

## 06 — File Delivery

**Audience:** Tenant IT. **Prasyarat:** [01-overview.md](#), [02-authentication.md](#), [03-callbacks-api.md](#). Doc ini adalah **pendalaman** §6 doc #3 — fokus penuh ke pipeline download, security posture, dan recovery.

Fitur ini (Phase 5.5) memungkinkan Tenant IT mengirim file (invoice PDF, foto unit, dokumen kontrak, dll) ke customer lewat channel aslinya. Tenant IT **tidak** upload file ke EMBAN secara langsung — Tenant IT meng-host file di server sendiri (atau CDN), lalu beri URL-nya di field `files[]` saat POST callback. EMBAN yang akan men-download, validasi, dan meneruskan ke customer.

---

### 1. Kenapa download-link, bukan upload langsung?

Alternatif (tenant upload file ke EMBAN via multipart) punya tiga masalah:

- **Bandwidth EMBAN** ditanggung untuk ingress dari N tenant → storage sementara.
- **Autentikasi + size cap** harus di-enforce di edge HTTP layer, bukan hanya di engine.
- **Tenant biasanya sudah punya file hosting** (S3, CDN, internal file server).  
Menduplikasi upload ke EMBAN = double storage.

Download-link pattern: EMBAN fetch dari URL Tenant IT, validasi, forward. Tenant IT tidak perlu tahu detail channel (Telegram media upload, WA multimedia, SMTP attachment). EMBAN yang handle.

Security posture disebut internal **Opsi X — tanpa virus scanning**. Aman karena: HTTPS only + hostname allowlist + MIME allowlist + content sniff + size cap. Bukan air-gap level, tapi cukup untuk use case B2C dengan file yang Tenant IT sendiri sudah produce (bukan user-uploaded content).

---

### 2. Flow singkat

Tenant IT backend

| POST callback /api/v1/callbacks/:submission\_id

| body.files = [{url, mime\_type, filename, expected\_size\_bytes}]



#### EMBAN — callbacks route

- | text message dulu di-deliver ke customer (§6 doc #3)

- | lalu untuk tiap file:



#### EMBAN — file-delivery orchestrator

- | fetch allowlist tenant (fresh per-request)



#### EMBAN — file-download (validasi 6 lapis, lihat §4)

- | tulis ke {FILE\_DELIVERY\_DIR}/{tenant\_id}/{ulid}.{ext}



#### Channel adapter

- | upload ke TG/WA / attach ke email



#### Customer menerima file

- | (text message sudah duluan)



#### Scheduler cleanup (hourly tick, 24h TTL)

- | delete file dari disk, set audit status=expired



### 3. Request shape (recap dari doc #3)

Callback body field `files[]` — max 10 entry per callback:

```
{
  "status": "success",
  "template_key": "booking_confirmed",
  "data": { "booking_id": "BK-2026-0412" },
  "files": [
    {
      "url": "https://files.tenant.example/invoice/BK-2026-0412.pdf",
      "filename": "invoice.pdf",
      "mime_type": "application/pdf",
      "expected_size_bytes": 524288
    }
  ]
}
```

| Field                  | Tipe              | Wajib | Catatan   |
|------------------------|-------------------|-------|---|
| <code>url</code>       | string URL        | ya    | <b>HTTPS wajib</b> ,<br>hostname harus di<br>allowlist      |
| <code>mime_type</code> | string            | ya    | MIME deklarasi —<br>harus di<br><code>MIME_ALLOWLIST</code> |
| <code>filename</code>  | string ≤ 255 char | tidak | Nama file yang<br>terlihat customer.                        |

| Field                            | Tipe             | Wajib | Catatan  |
|----------------------------------|------------------|-------|--|
|                                  |                  |       | Default:<br>auto-generated<br><code>{ulid}.{ext}</code>                          |
| <code>expected_size_bytes</code> | integer $\geq 0$ | tidak | Pre-check sebelum fetch.<br>Direkomendasikan biar file oversized di-reject cepat |

## 4. Security posture — enam lapis validasi

Urutan pengecekan (first-fail short-circuits):

1. **Declared MIME** harus ada di `MIME_ALLOWLIST` (lihat §5).
2. **URL** parse valid.
3. **Protocol** = `https:` (HTTP ditolak).
4. **Hostname** (lowercase) exact-match di tenant allowlist (lihat §6). Tidak ada wildcard, tidak ada port bypass.
5. **Size cap** — deklarasi `expected_size_bytes` > 10 MB → reject. HEAD pre-check `Content-Length` > 10 MB → reject. **Streaming byte counter** — abort kalau body overrun (HEAD bisa berbohong).
6. **Content sniff** — `file-type` library baca magic bytes dari buffer, dibandingkan dengan declared MIME. Mismatch → reject. (Exception: `text/plain` dan `text/csv` tidak punya magic bytes, sniff di-skip.)

Kalau semua 6 lulus: file ditulis ke disk dengan ULID name + extension dari declared MIME.

