# Document 1 — Technical Description (End‑to‑End)

**Version:** 1.0  
**Date:** 2025‑09‑25  
**Scope:** ServiceNow PDI  
**SN Compatibility:** Zurich (current), Yokohama, Xanadu  
**Solution Version Referenced:** DAWA-Loc v1.1.8

## Overview & Business Value

This solution enriches and validates cmn\_location records in ServiceNow with authoritative Danish address data from **DAWA – Danmarks Adressers Web API**. It standardizes user-entered addresses, stores the DAWA IDs (adresse + adgangsadresse), kommune metadata, and enforces on‑form data quality (mandatory **Country**).

## High‑Level Architecture

* **Client/UI**: Location form with UI Actions (e.g., *DAWA → Normalize*) and a UI Policy that makes **country** mandatory on form.
* **Server**: Script Include **DAWAAddressClient** encapsulates outbound REST calls to DAWA and maps results to cmn\_location fields.
* **Integration**: Outbound REST Message **“DAWA Address API”** with verbs calling DAWA HTTPS endpoints.
* **Data Store**: cmn\_location table extended with fields:
  + u\_dawa\_adresse\_id (string, 64)
  + u\_dawa\_adgangsadresse\_id (string, 64)
  + u\_kommune\_kode (string, 40)
  + u\_kommune\_navn (string, 100)

## Data Flow (Normalization)

1. User clicks *DAWA → Normalize* on a Location record.
2. Server calls DAWA endpoints in this order (using best‑effort filters):
   * **Autocomplete** (optional): /autocomplete with type=adresse|adgangsadresse, q, and optionally postnr or kommunekode.
   * **Search** (fallbacks): /adresser and/or /adgangsadresser using structured params (vejnavn, husnr, postnr, kommunekode) or free‑text q with struktur=mini.
   * **Resolve by ID** (final): /adresser/{id} or /adgangsadresser/{id} when a candidate is selected.
3. Mapping writes:
   * street = composed “vejnavn husnr [etage dør]”
   * zip = postnr
   * city = postnrnavn
   * country = “Denmark” (enforced mandatory on form)
   * DAWA metadata IDs and kommune fields as above.
4. Info message summarises outcome: normalized, already\_normalized, or no\_match.

## Field Mapping Details

* **Street**: prefer adresse.vejnavn + husnr → fallback to adgangsadresse.vejnavn + husnr.
* **City/ZIP**: from postnr/postnrnavn (address) else from access‑address postnr/postnrnavn.
* **Kommune**: kommunekode (e.g., 0101) and display name (e.g., København).
* **IDs**: adresse.id → u\_dawa\_adresse\_id; adgangsadresse.id → u\_dawa\_adgangsadresse\_id.

## Error Handling & Logging

* All integration messages are written with prefixes like **DAWA‑Loc**, **DAWA‑Test**, **DAWA‑OneShot** in *System Logs → All*.
* Typical reasons returned to the form: no\_match, normalized, already\_normalized, or an exception:<detail>.

## Configuration (sys\_properties)

* x\_bgb.dawa.base\_url = https://api.dataforsyningen.dk
* x\_bgb.dawa.country\_allowlist = DK
* x\_bgb.dawa.autocomplete.enabled = true|false (server‑side assist; optional)
* x\_bgb.dawa.validate.on\_update = true|false (if enabled, normalize on update)
* x\_bgb.dawa.backfill.batch\_size = integer for scheduled backfill (optional)

## Security & Performance

* **Security**: Public GET endpoints over HTTPS; no API key required.
* **Platform**: Use setWorkflow(false)/autoSysFields(false) in fix/backfill contexts only.
* **Performance**: Prefer struktur=mini, add postnr/kommunekode when known, debounce client actions.

## Modules & Automation

* **Module**: *Locations (DAWA)* → opens cmn\_location.list.
* **UI Actions**: *DAWA → Normalize* (server‑side), optionally *Open in DAWA* (new tab to the selected ID).
* **UI Policy**: Country mandatory on form (does **not** block imports/apis).
* **(Optional) Scheduled Job**: *DAWA Backfill* to cleanse existing records in batches.

