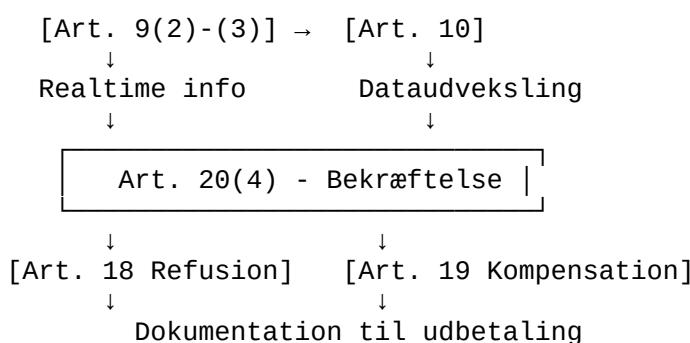
---

## 🔗 Artikler, som **20(4)** hænger sammen med

Artikel	Emne	Sammenhæng / Funktionel afhængighed
<b>Artikel 18</b>	Ret til refusion og omlægning	Hvis passageren vælger <i>refusion</i> (18(1)(a)) eller <i>omlægning</i> (18(1)(b)-(c)), skal der kunne dokumenteres, at toget var <i>forsinket, aflyst eller forbindelsen ikke blev opnået</i> . Bekræftelsen fra <b>20(4)</b> fungerer som <i>dokumentationsbevis</i> for, at kriteriet for art. 18 var opfyldt.
<b>Artikel 19</b>	Kompensation ved forsinkelse	Direkte nævnt i teksten (“denne bekræftelse finder også anvendelse i forbindelse med artikel 19”). Dvs. <b>bekræftelsen er det officielle bevis</b> , der aktiverer kompensationsretten. Appen skal derfor kunne indlæse dette automatisk (fx via RNE-data eller upload).
<b>Artikel 12</b>	Gennemgående billetter	Hvis rejsen består af flere etaper, afgør art. 12, <i>hvem</i> der hæfter for en mistet forbindelse. Bekræftelsen fra art. 20(4) skal derfor knyttes til netop den delstrækning, hvor <i>forbindelsen blev misset</i> — for at afgøre ansvar og om hele rejsen er kompensationsberettiget.
<b>Artikel 9, Oplysninger under stk. 2–3</b>	Oplysninger under stk. 2–3	Operatøren skal løbende give realtidsoplysninger om forsinkelse og forbindelser (bilag II, del II). 20(4)-bekræftelsen er den <i>officielle</i> (juridiske) version af disse oplysninger. Art. 9 giver data → Art. 20(4) formaliserer dokumentet.
<b>Artikel 10</b>	Adgang til trafik- og rejseoplysninger	Hvis art. 10 er fuldt implementeret, kan bekræftelsen (20(4)) sendes automatisk via API til billetudsteder eller app. Hvis art. 10 er undtaget, må bekræftelsen fremskaffes manuelt.
<b>Artikel 30(2)</b>	Oplysninger om passagerrettigheder	Operatøren skal oplyse passageren om <i>hvordan</i> man beder om denne bekræftelse (20(4)). Det er en <b>informationspligt</b> under art. 30(2). Appen kan vise dette som et standardlink eller pop-up.



### Sammenfattet “dependency map”



## 💡 Praktisk betydning for din app

Funktion i app	Relevant artikel	Kommentar
Henter realtidsdata (RNE, operatørfeed)	Art. 9, 10	Datagrundlag for at fastslå forsinkelse
Genererer “Delay Certificate” automatisk	Art. 20(4)	Appen kan generere og tidsstempele

Funktion i app	Relevant artikel	Kommentar
(fx som PDF eller log)		forsinkelsen, hvis operatøren ikke gør det
Vedlægger "Delay Certificate" som bilag til kompensationsformular	Art. 19	Påkrævet for sagsbehandling
Validerer "missed connection" via itinerary matching	Art. 12	Afgør ansvar og berettigelse
Viser info "Sådan anmoder du om bekræftelse"	Art. 30(2)	Oplysningspligt — kan automatiseres i UI

## Eksempel (juridisk og teknisk)

### Eksempel 1: Juridisk

En passager rejser Hamborg → København med DSB/DB og mister forbindelsen i Fredericia. Operatøren udsteder bekræftelse efter **art. 20(4)**:

"Train IC 393 delayed by 74 minutes – missed connection IC 1502 confirmed."

Denne bekræftelse bruges:

- til refusion af billet (art. 18), **eller**
- som bilag til kompensation (art. 19).

### Eksempel 2: I din app

Hvis operatøren ikke udsteder bekræftelse, kan du generere:

"Bekræftelse på forsinkelse: Data registreret via RNE-feed (timestamp, delay 82 min, missed connection confirmed)."

- Appen dokumenterer beviset og logger, at art. 20(4) ikke blev opfyldt af operatøren (non-compliance flag),  
 → men beviset kan alligevel bruges til art. 19-kompensation.

## Opsummering

Artikel	Sammenhæng med art. 20(4)	Funktion
<b>Art. 9(2)- (3)</b>	Forsinkelsesinformation → bekræftelse	Input
<b>Art. 10</b>	Datadistribution mellem aktører	Teknisk adgang
<b>Art. 12</b>	Afgør ansvar ved mistet forbindelse	Juridisk afgrænsning
<b>Art. 18</b>	Ret til refusion/omlægning	Bruger bekræftelsen som bevis
<b>Art. 19</b>	Ret til kompensation	Bruger bekræftelsen som bevis
<b>Art. 30(2)</b>	Informationspligt: skal oplyse om "hvordan man får	Kommunikationslag

Artikel	Sammenhæng med art. 20(4)	Funktion
bekræftelsen"		

---

Vil du have, at jeg laver et *kort “dependency flow”-diagram* (til dit projekt / kodningsflow), som viser art. 20(4)’s relationer og API-hooks — fx hvordan “delay certificate” data kan flyde mellem modulerne (Step Info → Step Refusion → Step Kompensation)?

## Ad step Billetinformationer - art. 9 (del 1-5)

Perfekt — her får du et “cykel-style” spørgesæt til **art. 9, stk. 1** (Bilag II, del I).

Tanken er den samme: **AUTO først** (OCR/PNR + RNE/operatør-API). Kun hvis data mangler eller **mismatch** → vis spørgsmål.

Jeg har lavet det som **copy-paste tekstblokke** til dine diagram-bokse + små “hooks”/logging-nøgler, så Copilot kan hjælpe med implementeringen.

---

## Art. 9(1) — Bilag II, del I (spørgsmål kun ved datamangler/mismatch)

Globalt (visningsregel)

- Vis spørgsmål **kun hvis missing(field) eller mismatch(ticket\_value, api\_value)**.
  - AUTO-kilder: **Billet-OCR/PNR**, RNE live/plan, operatør-API (hvis tilgængeligt).
- 

### Punkt 1) Almindelige betingelser (kontrakt/CoC)

**AUTO:** Link/henvisning til operatørens “Conditions of Carriage” (CIV/CoC).

**Hvis mangler/mismatch, vis:**

1. **“Så du (eller accepterede du) operatørens betingelser ved købet?”**  
– Ja / Nej / Ved ikke  
*hook: coc\_acknowledged*
2. **“Har du et link/pdf til betingelserne, som gjaldt ved købet?”**  
– Upload / Nej  
*hook: coc\_evidence\_upload*
3. **“Stod ‘CIV’ eller henvisning til fælles regler på billetten? (AUTO)”**  
– Ja / Nej / Ved ikke  
*vises kun hvis OCR ikke fandt ‘CIV’*  
*hook: civ\_marking\_present*

**UI-hjælp (kontekstlinje):**

“Vi bruger dette som **kontraktgrundlag** (Art. 4 + Bilag I). Hvis det mangler, styrker det ikke sagen i sig selv, men vi logger det som bevis.”

---

### Punkt 2) Køreplaner & betingelser for hurtigste rejse

**AUTO:** Planlagt afgang/ankomst + “hurtigste rejse” fra RNE/operatør.

**Hvis mangler/mismatch, vis:**

1. “Var den viste rejse ved købet markeret som ‘hurtigste’ eller ‘anbefalet’?”
  - Ja / Nej / Ved ikke

*hook: fastest\_flag\_at\_purchase*
2. “Var minimumsskiftetiden realistisk på den station, hvor du missede forbindelsen?”
  - Ja / Nej / Ved ikke

*hook: mct\_realistic*
3. “Så du alternative forbindelser ved købet?”
  - Ja, flere / Kun få / Nej / Ved ikke

*hook: alts\_shown\_precontract*

**UI-hjælp:**

“Hvis **urimelige skiftetider** blev solgt som realistiske, kan ansvar vurderes strengere (indirekte relevant for Art. 12 + Art. 18/19).”

---

## Punkt 3) Alle tilgængelige billetpriser (med fremhævelse af billigste)

**AUTO:** Pris/fareklasse fra billetten; prissammenligning kun hvis operatør-API tillader.

**Hvis mangler/mismatch, vis:**

1. “Fik du vist flere prisvalg for samme afgang?”
  - Ja (flere) / Nej / Ved ikke

*hook: multiple\_fares\_shown*
2. “Var ‘billigste pris’ markeret eller anbefalet?”
  - Ja / Nej / Ved ikke

*hook: cheapest\_highlighted*
3. “Vælg din købstype (AUTO):”
  - Standard/Non-flex / Semi-flex / Flex / Abonnement/Periodekort / Andet

*hook: fare\_flex\_type*
4. “Gælder billetten kun for tognummer X (AUTO) eller ‘any train that day’?”
  - Kun specifikt tog / Vilkårlig afgang samme dag / Ved ikke

*hook: train\_specificity*

**UI-hjælp:**

“Dette påvirker bl.a. **refundering/omlægning** (Art. 18) og beregningen (Art. 19(3) ved retur/segmenter).”

---

## Punkt 4) Tilgængelighed & assistance (PMR)

**AUTO:** Operatørens PMR-politik, assistancevinduer, tilgængelige faciliteter (fra TAF/TAP-TSI data / operatørside).

**Hvis mangler/mismatch, vis:**

1. “**Har du et handicap eller nedsat mobilitet, som krævede assistance?**”
  - Ja / Nej

*hook:* pmr\_user
2. “**Bestilte du assistance før rejsen? (AUTO, hvis billet/PNR viser det)**”
  - Ja (dato/tid) / Nej / Forsøgte men fik afslag

*hook:* pmr\_booked
3. “**Blev den bestilte assistance faktisk leveret?**”
  - Ja, fuldt / Delvist / Nej

*hook:* pmr\_delivered\_status
4. “**Manglede der PMR-faciliteter, som var lovet før købet?** (ramper, lift, platforminfo)\*\*”
  - Ja (beskriv) / Nej / Ved ikke

*hook:* pmr\_promised\_missing

**UI-hjælp:**

“PMR-info er en **forud-oplysningspligt** (Art. 9(1)). Manglende/fejl kan indgå som **downgrade-bevis** i refusion/assistance (Art. 18/20).”

---

## Punkt 5) Cykelkapacitet & vilkår (*integreret med din cykelflowboks*)

**AUTO:** Fra OCR (cykelikon/reservation) + RNE/operatør

**Vis kun ved datamangler/mismatch:**

1. “**Var cyklen omfattet af billet/reservation? (AUTO)**”
  - Ja, separat cykelreservation
  - Ja, inkluderet i billetten
  - Nej, ingen reservation krævet

*hook:* bike\_reservation\_type
2. “**Krævede denne afgang cykelreservation? (AUTO fra RNE)**”
  - Ja / Nej / Ukendt

*hook:* bike\_res\_required
3. “**Blev cyklen afvist ved ombordstigning?**”
  - Ja, uden begrundelse
  - Ja, begrundet (manglende plads/sikkerhed/teknisk)
  - Nej

*hook:* bike\_denied\_reason
4. “**Hvis afvist:** Fik du omlægning/refusion tilbuddt?\*\* (Art. 18)\*\*”
  - Omlægning / Refusion / Intet

*hook:* bike\_followup\_offer
5. “**Forsinkelse på ankomst pga. cykelhåndtering?**”
  - <60 / 60–119 / ≥120 min

*hook:* bike\_delay\_bucket

### Kontekstlinje (auto-hint):

“Hvis cykelplads var **reserveret/inkluderet** men afvist **uden behørig grund**, kan du have krav på **omlægning/refusion (Art. 18)**, **assistance (Art. 20(2))** og **kompensation (Art. 19)**.”

---

## “Vis/skip”-regler (kort)

- **Punkt 1–4:** vis *kun* hvis `missing/mismatch` på de respektive felter.
  - **Punkt 5 (cykel):** vis *kun* hvis `pm_bike_involved == True` ELLER `ticket_bike_hint == True` ELLER `mismatch(bike_rules)`.
  - **PMR-blok (pkt. 4):** vis *kun* hvis `pmr_user == True` ELLER `pmr_booking_hint == True` ELLER `missing(pmr_info)`.
- 

## Små UI-tekstblokke (kontekst/beskeder)

### Mismatch (hurtigste rejse):

“Systemet viser, at der fandtes en hurtigere forbindelse ved købet. Bekræft venligst, hvad der blev vist for dig.”