## 5. MIME allowlist

Hardcoded di EMBAN (`src/services/file-download.ts`). Nilai default yang sudah approved:

| MIME                         | Ext | Catatan      |
|------------------------------|-----|--------------|
| <code>application/pdf</code> | pdf | dokumen umum |

| MIME  | Ext  | Catatan                                   |
|---|------|---|
| image/png   | png  |   |
| image/jpeg  | jpg  |   |
| image/webp  | webp |   |
| image/gif   | gif  |   |
| application/msword  | doc  | Word 97-2003                              |
| application/vnd.openxmlformats-officedocument.wordprocessingml.document | docx | Word modern (sniff jadi application/zip)  |
| application/vnd.ms-excel  | xls  | Excel 97-2003                             |
| application/vnd.openxmlformats-officedocument.spreadsheetml.sheet       | xlsx | Excel modern (sniff jadi application/zip) |
| text/csv  | csv  | sniff di-skip                             |
| text/plain  | txt  | sniff di-skip                             |
| application/zip   | zip  |   |

**Belum didukung:** video (mp4, mov), audio (mp3, opus, wav), PowerPoint (ppt/pptx). Kalau butuh, ajukan ke tim EMBAN — perlu update allowlist + testing per channel.

## 6. Konfigurasi allowlist per-tenant

Field di DB: `tenants.file_download_allowlist` — JSON array of hostname string (lowercase, exact-match).

Contoh isi:

```
["files.tenant.example", "cdn.tenant.example", "s3.amazonaws.com"]
```

- **Empty / null** → **semua file di-reject** dengan `allowlist_empty`. Fitur opt-in per tenant.
- **Exact match only** — `files.tenant.example` **tidak** match `sub.files.tenant.example`. Daftar subdomain yang perlu di-allow.
- **Port** tidak di-match — `files.tenant.example:8443` = hostname `files.tenant.example`.
- Dibaca **fresh per-request** — tenant bisa tambah hostname lalu test di callback berikutnya, tanpa restart EMBAN.

**Cara Tenant IT minta allowlist di-update:** via onboarding tenant CS, atau escalate ke tim EMBAN (belum ada self-service API di v1).

## 7. Aturan size & timeout

| Parameter                    | Default  | Env   | Catatan   |
|------------------------------|----------|---|---|
| Max size per file            | 10 MB    | <code>FILE_DELIVERY_MAX_BYTES</code>                | Streaming cap — abort kalau overrun             |
| Max files per callback       | 10       | schema callback                                     | Hard-cap; kirim lebih dari 10 di split callback |
| Connect timeout (HEAD + GET) | 10 detik | <code>FILE_DELIVERY_CONNECT_TIMEOUT_MS</code>       | Per attempt                                     |
| Read timeout (GET body)      | 30 detik | <code>FILE_DELIVERY_READ_TIMEOUT_MS</code>          | Overall abort                                   |
| TTL on EMBAN disk            | 24 jam   | <code>FILE_DELIVERY_TTL_HOURS</code>                | Cleanup scheduler GC                            |
| Cleanup tick                 | 60 menit | <code>FILE_DELIVERY_CLEANUP_INTERVAL_MINUTES</code> | Hourly sweep                                    |

**TTL 24 jam artinya:** kalau customer tidak menerima file dalam 24 jam setelah callback sukses (mis. customer telegram-nya offline lama), file tidak bisa di-resend otomatis. Tenant IT harus re-POST callback baru. EMBAN tidak retry delivery.

## 8. Error codes & audit

Setiap attempt (sukses / gagal) menulis row ke `file_deliveries` table. Status value:

| Status                       | Artinya  |
|------------------------------|--|
| <code>downloaded</code>      | Lolos 6 lapis validasi, tersimpan di disk, <b>belum</b> dikirim ke channel   |
| <code>delivered</code>       | Sudah sampai ke customer via channel adapter                                 |
| <code>rejected</code>        | Gagal di policy (URL / MIME / allowlist / size) — tidak akan retry           |
| <code>fetch_failed</code>    | Gagal di transport (HTTP error, timeout, write_failed) — juga tidak di-retry |
| <code>delivery_failed</code> | Download sukses tapi channel adapter throw saat forward                      |
| <code>expired</code>         | Scheduler cleanup men-GC file setelah TTL                                    |

Reject reason detail yang tersimpan di kolom `error_code`:

| <code>error_code</code>       | Lapisan yang gagal   |
|-------------------------------|--|
| <code>mime_not_allowed</code> | MIME di luar allowlist                                     |
| <code>bad_url</code>          | URL tidak parseable  |
| <code>not_https</code>        | URL bukan <code>https://</code>                            |
| <code>host_not_allowed</code> | Hostname tidak ada di tenant allowlist                     |
| <code>allowlist_empty</code>  | Tenant belum konfigurasi allowlist                         |
| <code>size_exceeds_cap</code> | Deklarasi / HEAD / streaming overrun > 10 MB               |
| <code>head_failed</code>      | HEAD request error (fallback ke GET, tidak fatal biasanya) |
| <code>fetch_failed</code>     | GET 4xx/5xx atau network error                             |

| <b>error_code</b>             | <b>Lapisan yang gagal</b>  |
|-------------------------------|--|
| <b>timeout</b>                | Abort karena connect/read timeout                                      |
| <b>content_sniff_unknown</b>  | Declared MIME butuh magic bytes tapi buffer tidak match pattern apapun |
| <b>content_sniff_mismatch</b> | Declared MIME ≠ sniffed MIME   |
| <b>write_failed</b>           | Disk full / permission error   |

Tenant IT **tidak** akses langsung tabel ini — kalau butuh audit, escalate ke tim EMBAN dengan **submission\_id** atau **tenant\_id** + window waktu.