## External Sources (Key Endpoints)

* API index: https://dawadocs.dataforsyningen.dk/dok/api
* Addresses: https://dawadocs.dataforsyningen.dk/dok/api/adresse
* Access addresses: https://dawadocs.dataforsyningen.dk/dok/api/adgangsadresse
* Combined Autocomplete: https://dawadocs.dataforsyningen.dk/dok/api/autocomplete
* BBR notice (DAWA): https://dataforsyningen.dk/asset/PDF/KF-nyheder/news\_dataforsyningen-bbr.pdf

# Document 2 — Instruction for Use (Step‑by‑Step)

**Version:** 1.0  
**Audience:** No‑code users (Location editors)

## A. Before You Start (one‑time)

1. Open **All → System Properties → All Properties**, verify:
   * x\_bgb.dawa.base\_url = https://api.dataforsyningen.dk
   * x\_bgb.dawa.country\_allowlist = DK
2. Navigate **All → Locations (DAWA)** to work with locations.

## B. Create or Fix a Location (the happy path)

1. Click **New**.
2. Fill:
   * **Country** = *Denmark* (mandatory on form).
   * **Street** = e.g., *Borgergade 17A*.
   * **ZIP** = *1300*.
   * **City** = *København K*.
3. Click **Save** (stay on the record).
4. Click **DAWA → Normalize**.
   * You should see an info message like: *normalized* or *already\_normalized*.
   * The DAWA IDs and Kommune fields will populate automatically.
5. Click **Update** when you’re done.

### Notes

* If *no\_match* appears, double‑check spelling (e.g., use *17A* vs *17*) and include **ZIP**.
* You can open the selected DAWA record in a new tab via *Open in DAWA* (if shown).

## C. Edit an Existing Location

1. Open the record.
2. Adjust Street/ZIP/City; keep **Country = Denmark**.
3. Click **Save → DAWA → Normalize** again to re‑validate.

## D. Where to See Logs

* **System Logs → All**. Filter by **Message** contains DAWA-.

# Document 3 — DAWA Improvements (Backlog)

**Version:** 1.0  
**Status:** Proposed

## A. Client UX

* **Autocomplete (web widget)**: add client‑side suggestions using DAWA Autocomplete component for even faster data entry.  
  *Pros:* great UX; fewer typos. *Cons:* requires Client Script/Widget.  
  *Ref:* https://dawadocs.dataforsyningen.dk/dok/api/autocomplete
* **“Open in DAWA”**: always show a link icon to open the chosen address/access‑address ID in a new tab.

## B. Data Model Enhancements

* **Latitude/Longitude**: Map DAWA x/y (WGS84 or EPSG:25832 converted) into latitude/longitude on cmn\_location for GIS/Workplace apps.
* **Unit detail**: Persist etage and dør into custom fields (read‑only) when available.

## C. Reliability & Scale

* **Backfill Scheduler**: nightly job to normalize all missing/dirty records in batches (x\_bgb.dawa.backfill.batch\_size).
* **Rate Limiting/Retry**: introduce exponential backoff for HTTP 429/5xx; add guardrail to cap requests/sec.

## D. Governance & Observability

* **Health Check**: daily smoke test hitting /adresser/{id} for a known ID; alert on failure.
* **Dashboards**: KPIs (normalized %, no\_match %, average response time), with drill‑downs.

## E. Notifications (Requested by Brian)

* **Daily status email** to *briangbuur.work@gmail.com* at 08:00 CET summarizing: normalized/new today, no\_match count, last error, and link to a list report.  
  *Pros:* proactive visibility; *Cons:* requires outbound email configuration.  
  *Implementation sketch:* Scheduled Script Execution → gather metrics (GlideAggregate on cmn\_location), compose mail (GlideEmailOutbound), link to cmn\_location\_list.do?sysparm\_query=....

## F. Compatibility & Packaging

* Publish as an **Update Set** or **Scoped App** with properties, UI artifacts, and Script Include.

## Self‑Update (Brian’s section)

Fill in or adjust the variables below to reflect your environment.

DOC\_VERSION: 1.0  
ENV:  
 INSTANCE\_URL: <your PDI URL>  
 FAMILY: Zurich  
 APP\_SCOPE: x\_bgb  
 MODULE\_LABEL: "Locations (DAWA)"  
PROPERTIES:  
 BASE\_URL: https://api.dataforsyningen.dk  
 COUNTRY\_ALLOWLIST: DK  
 VALIDATE\_ON\_UPDATE: <true|false>  
 AUTOCOMPLETE\_ENABLED: <true|false>  
IMPROVEMENTS:  
 DAILY\_STATUS\_EMAIL:  
 ENABLED: false  
 TO: briangbuur.work@gmail.com  
 TIME\_CET: "08:00"  
 METRICS: [normalized\_today, no\_match\_today, last\_error]