### Pre-info om forsinkelse (Art. 19(9)):

“Hvis forsinkelsen var oplyst før køb, er der **ingen EU-kompensation**. Bekræft venligst, om du så en sådan meddelelse.”

### Cykelreservation mangler:

“Operatørens data indikerer krav om cykelreservation. Uden reservation kan transport lovligt afvises.”

### PMR-assistance ikke leveret:

“Bestilt PMR-assistance, der ikke blev leveret, registreres som bevis og kan udløse refusion/assistancekrav.”

---

## Logging-nøgler (til Copilot-hooks)

- `coc_acknowledged`, `coc_evidence_upload`, `civ_marking_present`
- `fastest_flag_at_purchase`, `mct_realistic`, `alts_shown_precontract`
- `multiple_fares_shown`, `cheapest_highlighted`, `fare_flex_type`, `train_specificity`
- `pmr_user`, `pmr_booked`, `pmr_delivered_status`, `pmr_promised_missing`
- `bike_reservation_type`, `bike_res_required`, `bike_denied_reason`, `bike_followup_offer`, `bike_delay_bucket`

# Mini-JSON (eksempel) til din form-renderer

```
{  
    "art9_precontract": {  
        "1_contract_terms": [  
            {"id": "coc_acknowledged", "ask_if": "missing(coc_link)||mismatch(coc)", "q": "Så du/accept. betingelserne ved købet?", "type": "single", "opts": ["Ja", "Nej", "Ved ikke"]},  
            {"id": "coc_evidence_upload", "ask_if": "missing(coc_link)", "q": "Upload link/pdf til betingelserne (hvis muligt)", "type": "file"},  
            {"id": "civ_marking_present", "ask_if": "missing(civ_mark)", "q": "Så du 'CIV' på billetten?", "type": "single", "opts": ["Ja", "Nej", "Ved ikke"]}  
        ],  
        "2_fastest_journey": [  
            {"id": "fastest_flag_at_purchase", "ask_if": "mismatch(fastest)", "q": "Var rejsen markeret 'hurtigste'?", "type": "single", "opts": ["Ja", "Nej", "Ved ikke"]},  
            {"id": "mct_realistic", "ask_if": "missing(mct)||mismatch(mct)", "q": "Var skiftetiden realistisk?", "type": "single", "opts": ["Ja", "Nej", "Ved ikke"]},  
            {"id": "alts_shown_precontract", "ask_if": "missing(alts)", "q": "Så du alternative forbindelser?", "type": "single", "opts": ["Ja, flere", "Kun få", "Nej", "Ved ikke"]}  
        ],  
        "3_fares": [  
            {"id": "multiple_fares_shown", "ask_if": "missing(fares_shown)", "q": "Fik du vist flere prisvalg?", "type": "single", "opts": ["Ja", "Nej", "Ved ikke"]},  
            {"id": "cheapest_highlighted", "ask_if": "missing(cheapest)", "q": "Var 'billigste' fremhævet?", "type": "single", "opts": ["Ja", "Nej", "Ved ikke"]},  
            {"id": "fare_flex_type", "ask_if": "missing(fare_type)", "q": "Vælg din købstype", "type": "single", "opts": ["Non-flex", "Semi-flex", "Flex", "Abonnement", "Andet"]},  
            {"id": "train_specificity", "ask_if": "missing(train_specificity)", "q": "Kun specifikt tog eller vilkårlig afgang?", "type": "single", "opts": ["Specifikt tog", "Vilkårlig samme dag", "Ved ikke"]}  
        ],  
        "4_pmr": [  
            {"id": "pmr_user", "ask_if": "missing(pmr_user)||hint(pmr)", "q": "Har du behov for PMR-assistance?", "type": "single", "opts": ["Ja", "Nej"]},  
            {"id": "pmr_booked", "ask_if": "pmr_user==Ja && (missing(pmr_booking)||mismatch(pmr_booking))", "q": "Bestilte du assistance?", "type": "single", "opts": ["Ja (angiv)", "Nej", "Forsøgte - afvist"]},  
            {"id": "pmr_delivered_status", "ask_if": "pmr_user==Ja && !missing(pmr_booking)", "q": "Blev assistancen leveret?", "type": "single", "opts": ["Ja, fuldt", "Delvist", "Nej"]},  
            {"id": "pmr_promised_missing", "ask_if": "pmr_user==Ja && missing(pmr_facilities)", "q": "Manglede lovede PMR-faciliteter?", "type": "text"}  
        ],  
        "5_bikes": [  
            {"id": "bike_reservation_type", "ask_if": "hint(bike)||mismatch(bike)", "q": "Var cyklen omfattet af billet/reservation?", "type": "single", "opts": ["Separat reservation", "Inkluderet i billetten", "Ingen reservation"]},  
            {"id": "bike_res_required", "ask_if": "mismatch(bike_res_required)||missing(bike_res_required)", "q": "Krævede afgang  
cykelreservation?", "type": "single", "opts": ["Ja", "Nej", "Ukendt"]},  
            {"id": "bike_denied_reason", "ask_if": "pm_bike_involved==true", "q": "Blev cyklen afvist?", "type": "single", "opts": ["Ja, uden begr.", "Ja, begrundet", "Nej"]},  
            {"id": "bike_followup_offer", "ask_if": "bike_denied_reason startswith 'Ja'", "q": "Fik du omlægning/refusion?", "type": "single", "opts": ["Omlægning", "Refusion", "Intet"]},  
        ]  
    }  
}
```

```
{"id":"bike_delay_bucket","ask_if":"bike_denied_reason!='Nej' ||  
hint(bike_delay)","q":"Forsinkelse på ankomst pga.  
cykel?","type":"single","opts":["<60","60-119","≥120"]}  
]  
}  
}
```

---

## Hvorfor dette hjælper din kompensationsdel

- **Art. 19(9):** vi fanger “forud oplyst forsinkelse” via pkt. 7 i del I (i praksis fra pkt. 2+7), så kompensation kan udelukkes korrekt.
- **Art. 18/20:** PMR- og cykel-info bruges til at dokumentere **assistance/refusion/omlægning** og evt. **downgrade**.
- **Art. 12:** svar i pkt. 2–3 understøtter vurderingen af **gennemgående billet** vs. særskilte kontrakter (realistiske skift, “fastest”, mv.), som påvirker **beregningen**.

Vil du have, at jeg også laver tilsvarende “fallback-spørgsmål” for **Bilag II del I, pkt. 6–10** i samme format (med små JSON-hooks), så hele art. 9(1) er pakket ind?

## Art. 9 del 6-10

Perfekt—her er “cykel-style” **fallback-spørgsmål for Bilag II, del I, pkt. 6–10 under art. 9(1).**  
Samme princip: **AUTO først** (OCR/PNR + RNE/operatør-API). Kun hvis data **mangler** eller **mismatch** → vis spørgsmål. Jeg giver også hooks (log-nøgler) og en lille JSON-skelet til din formrenderer.

---

## Visningsregel (global)

- Vis **kun** hvis `missing(field)` **eller** `mismatch(ticket_value, api_value)`.
  - AUTO-kilder: billet-OCR/PNR, RNE/TAF-TAP, operatør-site/API.
- 

## Pkt. 6) Pladser/klasser, ligge-/sovevogne

**AUTO:** Klasse (1./2.), pladsres., kupe/ligge/sove, coach/seat.

Hvis mangler/mismatch:

1. **“Hvilken klasse var købt? (AUTO)”**
  - 1. klasse / 2. klasse / Andet / Ved ikke

*hook: fare\_class\_purchased*
2. **“Fik du den klasse, du betalte for?”**
  - Ja / Nej, nedklassificeret / Nej, opgraderet

*hook: class\_delivered\_status*
3. **“Var der reserveret plads/kupe/ligge/sove? (AUTO)”**
  - Fast sæde / Fri plads / Liggevogn / Sovevogn / Ingen

*hook: berth\_seat\_type*
4. **“Blev reserveret plads/ligge/sove leveret?”**
  - Ja / Nej / Delvist

*hook: reserved\_amenity\_delivered*

Bruges til **downgrade-vurdering** i refusion/assistance (Art. 18) og evt. komp. (Art. 19).

---

## Pkt. 7) Afbrydelser/forsinkelser (planlagte & realtid)

**AUTO:** Disruption feed (planlagte works) + realtime forsinkelser.

Hvis mangler/mismatch:

1. **“Var der meddelt afbrydelse/forsinkelse før dit køb?”**
  - Ja / Nej / Ved ikke

*hook: preinformed\_disruption*
2. **“Hvis Ja: Hvor blev det vist?”**
  - Rejseplan / Operatør-site/app / Billetoverblik / Andet (tekst)

*hook: preinfo\_channel*

3. “**Så du realtime-opdateringer under rejsen?**”

– Ja, i app / Ja, i toget / Ja, på station / Nej

*hook: realtime\_info\_seen*

Kritisk for **Art. 19(9)** (ingen kompenstation hvis oplyst før køb) og til bevis for **missed connection**.

---

## Pkt. 8) Faciliteter i toget (Wi-Fi, toilet, service/assistance)

**AUTO:** Operatørens togprofil for afgang.

**Hvis mangler/mismatch:**

1. “**Hvilke faciliteter var lovet? (vælg)**”

– Wi-Fi / Toilet / Strøm / Servering / Familiezone / PMR-hjælp / Andet

*hook: promised\_facilities[]*

2. “**Hvad fik du faktisk?**”

– Samme / Kun delvist / Slet ikke

*hook: facilities\_delivered\_status*

3. “**Manglede en lovet facilitet og påvirkede det rejsen? (beskriv kort)**”

– Tekstfelt

*hook: facility\_impact\_note*

Bruges som **downgrade-bevis** i refusion/assistance.

---

## Pkt. 9) Oplysning før køb om gennemgående billet

**AUTO:** Betragt booking flow + billetmarkup (PNR-scope) + operatør/OTA attributter.

**Hvis mangler/mismatch:**

1. “**Blev du tydeligt informeret om, at din(e) billet(ter) var: ...?**”

– **Gennemgående billet** (én kontrakt)

– **Særskilte kontrakter** (samlet køb, men flere billetter)

– Ved ikke

*hook: through\_ticket\_disclosure*

2. “**Købte du alt i én transaktion hos operatøren? (AUTO)**”

– Ja / Nej / Ved ikke

*hook: single\_txn\_operator*

3. “**Eller købte du i én transaktion hos billetudsteder/rejsebureau? (AUTO)**”

– Ja / Nej / Ved ikke

*hook: single\_txn\_retailer*

4. “**Stod der udtrykkeligt, at billetterne var særskilte kontrakter?**”

– Ja / Nej / Ved ikke

*hook: separate\_contract\_notice*

Direkte til **Art. 12(2–5)** og dermed ansvarsplacering + beregning (Art. 19(3)).

---

## Pkt. 10) Klageprocedurer

**AUTO:** Operatørens officielle klagekanal (mail/webform), response SLA, NEB-link.

Hvis mangler/mismatch:

1. **“Så du info om, hvor klagen skulle indsendes?”**  
– Ja (operatørens formular/mail) / Nej / Ved ikke  
*hook: complaint\_channel\_seen*
2. **“Har du allerede indsendt en klage?”**  
– Ja (upload kvittering) / Nej  
*hook: complaint\_already\_filed + complaint\_receipt\_upload*
3. **“Ønsker du, at vi indsender via operatørens officielle kanal på dine vegne?”**  
– Ja / Nej  
*hook: submit\_via\_official\_channel*

Hjælper både **Art. 30** (info) og procesflow (din app som proxy).