---

## 9. Response callback dengan file report

EMBAN kembalikan per-file outcome di body callback response:

```
{
  "ok": true,
  "delivered": true,
  "customer_channel": "whatsapp",
  "submission_id": "01J1ZXK7...",
  "files": {
    "delivered": 2,
    "rejected": 1,
    "failed": 0,
    "details": [
      { "url": "https://files.tenant.example/a.pdf", "status": "delivered" },
      { "url": "https://files.tenant.example/b.png", "status": "delivered" },
```



```

{
  "url": "http://bad.example/c.pdf",

  "status": "rejected",

  "error": "not_https"

}

]

}

}

```

**Penting: text message selalu di-deliver dulu**, tidak tergantung status file. Kalau semua file gagal, customer tetap terima text message dari template. Tenant IT bisa kirim ulang callback hanya dengan `files[]` yang gagal saja (gunakan template minimal atau `template_key` yang sesuai — misal template "lampiran menyusul").

## 10. Channel-specific behavior

| Channel         | Cara delivery  | Batasan platform (eksternal)  |
|-----------------|--|---|
| <b>WhatsApp</b> | Upload via Baileys multimedia (image/document/audio/video API)                                     | WA sendiri batasi ~16 MB untuk video, ~100 MB untuk document. Kami cap 10 MB, jauh di bawah.                          |
| <b>Telegram</b> | Upload via <code>sendDocument</code> / <code>sendPhoto</code> / <code>sendVideo</code> sesuai MIME | TG batasi 50 MB untuk bot upload. Kami cap 10 MB.   |
| <b>Email</b>    | Attachment di SMTP2Go (SMTP)   | Provider biasanya batasi 25 MB total per email. Kami cap 10 MB per file × max 10 files = 100 MB total; potensi reject |

| Channel | Cara delivery | Batasan platform (eksternal)                       |
|---------|---------------|--|
|         |               | di SMTP layer kalau total attachment terlalu besar |

Kalau ingin support payload > 10 MB di future, butuh update `FILE_DELIVERY_MAX_BYTES` + test ulang per channel (WA dan email akan jadi bottleneck lebih dulu daripada TG).

---

## 11. Rekomendasi hosting file dari sisi Tenant IT

- **Gunakan HTTPS** dengan sertifikat yang di-trust OS (Let's Encrypt, DigiCert, dll). Self-signed cert akan gagal di TLS verify.
  - **Stable hostname** — hindari URL yang hostname-nya berubah (preview URL, ephemeral container URL). Setiap kali hostname baru, perlu update allowlist.
  - **Pre-signed URL** (S3, R2, GCS) OK asal hostname-nya fixed (`s3.amazonaws.com`, bukan `unique-id.s3.amazonaws.com`). Kalau pakai per-object subdomain, allowlist tidak akan cover.
  - **Serve dengan Content-Length header benar** — HEAD pre-check mengandalkan ini untuk early-reject oversized file, menghemat bandwidth.
  - **Jangan pakai redirect** — EMBAN `fetch()` default mengikuti redirect, tapi validasi URL hanya cek URL asli. Re-konfirmasi hostname final kalau pakai shortener/redirector.
  - **File TTL di sisi Tenant IT** — minimal 1 jam (buffer retry) + 24 jam untuk in-transit. Rekomendasi: URL valid minimal 2 jam.
  - **Naming**: isi `filename` dengan nama yang rapi untuk customer. Kalau di-skip, customer terima nama `{ulid}.{ext}` — kurang user-friendly.
- 

## 12. Checklist go-live

- ☐ Hostname file server sudah di `tenants.file_download_allowlist` (koordinasi via onboarding).
- ☐ Semua URL HTTPS + sertifikat valid.
- ☐ MIME type diisi benar (bukan `application/octet-stream` untuk PDF).
- ☐ `expected_size_bytes` diisi untuk early-reject oversized file.
- ☐ File hosting stabil + TTL  $\geq$  2 jam.
- ☐ Smoke test: 1 file sukses, 1 file size-overflow, 1 file MIME mismatch, 1 file hostname not allowlisted.

- ☐ Per-file report di response callback di-log untuk audit sendiri (supaya Tenant IT tahu file mana yang gagal, tanpa escalate ke tim EMBAN).

Lanjut: **#4 Ticketing API** — untuk flow "forward to human" kalau agent CS eskalasi.