---

## Mini-JSON skelet (pkt. 6–10)

```
{  
  "art9_precontract_ext": {  
    "6_class_berths": [  
      {"id": "fare_class_purchased", "ask_if": "missing(fare_class)||mismatch(fare_class)", "q": "Hvilken klasse var købt?", "type": "single", "opts": ["1. klasse", "2. klasse", "Andet", "Ved ikke"]},  
      {"id": "class_delivered_status", "ask_if": "missing(class_delivered)||mismatch(class_delivered)", "q": "Fik du den klasse, du betalte for?", "type": "single", "opts": ["Ja", "Nedklassificeret", "Opgraderet"]},  
      {"id": "berth_seat_type", "ask_if": "missing(berth_seat)||mismatch(berth_seat)", "q": "Var der reserveret plads/ligge/sove?", "type": "single", "opts": ["Fast sæde", "Fri plads", "Liggevogn", "Sovevogn", "Ingen"]},  
      {"id": "reserved_amenity_delivered", "ask_if": "berth_seat_type!=Ingen", "q": "Blev reserveret facilitet leveret?", "type": "single", "opts": ["Ja", "Delvist", "Nej"]}  
    ],  
    "7_disruptions": [  
      {"id": "preinformed_disruption", "ask_if": "missing(preinfo)||mismatch(preinfo)", "q": "Var afbrydelse/forsinkelse meddelt før køb?", "type": "single", "opts": ["Ja", "Nej", "Ved ikke"]},  
      {"id": "preinfo_channel", "ask_if": "preinformed_disruption=='Ja'", "q": "Hvor blev det vist?", "type": "multi", "opts": ["Rejseplan", "Operatør-app/website", "Billetoverblik", "Andet"]},  
    ]  
  }  
}
```

```

        {"id":"realtime_info_seen","ask_if":"missing(rtinfo_seen)","q":"Så du
realtime-opdateringer under rejsen?","type":"multi","opts":["App","I toget","På
station","Nej"]}

    ],
    "8_facilities_onboard": [
        {"id":"promised_facilities","ask_if":"missing(facilities_promised)||
mismatch(facilities)","q":"Hvilke faciliteter var lovet?","type":"multi","opts":
["Wi-Fi","Toilet","Strøm","Servering","PMR-hjælp","Andet"]},
        {"id":"facilities_delivered_status","ask_if":)!
empty(promised_facilities),"q":"Hvad fik du faktisk?","type":"single","opts":
["Samme","Kun delvist","Slet ikke"]},
        {"id":"facility_impact_note","ask_if":facilities_delivered_status!
='Samme',"q":"Kort beskrivelse af konsekvens","type":"text"}
    ],
    "9_through_ticket_info": [
        {"id":"through_ticket_disclosure","ask_if":"missing(through_disclosure)||
mismatch(through)","q":"Blev du informeret om gennemgående vs. særskilte
billetter?","type":"single","opts":["Gennemgående","Særskilte kontrakter","Ved
ikke"]},
        {"id":"single_txn_operator","ask_if":missing(single_txn_op),"q":"Købte
du alt i én transaktion hos operatøren?","type":"single","opts":["Ja","Nej","Ved
ikke"]},
{"id":"single_txn_retailer","ask_if":missing(single_txn_retailer),"q":"Eller i
én transaktion hos billetudsteder/rejsebureau?","type":"single","opts":
["Ja","Nej","Ved ikke"]},
        {"id":"separate_contract_notice","ask_if":through_ticket_disclosure!
='Gennemgående',"q":"Stod det udtrykkeligt, at billetterne var særskilte
kontrakter?","type":"single","opts":["Ja","Nej","Ved ikke"]}
    ],
    "10_complaints": [
{"id":"complaint_channel_seen","ask_if":missing(complaint_channel),"q":"Så du
info om klagekanal?","type":"single","opts":["Ja","Nej","Ved ikke"]},
        {"id":"complaint_already_filed","ask_if":complaint_channel_seen!
='Nej',"q":"Har du allerede indsendt en klage?","type":"single","opts":["Ja
(upload kvittering)","Nej"]},
        {"id":"complaint_receipt_upload","ask_if":complaint_already_filed
startswith 'Ja',"q":"Upload kvittering/dok.","type":"file"},

{"id":"submit_via_official_channel","ask_if":complaint_already_filed=='Nej'',"q
":"Ønsker du, at vi indsender via officiel kanal?","type":"single","opts":
["Ja","Nej"]}
    ]
}

```

}

---

## Små UI-tekstblokke (kan indsættes i bokse)

- **Pre-info låser kompensation:**

“Hvis forsinkelsen var oplyst før køb, gælder **ingen EU-kompensation** (Art. 19(9)). Vi logger derfor, hvor du så beskeden.”

- **Nedklassificeret plads/berth:**

“Nedklassificering eller manglende reserveret facilitet kan udløse **refusion/assistance** (Art. 18/20) og påvirke kompensationen.”

- **Gennemgående vs. særskilt:**

“Hvis du ikke blev tydeligt informeret om ‘særskilte kontrakter’, kan sælger få **udvidet ansvar** (Art. 12(4–6)).”

- **Klagekanal:**

“Vi kan indsende via operatørens **officielle klagekanal** (Art. 30), så du får timestamp og korrekt sporbarhed.”

---

## Hooks (log-nøgler) samlet

```
fare_class_purchased, class_delivered_status, berth_seat_type,  
reserved_amenity_delivered, preinformed_disruption,  
preinfo_channel, realtime_info_seen, promised_facilities[],  
facilities_delivered_status, facility_impact_note,  
through_ticket_disclosure, single_txn_operator,  
single_txn_retailer, separate_contract_notice,  
complaint_channel_seen, complaint_already_filed,  
complaint_receipt_upload, submit_via_official_channel
```

---

Vil du have, at jeg **fletter disse direkte** ind i din seneste PDF/diagramtekst som færdige bokse (samme stil som cykeldelen), så du kan copy-paste uden omskrivning?

## Step Billetinformationer – art. 12

Fremragende — vi tager **artikel 12** (“gennemgående billet” og “særskilte befordringskontrakter”) og bygger den op på **samme smarte, automatiske flow** som cykeldelen og bilag II-delen, men med tydelig fokus på:

- at **auto-tjekke** via billet/PNR-data og RNE,
- og **kun spørge brugeren**, hvis oplysningerne mangler eller er i konflikt.

Nedenfor får du:

- En logisk spørgestruktur opdelt i sektioner
- UI-tekstblokke til kontekst
- Hooks (feltnavne til logging/automatisering)
- Et JSON-skelet til direkte form-integration
- Dependencies-noter (hvordan det påvirker kompensationsberegnning)

---

## Artikel 12 — gennemgående billet / særskilte kontrakter

### Auto-grundlag

- OCR / PNR: antal kontrakter, transaktions-ID'er, operatør vs. rejsebureau
- RNE/TAF-data: sammenhæng i ruteplan, connection-time, ticket-scope
- OTA-/rejsebureau-flag: multi-segment bookings

Vis spørgsmål **kun hvis missing()** eller **mismatch()**.

---

### ◆ 1. Grundtype: gennemgående eller særskilte kontrakter

(Bilag II, del I, pkt. 9 + Art. 12 stk. 2–3)

**[1] “Var du tydeligt informeret om, at din(e) billet(ter) var ...?”**

- Gennemgående billet (én kontrakt)
- Særskilte kontrakter (samlet køb, men flere billetter)
- Ved ikke

*hook: through\_ticket\_disclosure*

**[2] “Købte du alle billetter i én transaktion hos operatøren?” (AUTO)**

- Ja / Nej / Ved ikke

*hook: single\_txn\_operator*

**[3] “Eller købte du samlet hos et rejsebureau eller billetudsteder?” (AUTO)**

- Ja / Nej / Ved ikke

*hook: single\_txn\_retailer*

**4 “Stod der udtrykkeligt, at billetterne var særskilte kontrakter?”**

– Ja / Nej / Ved ikke

*hook: separate\_contract\_notice*

**5 “Var alle billetter udstedt med samme ordrenummer / PNR?” (AUTO)**

– Ja / Nej / Ved ikke

*hook: shared\_pnr\_scope*

❖ **Afhængighed:** Hvis `through_ticket_disclosure == "Gennemgående"`

→ ansvar følger **Art. 12(4)** → operatøren dækker hele rejsen.

Hvis "Særskilte" men `shared_pnr_scope == Ja`

→ potentiel "skjult gennemgående billet" (Art. 12(5)).

---

◆ 2. Hvem solgte rejsen?

(Art. 12 stk. 2 + 3)

**6 “Var det en jernbanevirksomhed, der solgte rejsen?” (AUTO)**

– Ja / Nej / Ved ikke

*hook: seller\_type\_operator*

**7 “Eller et rejsebureau / billetudsteder?”**

– Ja / Nej / Ved ikke

*hook: seller\_type\_agency*

**8 “Var flere jernbanevirksomheder involveret i transporten?” (AUTO)**

– Ja / Nej / Ved ikke

*hook: multi\_operator\_trip*

❖ **Afhængighed:**

- Hvis **rejsebureau** → ansvar efter **Art. 12(5)** (de hæfter, hvis ikke informeret korrekt).
  - Hvis **flere operatører** og ingen tydelig information → "implicit gennemgående billet".
- 

◆ 3. Forbindelses- og tidsforhold

(Art. 12 stk. 4 + Bilag II del I pkt. 2)

**9 “Var skiftetider mellem tog realistiske?”**

– Ja / Nej / Ved ikke

*hook: mct\_realistic*

**10 “Viste købet, at forbindelserne var dækket af ét ansvar (én rejseplan)?”**

– Ja / Nej / Ved ikke

*hook: one\_contract\_schedule*

❖ Bruges ved **beregning af kompensation** (Art. 19 stk. 3) — afgør om forsinkelse i første tog dækker hele rejsen.

---

## ◆ 4. Kommunikations- og ansvars-info

(Art. 12 stk. 6–7 + Art. 30)

1[1] “Fik du oplyst, hvem du skulle kontakte ved forsinkelse eller aflysning?”

– Ja / Nej / Ved ikke

hook: contact\_info\_provided

1[2] “Var det klart, hvem der havde ansvaret for rejsen, hvis du missede forbindelsen?”

– Ja / Nej / Ved ikke

hook: responsibility\_explained

1[3] “Fik du ét bekræftelses-/booking-nummer for hele rejsen?” (AUTO)

– Ja / Nej / Ved ikke

hook: single\_booking\_reference

## ◆ 5. Undtagelser og sammenhæng

AUTO: tjek i din “exemption matrix” (Art. 2).

Hvis route / service er fritaget fra Art. 12 → vis:

1[4] “Denne afgang er undtaget fra reglerne om gennemgående billet. Ønsker du alligevel at fortsætte beregningen ud fra nationale regler?”

– Ja / Nej

hook: exemption\_override\_12



## Dependencies til beregning og kompensation

Reference	Afhængighed	Effekt
Art. 12 (4–5)	through_ticket_disclosure, shared_pnr_scope, seller_type_agency	afgør <b>hvem</b> hæfter ved forsinkelse og <b>om</b> <b>kompensation</b> gælder hele rejsen
Bilag II I.9 → Art. 19 (3)	sammen med pkt. 9–10 ovenfor	bruges i <b>Step Udregning</b> <b>kompensation</b>
Art. 12 (6–7) + Art. 30	contact_info_provided, responsibility_explained	skal vises i app som <b>oplysningsmodul</b>



## UI-tekstblokke (til brug i diagrammet)

### ! Uklar billettype

“Systemet kan ikke fastslå, om din billet er gennemgående eller flere særskilte kontrakter. Besvar venligst spørgsmålene her — det påvirker, hvem der skal udbetale kompensation.”

### 💡 Én transaktion, flere billetter

“Selvom du købte alle billetter på én gang, kan det stadig være særskilte kontrakter. Hvis det ikke blev tydeligt oplyst, hæfter sælger efter art. 12 (5).”



## Flere operatører

“Hvis flere operatører indgår i samme køb uden klar information, anses rejsen som gennemgående billet — og hele forsinkelsen kan give kompensation.”



## Kontakt- og ansvarsinfo

“Manglende information om, hvem der har ansvaret ved afbrydelse, registreres som dokumentationsfejl og styrker din sag.”



# JSON-skelet

```
{
  "art12_check": {
    "1_ticket_type": [
      {"id": "through_ticket_disclosure", "ask_if": "missing(through_disclosure) || mismatch(through)", "q": "Var du tydeligt informeret om billettypen?", "type": "single", "opts": ["Gennemg\u00e5ende billet", "S\u00e5rskilte kontrakter", "Ved ikke"]},
      {"id": "single_txn_operator", "ask_if": "missing(single_txn_op)", "q": "K\u00f8bte du alt i \u00e9n transaktion hos operatoren?", "type": "single", "opts": ["Ja", "Nej", "Ved ikke"]},
      {"id": "single_txn_retailer", "ask_if": "missing(single_txn_retailer)", "q": "K\u00f8bte du hos rejsebureau/billetudsteder?", "type": "single", "opts": ["Ja", "Nej", "Ved ikke"]},
      {"id": "separate_contract_notice", "ask_if": "through_ticket_disclosure != 'Gennemg\u00e5ende billet'", "q": "Stod det udtrykkeligt, at billetterne var s\u00e5rskilte kontrakter?", "type": "single", "opts": ["Ja", "Nej", "Ved ikke"]},
      {"id": "shared_pnr_scope", "ask_if": "missing(pnr_scope) || mismatch(pnr_scope)", "q": "Var alle billetter udstedt med samme ordrenummer/PNR?", "type": "single", "opts": ["Ja", "Nej", "Ved ikke"]}
    ],
    "2_seller_type": [
      {"id": "seller_type_operator", "ask_if": "missing(seller_type) || mismatch(seller_type)", "q": "Var det en jernbanevirksomhed, der solgte rejsen?", "type": "single", "opts": ["Ja", "Nej", "Ved ikke"]},
      {"id": "seller_type_agency", "ask_if": "seller_type_operator != 'Ja'", "q": "Eller et rejsebureau/billetudsteder?", "type": "single", "opts": ["Ja", "Nej", "Ved ikke"]},
      {"id": "multi_operator_trip", "ask_if": "missing(multi_operator) || mismatch(multi_operator)", "q": "Var flere operatører involveret?", "type": "single", "opts": ["Ja", "Nej", "Ved ikke"]}
    ],
    "3_connection_conditions": [
      {"id": "mct_realistic", "ask_if": "missing(mct) || mismatch(mct)", "q": "Var skiftetider realistiske?", "type": "single", "opts": ["Ja", "Nej", "Ved ikke"]}]
  }
}
```

```

        {"id":"one_contract_schedule","ask_if":"missing(one_contract)||mismatch(one_contract)","q":"Var forbindelserne samlet i én rejseplan?","type":"single","opts":["Ja","Nej","Ved ikke"]}

    ],
    "4_contact_responsibility": [
        {"id":"contact_info_provided","ask_if":"missing(contact_info)||mismatch(contact_info)","q":"Fik du oplyst, hvem du skulle kontakte ved forsinkelse?","type":"single","opts":["Ja","Nej","Ved ikke"]},
        {"id":"responsibility_explained","ask_if":"missing(resp_info)||mismatch(resp_info)","q":"Var det klart, hvem der havde ansvaret, hvis du missede forbindelsen?","type":"single","opts":["Ja","Nej","Ved ikke"]},
        {"id":"single_booking_reference","ask_if":"missing(booking_ref)||mismatch(booking_ref)","q":"Fik du ét bookingnummer for hele rejsen?","type":"single","opts":["Ja","Nej","Ved ikke"]}
    ],
    "5_exemptions": [
        {"id":"exemption_override_12","ask_if":"route_exempt==true","q":"Denne afgang er undtaget fra Art.12. Ønsker du at fortsætte med nationale regler?","type":"single","opts":["Ja","Nej"]}]
    ]
}
}

```

## Dependencies

- **Art. 12 (4–5)** → `through_ticket_disclosure`, `shared_pnr_scope`, `seller_type_agency`  
→ bruges i kompensationsberegning (**Step Udregning kompensation**)
- **Bilag II I.9 + Art. 19 (3)** → `mct_realistic`, `one_contract_schedule`  
→ vurder om hele rejseforløbet udløser kompensation
- **Art. 12 (6–7)** → `contact_info_provided`, `responsibility_explained`  
→ kobles til **Art. 30-information** (vises som hjælpetekst i app'en)

Vil du have, at jeg laver **samme struktur (auto-/spørgsmålsflow + JSON-hooks)** for **Art. 18–20** bagefter, så du kan koble hele refundering- og assistancedelen op på én samlet logik?

## Step Billetinformationer - Art. 19 kompensation

Fantastisk—her får du en “plug-and-play” kompensationspakke til **artikel 19** (ny forordning 2021/782), i samme stil som dine andre moduler:

- et **beregningsflow** (logik + edge cases)
- **fallback-spørgsmål** til brugerne (kun ved manglende data/mismatch)
- **hooks/feltnavne** til logging (så Copilot kan kode det direkte)
- et **mini-JSON-skelet** til din form-renderer
- korte **UI-tekstblokke** til kontekst

Alt er tænkt til at spille sammen med dine moduler for **Art. 12 (gennemgående)**, **Art. 9/10 (data)** og **exemptions-matrixen**.

---

## 1) Kompensations-beregning (Art. 19)

### Grundregler

- **Tærskel:** endelig **ankomstforsinkelse** (ikke afgang)  $\geq 60 \text{ min}$   
→ 25 % (60–119 min) / 50 % ( $\geq 120 \text{ min}$ ) af **prisgrundlaget**.
- **Ingen komp**, hvis:
  - **refusion** er valgt i stedet (Art. 18) for samme tjeneste, eller
  - **forsinkelsen var oplyst før køb** (Art. 19(9)), eller
  - **omlægning** gjorde endelig forsinkelse  $< 60 \text{ min}$  (Art. 19(9)).

### Prisgrundlag (Art. 19(3))

- Enkel enkeltbillet/segment: brug **pris betalt for den forsinkede tjeneste**.
- Retur: komp for den forsinkede **del**; hvis delpriser ikke fremgår →  $\frac{1}{2}$  af samlet pris.
- Flere etaper i **én befordiningskontrakt (gennemgående)**: brug **fuld pris** for kontrakten, men kun hvis forsinkelsen påvirker **ankomsten til endeligt bestemmessted** i den kontrakt.
- **Separate kontrakter**: beregn pr. kontrakt (Art. 12-modulet afgør).

### EU/ikke-EU (Art. 19(4) + præambel (22))

- Ignorér forsinkelser **opstået uden for EU** i selve **forsinkelsesminutterne**.  
→ Prisgrundlaget ændres ikke af dette—det er **forsinkelsesmålingen**, der filtreres.
- Gennemgående billet med EU+ikke-EU: medregn **kun den del af forsinkelsen, som er opstået i EU**.

## Undtagelser (din matrix)

- Hvis **Art. 19 (helt) eller (a)/(b)** er **fritaget** i en medlemsstat/tjeneste:
  - **Disable** kompensationsberegning for de **berørte segmenter/kontrakter**.
  - Hvis rejsen også indeholder **ikke-fritagte** dele (f.eks. internationalt/anden stat), beregn **kun** for de dele, hvor Art. 19 gælder.

## Operatørens undskyldende årsager (Art. 19(10))

- **Ekstraordinære forhold** (vejr, store naturkatastrofer, større folkesundhedskriser) → komp kan afvises.
- **Egen skyld hos passager** → komp kan afvises (kobles til dit Refusion-modul/CIV).
- **Tredjepart** (fx personer på spor, kabeltyveri, nødsituation, politiindsats, sabotage, terror) → komp kan afvises.  
**Men:** **strejker** hos jernbanevirksomhedens eget personale, handlinger/undladelser fra andre virksomheder på samme infrastruktur, infrastrukturforvaltere og stationsledere **er ikke** omfattet af tredjeparts-undtagelsen → **komp beholdes**.

## Udbetaling & form (Art. 19(7),(8))

- Udbetal **senest 1 måned** efter ansøgning.
- Vouchers må kun bruges, hvis **fleksible; kontant udbetaling** på passagerens **anmodning**.
- Ingen fradrag for transaktionsudgifter; tilladt **minimumsgrænse**  $\leq 4$  EUR pr. billet.

---

## 2) Databehov (AUTO) + fallback til bruger

Din pipeline (AUTO først):

1. **Billet/PNR-OCR** → pris, PNR-scope, operatør/retailer, klasse, segmenter, tider.
2. **RNE/operatør-API** → plan & realtid, ankomstforsinkelse, disruptions, segment-geografi (EU?), gennemgående/kontrakt-hint.
3. **Exemptions-matrix** → Art. 19 gælder? delvis?
4. **Art. 12-modul** → gennemgående vs. separate kontrakter → prisgrundlag.

**Kun hvis** data mangler/mismatch → stil spørgsmålene herunder.

---

### 3) Spørgsmål til brugeren (kun ved manglende data/mismatch)

#### A) Forsinkelser & ankomst

##### 1. Hvad var din faktiske ankomsttid på det endelige bestemmelsessted?

– hh:mm (lokal tid)

*hook: actual\_arrival\_time\_final*

##### 2. Hvis du manglede en forbindelse: hvor og hvornår?

– Station + hh:mm

*hook: missed\_connection\_station\_time*

##### 3. Fik du omlægning, så endelig forsinkelse blev < 60 min? (AUTO fra RNE)

– Ja / Nej / Ved ikke

*hook: reroute\_under\_60\_confirmed*

##### 4. Blev forsinkelsen oplyst før købet? (AUTO fra Bilag II I.7/RNE)

– Ja / Nej / Ved ikke

*hook: delay\_preinformed\_before\_purchase*

#### B) Pris & kontrakt (Art. 19(3) + Art. 12)

##### 5. Var der tale om gennemgående billet (én kontrakt) for hele rejsen? (AUTO)

– Ja / Nej / Ved ikke

*hook: through\_ticket\_scope*

##### 6. Hvis retur: står ud- og hjemrejse som særskilte priser på billetten?

– Ja / Nej / Ved ikke

*hook: return\_leg\_priced\_separately*

##### 7. Hvis flere segmenter: er pris pr. segment angivet?

– Ja / Nej

*hook: segment\_prices\_shown*

##### 8. Samlet pris betalt for den forsinkede tjeneste / kontrakt? (AUTO)

– Beløb + valuta

*hook: fare\_amount\_basis*

#### C) EU/ikke-EU-filtrering (Art. 19(4))

##### 9. Indgik der en del af rejsen uden for EU? (AUTO via georute)

– Ja (angiv station/land) / Nej / Ved ikke

*hook: non\_eu\_segment\_present*

##### 10. Hvis Ja: Hvilket segment (tognr./station → station) var uden for EU?

– Tekst

*hook: non\_eu\_segment\_detail*

## D) Ekstraordinære forhold (Art. 19(10))

### 11. Oplyste operatøren, at forsinkelsen skyldtes “ekstraordinære forhold”?

- Ja (vælg: vej/natur/folkesundhed/anden) / Nej / Ved ikke
- hook: claimed\_extraordinary\_cause*

### 12. Var der nævnt strejke hos operatørens personale?

- Ja / Nej / Ved ikke
- hook: operator\_strike\_flag (NB: ikke tredjepart → komp beholdes)*

### 13. Upload evt. meddelelser fra operatøren (push/mail/skærmfoto).

- Upload
- hook: operator\_notice\_upload*

## E) Abonnement/periodekort (Art. 19(2))

### 14. Rejste du på abonnements-/periodekort?

- Ja / Nej
- hook: season\_pass\_used*

### 15. Oplevede du gentagne forsinkelser/aflysninger i kortets periode?

- Ja (angiv antal og datoer) / Nej
- hook: recurrent\_delays\_count*

### 16. Har operatøren et publiceret skema for kompensation til abonnementer? (AUTO)

- Ja (link) / Nej / Ved ikke
- hook: season\_comp\_scheme\_link*

---

## 4) Beregningslogik (pseudokode)

```
if exemptions_matrix.disables_art19(segment_or_contract):
    return {eligible: false, reason: "Art. 19 undtaget for denne tjeneste/stat"}

if delay_preinformed_before_purchase == true:
    return {eligible: false, reason: "Forsinkelser oplyst før køb, Art. 19(9)"}

final_delay = compute_final_arrival_delay_EU_only(rne_data, user_inputs,
geo_filter=EU)

if reroute_under_60_confirmed == true OR final_delay < 60min:
    return {eligible: false, reason: "Endelig forsinkelse < 60 min, Art. 19(9)"}

price_basis = determine_price_basis(
    contract_scope=through_ticket_scope,
```

```

    segment_pricing=segment_prices_shown,
    return_split=return_leg_priced_separately,
    fare_amount=fare_amount_basis
)

if extraordinary_cause_is_valid(operator_claims) == true:
    return {eligible: false, reason: "Art. 19(10) extraordinary circumstances"}
# NB: operator_strike_flag => NOT extraordinary -> ignore this branch

comp_rate = 0.25 if 60<=final_delay<120 else 0.50
comp_amount = round(price_basis * comp_rate, 2)

# Min threshold check
if comp_amount < 4.00 (EUR-equivalent):
    return {eligible: false, reason: "Kompensation under min. udbetalingsgrænse
≤ 4 EUR (Art. 19(8))"}

return {
    eligible: true,
    final_delay_minutes: final_delay,
    rate: comp_rate,
    basis: price_basis,
    amount: comp_amount,
    payout_modes: ["kontant", "voucher (kun hvis fleksibel)"],
    deadline: "betaling senest 1 måned (Art. 19(7))"
}

```

---

## 5) Hooks (feltnavne til din backend)

```

actual_arrival_time_final
missed_connection_station_time
reroute_under_60_confirmed
delay_preinformed_before_purchase
through_ticket_scope
return_leg_priced_separately
segment_prices_shown
fare_amount_basis

```

```
non_eu_segment_present
non_eu_segment_detail
claimed_extraordinary_cause
operator_strike_flag
operator_notice_upload
season_pass_used
recurrent_delays_count
season_comp_scheme_link
```

---

## 6) Mini-JSON-skelet (til form-renderer)

```
{
    "art19_compensation": {
        "A_delay": [
            {"id": "actual_arrival_time_final", "ask_if": "missing(final_arrival) || mismatch(final_arrival)", "q": "Faktisk ankomsttid ved endeligt bestemmessted?", "type": "time"},

            {"id": "missed_connection_station_time", "ask_if": "hint(missed_conn) || mismatch(connection)", "q": "Missede du en forbindelse (station + tid)?", "type": "text"},

            {"id": "reroute_under_60_confirmed", "ask_if": "missing(reroute) || mismatch(reroute)", "q": "Blev du omlagt til <60 min endelig forsinkelse?", "type": "single", "opts": ["Ja", "Nej", "Ved ikke"]},

            {"id": "delay_preinformed_before_purchase", "ask_if": "missing(preinformed) || mismatch(preinformed)", "q": "Var forsinkelsen oplyst før køb?", "type": "single", "opts": ["Ja", "Nej", "Ved ikke"]}
        ],
        "B_price_contract": [
            {"id": "through_ticket_scope", "ask_if": "missing(through) || mismatch(through)", "q": "Var det en gennemgående billet (én kontrakt)?", "type": "single", "opts": ["Ja", "Nej", "Ved ikke"]},

            {"id": "return_leg_priced_separately", "ask_if": "hint(return) || missing(ret_split)", "q": "Returrejse prisopdelt på billetten?", "type": "single", "opts": ["Ja", "Nej", "Ved ikke"]},

            {"id": "segment_prices_shown", "ask_if": "through_ticket_scope != 'Ja' && missing(seg_price)", "q": "Pris pr. segment angivet?", "type": "single", "opts": ["Ja", "Nej"]},

            {"id": "fare_amount_basis", "ask_if": "missing(price_basis)", "q": "Samlet pris for den forsinkede tjeneste/kontrakt", "type": "money"}
        ],
        "C_eu_filter": [
    }
```

```

        {"id":"non_eu_segment_present","ask_if":"hint(nonEU)||  
missing(nonEU)","q":"Indgik der et ikke-EU segment?","type":"single","opts":  
["Ja","Nej","Ved ikke"]},  
  

        {"id":"non_eu_segment_detail","ask_if":"non_eu_segment_present=='Ja'","q":"Angiv  
segment uden for EU (tognr./station-station)","type":"text"}  
  

    ],  
  

    "D_causes": [  
  

        {"id":"claimed_extraordinary_cause","ask_if":"missing(cause)||  
mismatch(cause)","q":"Angav operatøren 'ekstraordinære  
forhold?'","type":"single","opts":  
["Vejr","Naturkatastrofe","Folkesundhed","Andet","Nej","Ved ikke"]},  
  

        {"id":"operator_strike_flag","ask_if":"missing(strike)||  
mismatch(strike)","q":"Var der strejke hos operatørens  
personale?","type":"single","opts":["Ja","Nej","Ved ikke"]},  
  

        {"id":"operator_notice_upload","ask_if":"claimed_extraordinary_cause!=  
'Nej'","q":"Upload operatørmeddelelser (screenshot/mail)","type":"file"}  
  

    ],  
  

    "E_season": [  
  

        {"id":"season_pass_used","ask_if":"missing(season)||  
mismatch(season)","q":"Rejste du på  
abonnement-/periodekort?","type":"single","opts":["Ja","Nej"]},  
  

        {"id":"recurrent_delays_count","ask_if":"season_pass_used=='Ja'","q":"Gentagne  
forsinkelser/aflysninger (antal/datoer)","type":"text"},  
  

        {"id":"season_comp_scheme_link","ask_if":"season_pass_used=='Ja' &&  
missing(season_scheme)","q":"Kender du operatørens kompensationsordning  
(link)?","type":"url"}  
  

    ]  
}
}

```

## 7) UI-tekstblokke (kan indsættes kontekstuel)

- **Tærskel:**  
“EU-kompensation starter ved **60 minutter** på ankomst. 60–119 min: **25 %**;  $\geq$ 120 min: **50 %**.”
- **Pre-info låser kompensation (Art. 19(9))**  
“Hvis forsinkelsen var oplyst **før** du købte, er der **ingen kompensation**.”
- **Omlægning <60 min (Art. 19(9))**  
“Hvis du nåede frem med **<60 min** forsinkelse pga. omlægning, bortfalder kompensationen.”

- **Extraordinary vs. strejke (Art. 19(10))**

“Ekstraordinære forhold kan afskære kompensation. **Strejker hos operatøren** er ikke en sådan undtagelse.”

- **Udbetaling (Art. 19(7),(8))**

“Udbetales **senest 1 måned** efter ansøgning. Du kan vælge **kontant**. Vouchers kun hvis **fleksible**.”

---

## 8) Minimal “beregnings-step” til din sidste kasse

- Henter: `final_delay_minutes_EU, price_basis, preinformed, reroute_under_60, extraordinary_valid, exemption_status`.
  - Returnerer: `{eligible, rate, amount, reason?, payout_modes, deadline}`.
  - Logger alle hooks + en “explain” streng (så Copilot kan autoskrive afgørelsestekst til PDF).
- 

Når du er klar, laver jeg tilsvarende **Art. 18 (refusion/omlægning)** og **Art. 20 (assistance)** i samme stil, så hele kæden kører ensartet.

## Step Refusion

Perfekt — du er nu på det **svære, men afgørende stykke af flowet**: *Step Refusion* (Art. 18 + Art. 4 CIV + Art. 10 + dele af 20).

Jeg har omskrevet og samlet det til en **struktureret, automatisérbar model**, der matcher formatet for *Art. 19 kompensation* – dvs.:

- klar juridisk struktur
- AUTO → spørg → logik
- hooks og JSON-skelet
- tekstblokke til UI/diagram
- dependencies forklaret

---

## STEP REFUSION — samlet struktur

### 1 Formål

At afgøre **om og hvordan passageren har ret til refusion eller omlægning** i henhold til:

- **Art. 18 (1–3)** i forordning 2021/782,
- **CIV art. 6–25** (bilag I → selvforskyldt/ikke-selvforskyldt),
- **Art. 4 + Art. 10 (b)** (bortfald af kompensation ved egen skyld),
- samt **Art. 20 (2–4)** (omkostninger og beviser).

---

### ◆ Del A – CIV-check / selvforskyldt situation

**AUTO:** Billet-OCR + RNE-API + operatørdata for rejseafvisninger.

Vis kun spørgsmål hvis ukendt, mismatch, eller “self-inflicted cause” markeret.

#### Spørgsmål:

1. **Havde du en gyldig billet under hele rejsen?**
  - Ja / Nej / Ved ikke

*hook: valid\_ticket\_present*
2. **Blev du bortvist eller nægtet ombordstigning?**
  - Ja (årsag?) / Nej

*hook: boarding\_refused\_reason*
3. **Udviste du adfærd, der udgjorde en sikkerhedsrisiko eller krænkede andre?**
  - Ja / Nej / Ved ikke

*hook: misconduct\_flag*
4. **Medbragte du dyr/genstande, som ikke var tilladt?**
  - Ja / Nej / Ved ikke

*hook: prohibited\_items\_flag*

**5. Overholdt du told- og myndighedskrav under rejsen?**

– Ja / Nej / Ved ikke

*hook: customs\_compliance\_flag*

**6. Har du dokumentation (påtegning fra operatør) for aflysning/forsinkelse?**

– Upload / Nej

*hook: operator\_endorsement\_upload*

**Automatisk logik:**

Hvis nogen “egen skyld” = Ja → `self_inflicted=true`

→ afslut refusions-/kompensationsflow med **afvisning (Art. 10 b)**.

---

◆ Del B – Årsag til anmodningen (PDF punkt 1)

**MCQ-flueben:**

- Forsinkelse
- Aflysning
- Mistet forbindelse pga. forsinkelse/aflysning  
*hook: reason\_delay, reason\_cancellation, reason\_missed\_conn*

**AUTO:** Forsinkelsesdata fra RNE → forudafkrydsning.

Bruger kan rette manuelt.

---

◆ Del C – Hvilken type krav? (PDF punkt 4)

**MCQ-flueben med multiple valg:**

- **Refusion af billetten** pga. aflysning/forsinkelse (Art. 18 (1)(a))
- **Omlægning/videreførelse** af rejsen (Art. 18 (1)(b) eller (c))
- **Kompensation** (Art. 19 – via Step Kompensation)
- **Refusion af ekstraudgifter** (bus, taxi, hotel osv.) (Art. 20 (2))  
*hook: claim\_refund\_ticket, claim\_rerouting, claim\_compensation, claim\_other\_costs*

◆ Del D – Refusion (Art. 18 (1)(a))

**1. Blev hele rejsen aflyst, og vendte du tilbage til udgangspunktet?**

– Ja / Nej

*hook: trip\_cancelled\_return\_to\_origin*

**2. Anmodede du om refusion fra operatøren? (AUTO)**

– Ja / Nej / Ved ikke

*hook: refund\_requested*

**3. Hvis ja, hvilken form for refusion?**

– Kontant / Voucher / Andet

*hook: refund\_form\_selected*

---

◆ ~~Del E Omlægning (Art. 18 (1)(b) & (c))~~

**1. Fik du tilbuddt omlægning på tilsvarende vilkår ved først give lejlighed?**

– Ja / Nej / Ved ikke

*hook: reroute\_same\_conditions\_soonest*

**2. Eller ønskede du omlægning på et senere tidspunkt efter eget valg?**

– Ja / Nej

*hook: reroute\_later\_at\_choice*

**3. Fik du besked om mulighederne inden for 100 minutter? (Art. 18 (3))**

– Ja / Nej / Ved ikke

*hook: reroute\_info\_within\_100min*

**4. Medførte omlægningen ekstra udgifter for dig (højere klasse/andet transportmiddel)?**

– Ja / Nej / Ved ikke

*hook: reroute\_extra\_costs*

**5. Var du nedklassificeret pga. omlægning (lavere kategori)?**

– Ja / Nej

*hook: downgrade\_occurred*

**Hvis downgrade\_occurred=true → link til Bilag II I.4 – I.6 – I.8**

(for at beregne tilbagebetaling for lavere standard).

---

◆ ~~Del F Udgifter (Art. 20 (2)(a) – (e))~~

**1. Fik du tilbuddt måltider/førfriskninger under ventetiden?**

– Ja / Nej

*hook: meal\_offered*

**2. Fik du hotel/indkvartering (inkl. transport dertil)?**

– Ja / Nej

*hook: hotel\_offered*

**3. Hvis toget var blokeret, fik du alternativ transport?**

– Ja / Nej

*hook: blocked\_train\_alt\_transport*

**4. Fik du alternativ transport, hvis forbindelsen blev helt afbrudt? (Art. 20 (3))**

– Ja / Nej

*hook: alt\_transport\_provided*

5. **Har du haft udgifter til taxi, bus, hotel, mad osv.?**

– Ja (upload bilag) / Nej

*hook: extra\_expense\_upload*

6. **Fik du skriftlig bekræftelse på forsinkelsen/aflysningen?**

– Ja / Nej

*hook: delay\_confirmation\_received*

---

◆ Del G – Force Majeure / Egen skyld boks

Kombinerer CIV art. 6–25 + Art. 19 (10):

- **Ekstraordinære forhold** (natur, folkesundhed, sabotage)  
*hook: extraordinary\_flag*
- **Selvforskyldt forhold** (CIV)  
*hook: self\_inflicted*
- **Tredjepartsfejl** (anden operatør/infrastrukturforvalter)  
*hook: third\_party\_fault*

Hvis én af ovenstående markeres → automatisk “ingen kompensation/refusion”.

---

◆ Del H – Oplysninger under rejsen (Art. 10 + 30(2))

AUTO fra Bilag II del II (livedata).

Vis kun spørgsmål hvis mismatch.

1. **Viste toget/skærmen næste station i realtid?**

– Ja / Nej

*hook: next\_station\_displayed*

2. **Fik du realtidsopdatering om forsinkelser/afbrydelser?**

– Ja / Nej

*hook: realtime\_info\_available*

3. **Blev der oplyst forbindelser til andre transportmidler (bus, metro, tog)?**

– Ja / Nej

*hook: connections\_info\_displayed*

4. **Fik du sikkerhedsinformation ombord?**

– Ja / Nej

*hook: safety\_info\_provided*

---

◆ Del I – Output/afgørelse

Automatisk vurdering:

`if self_inflicted or extraordinary_flag:`

```

        outcome = "Afslag (egen skyld / force majeure)"

elif claim_refund_ticket:
    outcome = "Refusion jf. Art.18(1)(a)"

elif claim_rerouting and reroute_info_within_100min:
    outcome = "Omlægning jf. Art.18(1)(b/c)"

elif claim_other_costs and meal_offered==false:
    outcome = "Manglende assistance, Art.20(2) -> udgiftsrefusion"

else:
    outcome = "Krav sendes til kompensationsflow (Art.19)"

```

---

## ◆ JSON-skelet

```
{
  "art18_refund": {
    "A_civ_self": [
      {"id": "valid_ticket_present", "q": "Gyldig billet under hele rejsen?", "type": "single", "opts": ["Ja", "Nej", "Ved ikke"]},
      {"id": "boarding_refused_reason", "q": "Blev du nægtet ombordstigning?", "type": "text"},
      {"id": "misconduct_flag", "q": "Udviste du adfærd der udgjorde sikkerhedsrisiko?", "type": "single", "opts": ["Ja", "Nej", "Ved ikke"]},
      {"id": "prohibited_items_flag", "q": "Medbragte du dyr/genstande der ikke var tilladt?", "type": "single", "opts": ["Ja", "Nej", "Ved ikke"]},
      {"id": "customs_compliance_flag", "q": "Overholdt du told/myndighedskrav?", "type": "single", "opts": ["Ja", "Nej", "Ved ikke"]},
      {"id": "operator_endorsement_upload", "q": "Upload påtegning fra operatør (aflysning/forsinkelse)", "type": "file"}
    ],
    "B_reason_flags": [
      {"id": "reason_delay", "q": "Forsinkelse", "type": "checkbox"},
      {"id": "reason_cancellation", "q": "Aflysning", "type": "checkbox"},
      {"id": "reason_missed_conn", "q": "Mistet forbindelse pga. forsinkelse/aflysning", "type": "checkbox"}
    ],
    "C_claim_type": [
      {"id": "claim_refund_ticket", "q": "Refusion af billetten (Art.18(1)(a))", "type": "checkbox"},
      {"id": "claim_rerouting", "q": "Omlægning af rejsen (Art.18(1)(b/c))", "type": "checkbox"},
      {"id": "claim_compensation", "q": "Kompensation (Art.19)", "type": "checkbox"}
    ]
  }
}
```

```

        {"id":"claim_other_costs","q":"Refusion af andre udgifter  

(Art.20(2))","type":"checkbox"}  

],  

"D_refund_specific": [  

    {"id":"trip_cancelled_return_to_origin","q":"Aflyst rejse, retur til  

start?","type":"single","opts":["Ja","Nej"]},  

    {"id":"refund_requested","q":"Anmodet om refusion hos  

operator?","type":"single","opts":["Ja","Nej","Ved ikke"]},  

    {"id":"refund_form_selected","q":"Hvilken form for  

refusion?","type":"single","opts":["Kontant","Voucher","Andet"]}  

],  

"E_rerouting": [  

    {"id":"reroute_same_conditions_soonest","q":"Omlægning på tilsvarende  

vilkår hurtigst muligt?","type":"single","opts":["Ja","Nej","Ved ikke"]},  

    {"id":"reroute_later_at_choice","q":"Omlægning på senere tidspunkt efter  

eget valg?","type":"single","opts":["Ja","Nej"]},  

    {"id":"reroute_info_within_100min","q":"Fik du besked om mulighederne  

inden for 100 min?","type":"single","opts":["Ja","Nej","Ved ikke"]},  

    {"id":"reroute_extra_costs","q":"Medførte omlægningen ekstra  

udgifter?","type":"single","opts":["Ja","Nej","Ved ikke"]},  

    {"id":"downgrade_occurred","q":"Blev du  

nedklassificeret?","type":"single","opts":["Ja","Nej"]}  

],  

"F_expenses": [  

    {"id":"meal_offered","q":"Fik du måltider/forfriskninger under  

ventetid?","type":"single","opts":["Ja","Nej"]},  

    {"id":"hotel_offered","q":"Fik du hotel/indkvartering + transport  

dertil?","type":"single","opts":["Ja","Nej"]},  

    {"id":"blocked_train_alt_transport","q":"Var toget blokeret - fik du  

alternativ transport?","type":"single","opts":["Ja","Nej"]},  

    {"id":"alt_transport_provided","q":"Fik du alternativ transport ved  

afbrudt forbindelse?","type":"single","opts":["Ja","Nej"]},  

    {"id":"extra_expense_upload","q":"Upload kvitteringer (taxi, hotel, mad,  

mv.)","type":"file"},  

    {"id":"delay_confirmation_received","q":"Fik du bekræftelse på  

forsinkelsen/aflysningen?","type":"single","opts":["Ja","Nej"]}  

],  

"G_force_self": [  

    {"id":"extraordinary_flag","q":"Ekstraordinære forhold (natur,  

folkesundhed mv.)","type":"checkbox"},  

    {"id":"self_inflicted","q":"Selvforskyldt forhold (CIV art.6-  

25)","type":"checkbox"},  

]

```

```

        {"id":"third_party_fault","q":"Tredjepartsfejl (anden
operator/infrastruktur)","type":"checkbox"}
    ],
    "H_info_onboard": [
        {"id":"next_station_displayed","q":"Viste toget næste station i
realtid?","type":"single","opts":["Ja","Nej"]},
        {"id":"realtime_info_available","q":"Fik du realtime-info om
forsinkelser?","type":"single","opts":["Ja","Nej"]},
        {"id":"connections_info_displayed","q":"Oplysning om forbindelser til
bus/metro/tog?","type":"single","opts":["Ja","Nej"]},
        {"id":"safety_info_provided","q":"Fik du sikkerhedsinformation
ombord?","type":"single","opts":["Ja","Nej"]}
    ]
}
}

```

---

## ◆ Tekstblokke til UI/diagram

- **Egen skyld / force majeure:**

“Hvis rejsen blev afvist eller afbrudt pga. din egen adfærd eller ekstraordinære forhold (Art. 10 b + 19 (10)), bortfalder retten til refusion og kompensation.”

- **Refusion:**

“Ved aflysnings kan du vælge at få **billetten refunderet** eller **rejse videre** (Art. 18 (1)(a–c)).”

- **Omlægning:**

“Operatøren skal inden 100 minutter tilbyde omlægning på **tilsvarende vilkår** eller efter eget valg (Art. 18 (3)).”

- **Manglende assistance:**

“Manglende tilbud om måltider, hotel eller alternativ transport registreres som **brud på Art. 20 (2)** og kan give ret til refusion af udgifter.”

- **Downgrade:**

“Hvis du blev placeret i lavere kategori end købt, udløser det **delvis refusion** (Art. 18 + Bilag II I.4–6–8).”

## ◆ Dependencies (til dine moduler)

Reference	Afhængighed	Effekt
Art. 4 + CIV 6–25	valid_ticket_present ...	Afvisning ved selvforskyldt hændelse
Art. 18 (1–3)	claim_refund_ticket, reroute_same_conditions_soonest ...	Beregner ret til refusion/omlægning
Art. 19 (10)	extraordinary_flag	Stopper både refusion & kompensation

Reference	Afhængighed	Effekt
<b>Art. 20 (2–4)</b>	meal_offered, hotel_offered, ...	
Tilføjer refusion for udgifter		
<b>Art. 12</b>   through_ticket_scope	Sikrer korrekt beregning ved gennemgående billet	
<b>Bilag II I.4,6,8</b>   downgrade_occurred	Delvis refusion (lavere kategori)	
<b>Bilag II II.1–4</b>   realtime_info_available	Bevis for manglende information i realtid	

---

Vil du have, at jeg nu laver **Step Assistance (Art. 20)** i præcis samme format (spørgsmål + JSON + logik + dependencies)?

Så bliver det sidste led i “core chain”: billet → refusion → kompensation → assistance.

## Replace punkt E og F ovenfor :

Yes — replace your current blocks with these tightened versions (klar til copy-paste).

Jeg har lagt “AUTO/mismatch”-noter, exemptions-hook og 3-nætters-loftet ind, samt små UI-prompter.

---

## Del E – Omlægning (Art. 18(1)(b)-(c))

Vis kun, hvis forsinkelse  $\geq 60$  min, aflysning eller mistet forbindelse (AUTO fra RT/RNE).

### 1. Fik du tilbuddt omlægning på tilsvarende vilkår ved først givne lejlighed?

– Ja / Nej / Ved ikke

hook: reroute\_same\_conditions\_soonest

### 2. Ønskede du i stedet omlægning på et senere tidspunkt efter eget valg?

– Ja / Nej

hook: reroute\_later\_at\_choice

### 3. Fik du besked om mulighederne for omlægning inden for 100 minutter? (Art. 18(3))

– Ja / Nej / Ved ikke

hook: reroute\_info\_within\_100min

UI hint (auto): “Vi bruger planlagt afgang + første omlægnings-besked til at vurdere 100-min-reglen.”

### 4. Medførte omlægningen ekstra udgifter for dig? (højere klasse/andet transportmiddel)

– Ja / Nej / Ved ikke

hook: reroute\_extra\_costs

If Ja → beløb + valuta (upload kvitteringer)

hook: reroute\_extra\_costs\_amount, currency, extra\_expense\_upload

### 5. Blev du nedklassificeret pga. omlægning (lavere kategori end købt)?

– Ja / Nej

hook: downgrade\_occurred

Hvis downgrade\_occurred=true → kør delvis tilbagebetaling efter **Bilag II del I, pkt. 4/6/8** (sæder/ligge/sove).

hook: downgrade\_comp\_basis

*Exemptions-note (AUTO): Hvis land/tjeneste er fritaget for Art. 18(3) (100-min) i din matrix, vis banner:*

“ $\Delta$  100-min-reglen kan være undtaget her. Vi logger stadig dine udgifter og afprøver krav efter lokal praksis.”

---

# Del F – Udgifter / Assistance (Art. 20(2)(a)-(c), 20(3), 20(4))

**Tærskel (AUTO):** Aktiveres ved forsinkelse  $\geq 60$  min, aflysning eller afbrudt forbindelse.

**Ekstraordinære forhold (Art. 19(10)) påvirker kun hotel-loft → max 3 nætter.**

## A) Tilbuds assistance

### 1. Fik du måltider/forfriskninger under ventetiden? (20(2)(a))

– Ja / Nej

*hook:* meal\_offered

### 2. Fik du hotel/indkvartering + transport dertil? (20(2)(b))

– Ja / Nej

*hook:* hotel\_offered

*If Nej: Blev overnatning nødvendig? – Ja / Nej*

*hook:* overnight\_needed

*UI hint (extraordinary AUTO): “Ved ekstraordinære forhold kan hotel begrænses til 3 nætter.”*

### 3. Var toget blokeret på sporet — fik du transport væk? (20(2)(c))

– Ja / Nej

*hook:* blocked\_train\_alt\_transport

## B) Alternative transporttjenester

### 4) Fik du alternative transporttjenester, hvis forbindelsen blev afbrudt? (20(3))

– Ja / Nej

*hook:* alt\_transport\_provided

## C) Dokumentation & udgifter

### 5) Har du haft udgifter (taxi, bus, hotel, mad osv.)?

– Ja (upload) / Nej

*hook:* extra\_expense\_upload + *breakdown:* meals, hotel\_nights,

local\_transport, other\_amounts, currency

### 6. Fik du skriftlig bekræftelse på forsinkelse/aflysning/mistet forbindelse? (20(4))

– Ja (upload) / Nej

*hook:* delay\_confirmation\_received, delay\_confirmation\_upload

## D) Ekstraordinære forhold (kobling til Art. 19(10))

### 7) Henviste operatøren til ekstraordinære forhold?

– Ja / Nej / Ved ikke

*hook:* extraordinary\_claimed

*If Ja: Type – Vejr / Naturkatastrofe / Folkesundhed / Andet*

*hook:* extraordinary\_type

*UI hint:* “Dette påvirker kun loftet for hotel (op til 3 nætter), ikke dine øvrige assistance-krav.”

*Exemptions-note (AUTO):* Hvis Art. 20(2) er fritaget på strækningen → vis banner:  
“⚠ Assistance (måltider/hotel/transport) kan være undtaget her. Vi logger dine udgifter og rejser  
krav efter lokale regler/kontraktvilkår.”

---

## Hooks (samlet)

```
reroute_same_conditions_soonest
reroute_later_at_choice
reroute_info_within_100min
reroute_extra_costs
reroute_extra_costs_amount
currency
downgrade_occurred
downgrade_comp_basis

meal_offered
hotel_offered
overnight_needed
blocked_train_alt_transport
alt_transport_provided
extra_expense_upload
delay_confirmation_received
delay_confirmation_upload
extraordinary_claimed
extraordinary_type
```

## Mini-UI-tekster (kan indsættes som tooltips/bannere)

- “Ved **≥60 min** forsinkelse/aflysnings har du krav på assistance (Art. 20).”
- “**100-min-regel:** operatøren skal informere om omlægning inden 100 min (Art. 18(3)).”
- “**Ekstraordinære forhold** påvirker kun hotel-loftet (**max 3 nætter**).”
- “**Exemptions aktiv:** vi håndterer dine udgifter efter lokal praksis.”

Sig til, hvis du vil have det som et lille JSON-schema eller TypeScript-interface til din form-renderer — så får Copilot noget helt konkret at bygge UI’et ud fra.

## Step udregning af kompensationen

Perfekt — du er nu dér, hvor vi kan samle hele den endelige “kerne-flowkæde”:

→ **Billetinfo** → **Art. 12 (gennemgående billet)** → **Refusion (Art. 18 + CIV)** → **Kompensation (Art. 19)** → **Assistance (Art. 20)** → **Bevisførelse**.

Her får du **den færdige, integrerede version** af  
“**Step Udregning af kompensation + Bevisførelse**”,  
i samme format som de foregående moduler — klar til diagram / app-flow / automatisering.  
Alt er sat op med dependencies, AUTO-kilder, hooks og beviskæder.

---

## ⌚ Step Udregning af kompensation (Art. 19 + relaterede artikler)

### 🎯 Formål

Automatisk at fastlægge **kompensationsberettigelse og beløb**,  
ved at sammenholde **billet-, RNE- og brugerdata** med:

- Art. 12 (1–7) → gennemgående billet / kontraktansvar
- Bilag II del 1 → forudoplysninger (hvad passageren fik lovet)
- Art. 18 → om refusion eller omlægning allerede skete
- Art. 19 → selve kompensationsberegningen
- Art. 10 + CIV 6–25 → egen skyld / force majeure
- Art. 20 → assistance (som bevis og alternativ kravtype)

---

### ◆ 1. Dependencies (input-kæde)

Input	Kilde	Bruges til
Billetdata	OCR / API	kontrakt, pris, klasse, segmenter
RT data (RNE)	RT-feed	faktisk forsinkelse, aflysninger
Art. 12-output	Step Billetinformation pkt. C	afgør om gennemgående billet
Bilag II del 1 pkt. 7	Step Billetinfo pkt. D	info om planlagte/aktuelle forsinkelser
Refusion-output	Step Refusion	allerede refundert?
Assistance-output	Step Assistance	supplerende udgifter/bevis
CIV doktrin	Bilag I (art. 6–25)	selforskyldt/ikke-selforskyldt
Art. 10 / force majeure	Step Refusion E10	vurdering og flagning
National tabel	Bilag X	nationale takst-modifikationer

---

### ◆ 2. Grundlogik (Auto)

```
if self_inflicted or extraordinary_flag:  
    outcome = "No compensation (Art.10(b)/19(10))"  
elif refund_already_paid:  
    outcome = "Already reimbursed under Art.18"
```

```

else:
    # Beregn forsinkelse (EU-segmenter)
    delay = calc_delay_EU_only(rne_segments)

    if delay < 60min:
        outcome = "No compensation (<60min)"

    elif 60 <= delay < 120:
        compensation_pct = 25

    else:
        compensation_pct = 50

amount = ticket_price * (compensation_pct/100)

```

**NB:** Nationale billetter → check bilag X for lavere minimumssatser.

---

#### ◆ 3. Force Majeure / Doktrin-check (E10)

**AUTO:** Sammenlign operatørens årsagskode (Art. 10 data) med brugerens beskrivelse.  
Hvis mismatch eller ukendt → flag til manuel vurdering.

Type	Handling
a) Natur, folkesundhed, sabotage	markér <code>extraordinary_flag=true</code> → udeluk kompensation
b) Egen adfærd (CIV 6–25)	<code>self_inflicted=true</code> → afvis
c) Tredjepartsfejl (infrastruktur, anden operatør)	stadic kompensation (Art. 19 (10) d))
d) Strejke hos jernbaneoperatør	<b>ikke undtagelse</b> → kompensation

Flag alle (a–d) sager → `manual_review=true`.

---

#### ◆ 4. Utdeling af kompensationsbeløb

##### **[1] Fastlæg prisgrundlag (Art. 19(3)):**

- Hvis returrejse → halv billetpris.
- Hvis flere segmenter uden prisangivelse → halv pris.
- Hvis gennemgående billet (Art. 12) → andel efter forsinket del.

##### **[2] Tæl kun EU-segmenter (Art. 19(4)):**

- `segment.country in EU` → medtag
- udenfor EU → ignorer i beregning.

##### **[3] Beregn procent**

- 25 % ved 60–119 min
- 50 % ved ≥120 min

#### 4 Fratræk evt. tidlige refusion (Art. 18)

→ ingen dobbelt dækning.

---

#### ◆ 5. Særlige regler / nationalt tillæg

- Nationale ruter kan have lavere satser (bilag X).
  - Appen henter country\_code fra billet og vælger sats.
- 

#### ◆ 6. Output-felter (hooks)

compensation\_eligible

delay\_minutes

delay\_category

ticket\_price

compensation\_pct

compensation\_amount

extraordinary\_flag

self\_inflicted

manual\_review

refund\_already\_paid

eu\_segments\_only

national\_adjustment

---

## Step Bevisførelse — samling og krydstjek

### Formål

At dokumentere **hele sagskæden** fra billet → RNE → brugerdata, så operatøren ikke kan afvise pga. manglende dokumentation.

---

#### ◆ 1. Beviskilder

Kategori	Kilde	Brug i sagsvurdering
Billet-/PNR-OCR	upload	pris, klasse, kontrakt, rute
RNE API	realtidsdata	bekræfter forsinkelse/aflysning
Operatørpåtegning	upload	Art. 20(4) bevis for disruption
Refusion/komp. kvittering	upload	undgå dobbeltkrav
Udgiftsbilag	upload	hotel, mad, transport
Kommunikation fra operatør	upload	årsagsangivelse (force majeure)
Bilag II del 1	præ-kontrakt-info	sammenligning ved downgrade/refusion
Bilag II del II	livedata	verifikation af faktisk drift
Art. 12-scope	interne data	ansvarsplacering

Kategori	Kilde	Brug i sagsvurdering
Exemption-matrix	database	styrer hvilke artikler der gælder
<hr/>		
◆ 2. Krydkontrol (automatisk)		
Sammenligning	Hvad kontrolleres	Resultat
Billet ↔ RNE	Rejsetid, tog-ID, klasse	bekræft/afvigelse
RNE ↔ bruger	faktisk vs. oplyst forsinkelse	fastlæg delay
Operatør-årsag ↔ bruger-årsag	force majeure-match	flag mismatch
Art. 12 ↔ pris	del-kontrakt vs. samlet billet	deling af prisgrundlag
Bilag II del 1 ↔ del II	lovet vs. leveret service	downgrade-vurdering
Assistance ↔ bilag	manglende tilbud → refusionskrav	bevis for brud på Art. 20
Exemptions ↔ jurisdiktion	artikler gælder/ikke	justér kravtype

---

◆ 3. Bevis-output (JSON-objekt)

```
{
  "evidence_bundle": {
    "ticket_doc": "file_id_...",
    "rne_snapshot": "json_payload",
    "operator_certificate": "file_id_...",
    "expense_receipts": ["file_id_..."],
    "operator_reason": "text / code",
    "user_reason": "text",
    "delay_minutes": 128,
    "force_majeure_classification": "none | extraordinary | self_inflicted | third_party",
    "compensation_amount": 42.50,
    "currency": "EUR",
    "manual_review": false
  }
}
```

---

◆ 4. Output-logik

```
if manual_review or evidence_incomplete:
    status = "Pending manual verification"
else:
    status = "Ready for operator submission"
```

---

## ◆ 5. UI-tekstblokke

- **Manglende bilag:**  
“Upload billet, forsinkelsesbekræftelse og evt. udgiftsbilag for at fuldføre bevisførelsen.”
  - **Force majeure flag:**  
“Operatøren har angivet årsag som ‘ekstraordinære forhold’. Sagen markeres til manuel gennemgang.”
  - **Afvisning grundlag:**  
“Egen skyld (CIV 6–25) eller force majeure udelukker kompensation, jf. Art. 10 b.”
  - **Kompensationsresultat:**  
“Du er berettiget til **{compensation\_pct} %** af billetprisen, svarende til **{compensation\_amount} {currency}**.”
- 

## ◆ 6. Efterfølgende trin

- [1] Generér PDF-sagsresumé med alle beviser (til operatør).
  - [2] Log audit trail (AUTO-hash af uploads).
  - [3] Tilføj 25 % kommissions-fee ved overdragelse af krav → `claim_transfer_fee=0.25`.
- 

## ◆ 7. Dependencies-kort (for diagram)

Billetinfo → Art.12 (gennemgående billet)

- Bilag II del I-II
  - Step Refusion (Art.18/CIV)
  - Step Kompensation (Art.19)
  - Step Assistance (Art.20)
  - Art.10 / Force majeure check
  - ↳ Step Bevisførelse (samler alt)
- 

### Resultat:

Dette modul afslutter hele kæden: alle artikler 4 → 12 → 18 → 19 → 20 (+ CIV 6–25 + Bilag II) er forbundet med automatisk vurdering, manual-flagging og fuld bevisførelse.

---

Vil du have, at jeg formaterer det som et samlet **flow-diagram-tekstark** (til draw.io / Lucidchart) med boks-labels og pile, så du kan indsætte det direkte i dit ODG-diagram?

## **step Fuldmagt og Identifikation**

Perfekt — du rammer lige præcis det trin, hvor **compliance, legitimation og samtykke** skal hænge sammen med de juridiske krav i **EU-passagerrettigheder, GDPR og anti-fraud-regler**.

Her får du et færdigt “**Blueprint for Step Fuldmagt og Identifikation**”, som du kan indsætte i dit flow (samme struktur som de øvrige steps).

Jeg har opdelt det i:

- [1] Formål og retsgrundlag**
  - [2] Fuldmagt (power of attorney)**
  - [3] Identifikation (eID / ID-upload)**
  - [4] Sammenkørsel af signatur + ID**
  - [5] GDPR og datasikkerhed**
  - [6] Teknisk model (hooks + flow)**
- 

## **■ Step Fuldmagt & Identifikation**

(“*Verification & Power of Attorney Step*”)

### **🎯 Formål**

At sikre, at **kravet indgives på vegne af en identificeret person**,  
at **du som tredjepart (claim handler)** har **juridisk mandat** til at handle,  
og at **udbetaling** sker til rette person i overensstemmelse med:

- Forordning (EU) 2021/782 (passagerrettigheder)
  - GDPR (EU) 2016/679
  - Anti-fraud / AML-principper
  - Nationale eID-løsninger
- 

### **[1] Retsgrundlag**

<b>Grundlag</b>	<b>Anvendelse</b>
Art. 19–20 i forordning 2021/782	Udbetaling af kompensation og refusion
Art. 5(1)(a) og (b) GDPR	Samtykke + kontraktmæssig behandling
eIDAS-forordningen (EU) 910/2014	Anerkendelse af elektronisk identitet og signatur
National civilret	Fuldmagtens gyldighed og formkrav

---

### **[2] Fuldmagt (Power of Attorney)**

**Formål:** Dokumentere, at brugeren (passageren) overdrager retten til at håndtere og modtage kompensation/refusion.

**Minimumsindhold (gælder i hele EU):**

- Fulde navn, adresse, fødselsdato, nationalitet
- Identifikation (ID-type, nummer, udsteder)
- Henvisning til rejsen (PNR/billetnr., dato, operatør)
- Formål: *“Jeg bemyndiger [din virksomhed] til at fremsætte, behandle og modtage betaling i forbindelse med krav om kompensation/refusion efter EU-forordning 2021/782.”*
- Angivelse af, at fuldmagten kan tilbagekaldes skriftligt
- Dato og underskrift (fysisk eller eID)

#### **Oversættelse / lokal version:**

- Fuldmagtsteksten vises automatisk på relevant sprog (baseret på landekode).
  - EU's sprogregel: alle medlemsstater skal acceptere originalsprog + engelsk version ved elektronisk signatur (eIDAS art. 25).
- 

### **[3] Identifikation (ID / eID)**

#### **Primær metode (AUTO):**

- Brug nationalt eID (f.eks. MitID, BankID, NemID, eIDAS-notified systemer).  
→ verificer identitet og bind signaturen direkte til eID-assertion.

#### **Fallback-metode:**

- Upload af gyldigt ID-dokument (pas, ID-kort, kørekort)
- Selfie med ID (anti-fraud-match)
- Automatisk OCR-validering og “liveness check”

#### **Gemte felter (hooks):**

```
user_fullname  
user_birthdate  
user_nationality  
id_type  
id_number  
id_issuer  
id_verified (true/false)  
signature_hash  
signature_timestamp  
poa_signed (true/false)
```

---

### **[4] Sammenkørsel af Signatur + ID**

**Mål:** ID og underskrift knyttes sammen i én transaktion.

#### **Teknisk model (EU-kompatibel):**

- Signering og identifikation sker i ét flow (Single Sign-On eIDAS).
- Hvis brugeren logger ind via MitID/BankID → eID-data anvendes som digital underskrift.  
*(du behøver ingen separat “fysisk” signatur)*

## Alternativ (hvis ikke eID):

- Brugeren underskriver digital fuldmagt (PDF eller HTML-signing).
  - Derefter uploades ID → systemet matcher navn og fødselsdato.
  - Hashes gemmes (SHA256) → danner bevis for uafviselighed (non-repudiation).
- 

## 5 GDPR & datasikkerhed

Behandling	Retsgrundlag	Opbevaringsperiode	Kommentar
ID-validering	Art. 6(1)(b): kontrakttopfyldelse	max 5 år efter kravslut	Bruges kun til legitimation
Fuldmagt (POA)	Art. 6(1)(a): samtykke	indtil krav afsluttet + 3 år	Skal kunne dokumenteres
Transaktionslog	Art. 6(1)(f): legitim interesse	5 år	Anti-fraud, revision
Uploadedede dokumenter	Art. 6(1)(b)/(f)	12 mdr. efter afslutning	Krypteres og slettes automatisk

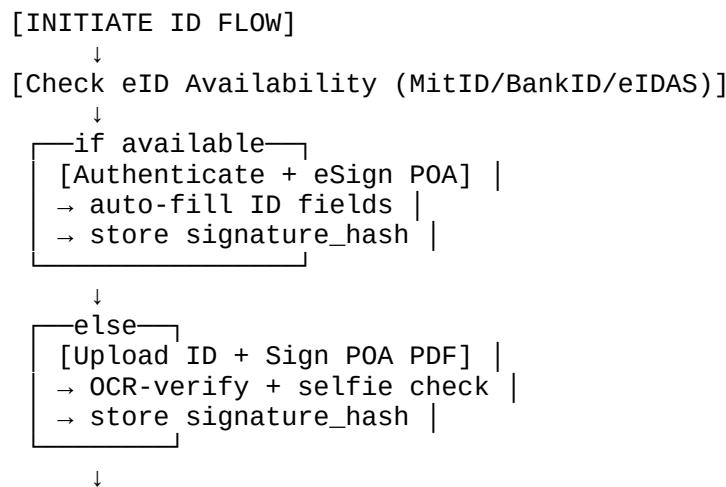
### Brugerrettigheder:

- Ret til indsigt og sletning (GDPR Art. 15, 17)
- Ret til dataportabilitet (Art. 20)
- Samtykke kan trækkes tilbage via “Min sag”

### Datasikkerhed:

- Kryptering (AES-256, HTTPS/TLS 1.3)
  - Hashing af underskrifter
  - Audit trail med tidsstemplere
  - Geo-fencing: Data opbevares i EU/EØS
- 

## 6 Teknisk Blueprint / Flow



```
[Store Consent + Log GDPR Audit]
```

```
↓
```

```
[Return verified_user=true]
```

## Output:

```
{  
  "verified_user": true,  
  "poa_signed": true,  
  "id_verified": true,  
  "signature_hash": "sha256:abcd1234...",  
  "signature_timestamp": "2025-10-09T13:45:00Z",  
  "consent_gdpr": true  
}
```

---



### UI-tekstblokke (til diagram)

- “Bekræft din identitet med MitID eller anden EU-godkendt eID.”
  - “Ved at underskrive giver du fuldmagt til, at vi håndterer dit krav og modtager eventuel kompensation på dine vegne.”
  - “Vi gemmer kun dine ID-oplysninger sikkert i overensstemmelse med GDPR.”
  - “Du kan til enhver tid trække din fuldmagt tilbage via ‘Min sag’.”
- 

### 🧠 BONUS: Mulige eID-løsninger pr. land (eIDAS-notified)

#### Land      Løsning

DK	MitID
SE	BankID
NO	BankID
FI	Suomi.fi eID
DE	eID Karte / Verimi
FR	FranceConnect+
NL	DigiD
IT	SPID / CIE
ES	Cl@ve
BE	Itsme
AT	ID Austria

---

Vil du have, at jeg udarbejder **en standard POA-tekstskabelon (på dansk og engelsk)** med de korrekte juridiske formuleringer til signering via MitID / eIDAS, så du kan indlejre den i appen som dokumenttekst?

Yes—lad os samle **refusion + kompensation + udgifter** i én samlet udregning, så du kan trække dit **25 % servicefee** af *det samlede krav*. Her er en skarp blueprint, som Copilot kan kode direkte efter (med eksempler på dine tre fixtures).

---

# ClaimCalculator v1 — samlet udregning

## Input (krav til dit service-lag)

```
type ClaimInput = {  
    country_code: string           // til exemptions/nationale satser  
    currency: string               // fx "EUR" / "DKK"  
    ticket_price_total: number     // samlet betalt pris for hele  
    kontrakten/billetten  
  
    trip: {  
        through_ticket: boolean    // Art. 12(1-7)  
        legs: Array<{  
            from:string,to:string, eu:boolean,  
            scheduled_dep:string, scheduled_arr:string,  
            actual_dep?:string, actual_arr?:string,  
            price?:number           // hvis oplyst pr. strækning  
        }>  
    },  
    disruption: {  
        delay_minutes_final?: number // hvis du allerede har det  
        notified_before_purchase?: boolean // Art. 19(9)  
        rerouting_offered_within_100m?: boolean // Art. 18(3)  
        cause_code?: string          // fra Art. 10-data / operatør  
        user_cause_text?: string  
        extraordinary?: boolean      // E.10 a/c  
        self_inflicted?: boolean     // CIV 6-25  
    },  
    choices: {  
        // Art. 18-grene  
        wants_refund?: boolean      // 18(1)(a)  
        wants_reroute_same_soonest?: boolean // 18(1)(b)  
        wants_reroute_later_choice?: boolean // 18(1)(c)  
    },  
    expenses: {
```

```

    meals?: number, hotel?: number, alt_transport?: number, other?: number
  },
                                // Art. 20(2)(a-c) + 20(3)
  already_refunded?: number      // (Art. 18) for at undgå dobbelt dækning
}

```

## Regler (kort)

### 1. Gatekeepers

- Hvis **self\_inflicted** === true → **ingen kompensation** (CIV 6–25 / Art. 10(b)); refusion/udgifter kan stadig bero på de konkrete forhold.
- Hvis **extraordinary** === true → **ingen kompensation** (Art. 19(10) a/c), men **refusion (18)** og **assistance/udgifter (20)** kan stadig være aktuelle.
- Hvis **notified\_before\_purchase** === true → **ingen kompensation** (Art. 19(9)).
- Tjek **exemptions** pr. land/tjeneste (regional/long domestic/international, Art. 2 B) og slå *kompensation/assistance* til/fra.

### 2. Refusion (Art. 18)

- **18(1)(a) – refund:**
  - *Aflysning / ikke igangsat rejse*: refundér **100 %** af prisen.
  - *Delvist gennemført rejse*: refundér **u-brugt** del.
  - *Return to origin pga. afbrydelse*: refundér **100 %**.
- Refusion **kan ikke** kombineres med kompensation **for samme del** af rejsen (dobbelt dækning undgås).

### 3. Kompensation (Art. 19)

- Beregn **forsinkelse ved endelig destination** (kun **EU-segmenter**, Art. 19(4)).
- Tærskler: **60–119 min** → **25 %**, **≥120 min** → **50 %** af prisgrundlaget.
- **Prisgrundlag** (Art. 19(3)):
  - Retur →  $\frac{1}{2}$  af billetprisen pr. vej.
  - Flere strækninger uden prisangivelse →  $\frac{1}{2}$ , eller fordel efter regler/algoritme (fx forholdstal af rejsetid).
  - Gennemgående billet → tag del/andel for forsinket del (eller hele, hvis kun ét ben).
- Nationale satser kan være **mere lempelige** (lokal ordning) → overskriv EU-minimum, hvis gunstigere.

### 4. Udgifter/assistance (Art. 20)

- **20(2)(a–c)** ved  $≥60$  min forsinkelse/aflysnings: måltider/forfriskninger, hotel+transport, transport ved blokeret tog.

- **20(3)**: alternative transporttjenester ved afbrudt forbindelse.
- Udgifter erstattes **faktisk udgift** (mod bilag). Udgifter tæller **uden procent**.

## 5. Samlet brutto-krav

```
gross_claim = refund_amount + compensation_amount +
reimbursable_expenses_total

gross_claim -= (already_refunded || 0)

gross_claim = max(gross_claim, 0)
```

## 6. Servicefee (25 %)

```
service_fee = round(gross_claim * 0.25, 2)

net_to_client = round(gross_claim - service_fee, 2)
```

*(Tilpas evt. med momsregler — i DK vil din fee normalt være momsbelagt over for klienten.)*

---

# Output (standardiseret)

```
type ClaimOutput = {

  breakdown: {

    refund: {basis:string, amount:number} // Art. 18

    compensation: {

      eligible:boolean, delay_minutes:number, pct:number,
      basis:string, amount:number, rule:"EU"|"National"
    } // Art. 19

    expenses: { meals:number, hotel:number, alt_transport:number, other:number,
    total:number } // Art. 20

    deductions: { already_refunded:number }

  },
  totals: {

    gross_claim:number,
    service_fee_pct:number, service_fee_amount:number,
    net_to_client:number,
    currency:string

  },
  flags: {

    extraordinary:boolean, self_inflicted:boolean, exemptions_applied:string[]||null,
    manual_review:boolean

  }
}
```

---

# Eksempler på dine fixtures

**Bemærk:** I eksemplerne antager vi *ingen* allerede udbetalt refusion/udgifter, og ingen nationale særregler, medmindre angivet.

## 1) SNCF TGV INOUI (PNR SNCF-7H2K9Q)

- Pris: **€89.00**
- Delay ved slut: **45 min** → **<60** → **ingen kompensation** (Art. 19).
- Rejsen gennemført → **ingen refusion** (18).
- Udgifter: 0.
- **Brutto = 0** → **Fee 25 % = 0** → **Netto = 0**.

**Output (kort):**

```
{  
  "breakdown": {  
    "refund": {"basis": "none", "amount": 0},  
    "compensation": {"eligible": false, "delay_minutes": 45, "pct": 0, "basis": "-", "amount": 0, "rule": "EU"},  
    "expenses": {"meals": 0, "hotel": 0, "alt_transport": 0, "other": 0, "total": 0},  
    "deductions": {"already_refunded": 0}  
  },  
  "totals": {"gross_claim": 0, "service_fee_pct": 25, "service_fee_amount": 0, "net_to_client": 0, "currency": "EUR"},  
  "flags": {"extraordinary": false, "self_inflicted": false, "exemptions_applied": null, "manual_review": false}  
}
```

---

## 2) DB ICE (PNR DB-ICE-AB12CD)

- Pris: **€109.90**
- Delay ved slut: **85 min** → **25 %** (Art. 19(1)(a)).
- Gennemført rejse → **ingen refusion** (18).
- Udgifter: 0.
- **Kompensation** =  $109.90 \times 0.25 = €27.475$  → afrund **€27.48**
- **Brutto = 27.48** → **Fee 25 % = €6.87** → **Netto = €20.61**.

**Output (kort):**

```
{
```

```

"breakdown": {
    "refund": {"basis":"none", "amount":0},
    "compensation": {"eligible":true, "delay_minutes":85, "pct":25,
"basis":"Art.19(3) whole fare", "amount":27.48, "rule":"EU"},
    "expenses": {"meals":0,"hotel":0,"alt_transport":0,"other":0,"total":0},
    "deductions": {"already_refunded":0}
},
"totals":
{"gross_claim":27.48,"service_fee_pct":25,"service_fee_amount":6.87,"net_to_client":20.61,"currency":"EUR"},

"flags":
{"extraordinary":false,"self_inflicted":false,"exemptions_applied":null,"manual_review":false}
}

```

---

### 3) DSB IC (PNR DSB-1Z9X8Y)

- Pris: **DKK 399.00**
- Delay ved slut: **102 min → 25 %** (Art. 19(1)(a)).
- Gennemført rejse → **ingen refusion**.
- Udgifter: 0.
- **Kompensation** =  $399.00 \times 0.25 = \text{DKK } 99.75$
- **Brutto** = 99.75 → **Fee 25 % = DKK 24.94** → **Netto = DKK 74.81**.

(Vil du teste 50 %, sæt delay ≥120 min i din mock.)

#### Output (kort):

```
{
"breakdown": {
    "refund": {"basis":"none", "amount":0},
    "compensation": {"eligible":true, "delay_minutes":102, "pct":25,
"basis":"Art.19(3) whole fare", "amount":99.75, "rule":"EU"},
    "expenses": {"meals":0,"hotel":0,"alt_transport":0,"other":0,"total":0},
    "deductions": {"already_refunded":0}
},
"totals":
{"gross_claim":99.75,"service_fee_pct":25,"service_fee_amount":24.94,"net_to_client":74.81,"currency":"DKK"},

"flags":
{"extraordinary":false,"self_inflicted":false,"exemptions_applied":null,"manual_review":false}
}
```

---

## Edge-cases Copilot skal håndtere

- **Refusion + kompensation sammen:** kun hvis de vedrører **forskellige dele** af rejsen (fx refusion for sidste, u-brugte ben; kompensation for første ben, der blev forsinket). Ellers afvis dobbelt dækning automatisk.
  - **Returrejse:** anvend  $\frac{1}{2}$  billetpris pr. vej (Art. 19(3)).
  - **Flere ben uden prisopdeling:** brug  **$\frac{1}{2}$ -reglen** eller fordel efter tids-/afstandsvægt. Log metode i bevispakken.
  - **International beyond EU:** ignorer ikke-EU segmenters forsinkelse i kompensation (Art. 19(4)), men refusion/udgifter kan stadig være relevante.
  - **Exemptions:** hvis land/tjeneste har undtagelse for Art. 19/20(2), slå kompensation/assistance fra og vis lokalt regelsæt (din matrix).
  - **Nationalt gunstigere regime:** overskriv EU-satser, hvis dine nationale tabeller viser højere udbetaling.
- 

## Mini-pseudokode (til service-laget)

```
def calculate_claim(input: ClaimInput) -> ClaimOutput:  
    # 1) Exemptions  
    ex = load_exemptions(input.country_code, scope_from_trip(input.trip))  
    comp_allowed = ex.art19_allowed  
    assist_allowed = ex.art20_2_allowed  
  
    # 2) Gatekeepers  
    if input.disruption.self_inflicted: comp_allowed = False  
    if input.disruption.extraordinary: comp_allowed = False  
    if input.disruption.notified_before_purchase: comp_allowed = False  
  
    # 3) Refusion  
    refund_amount = 0.0  
    if input.choices.wants_refund:  
        refund_amount = compute_refund_under_art18(input) # 100% / unused  
        part / return to origin  
  
    # 4) Kompensation  
    comp_pct = 0  
    comp_amount = 0.0
```

```

if comp_allowed and not overlaps_same_portion_as_refund(input):
    delay = input.disruption.delay_minutes_final or
calc_delay_EU_only(input.trip.legs)
    if delay >= 120: comp_pct = 50
    elif delay >= 60: comp_pct = 25
    # prisgrundlag jf. Art. 19(3)
    base = compute_price_basis(input)
    comp_amount = round(base * comp_pct/100, 2)
    # national override
    comp_amount = apply_national_overrides(comp_amount, input, delay)

# 5) Udgifter (Art. 20)
expenses_total = 0.0
if assist_allowed:
    expenses_total = sum(v or 0 for v in input.expenses.__dict__.values())

# 6) Samlet
gross = max(refund_amount + comp_amount + expenses_total -
(input.already_refunded or 0), 0)
fee = round(gross * 0.25, 2)
net = round(gross - fee, 2)

return build_output_json(...)

```

## Hvad du siger til Copilot (kort prompt-tekst)

- “Implementér **ClaimCalculator v1** som beskrevet.
- Læs PNR fra OCR og map til vores **/mocks** (SNCF, DB, DSB).
- Beregn **refusion (Art. 18)**, **kompensation (Art. 19)**, **udgifter (Art. 20)** og lav **samlet brutto, servicefee 25 %, netto til klient**.
- Respekter **exemptions** pr. land/tjeneste og **gatekeepers** (CIV/force majeure/forudvarsling).
- Returnér **ClaimOutput JSON** + render en kort **UI-breakdown** (Refusion, Kompensation, Udgifter, Fee, Netto).
- Tilføj unit tests for de tre fixtures, inkl. en DSB-case  $\geq 120$  min.”

Hvis du vil, kan jeg også udlevere et helt **unit-test-sæt** (Jest/PHPUnit) med forventede outputs for de tre billetter.

