

Deep Harbor Member Management System

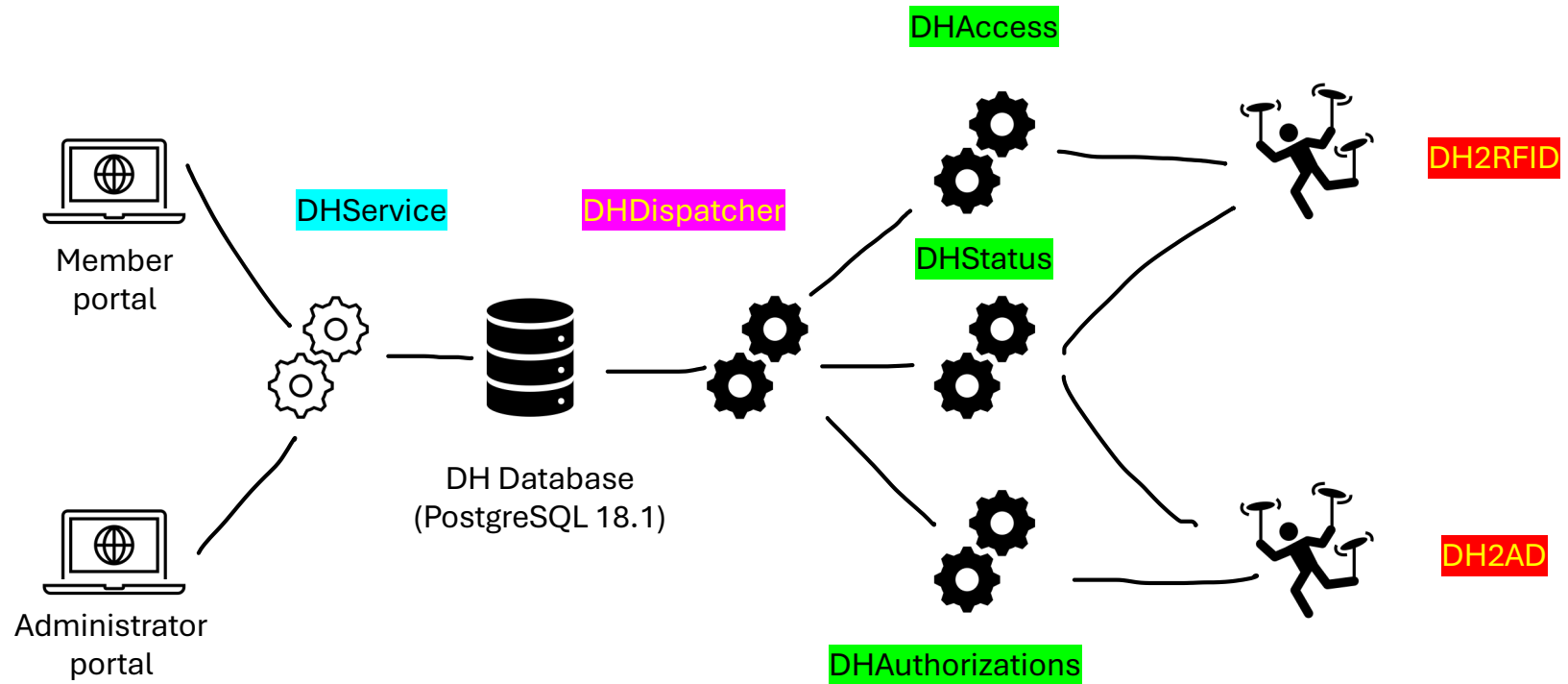
December 2025

Wait, hold up. What's with the name?

A deep-water harbor is where ships would travel and people from all over the world could meet and exchange ideas.

The name is meant to invoke the idea of collaboration and community made up of unique individuals.

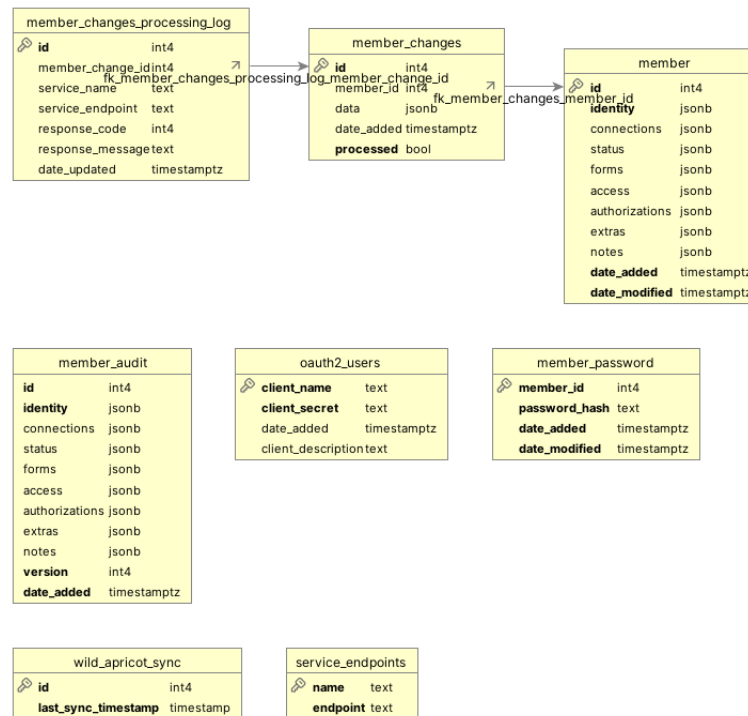
Design



Deep Harbor Database

- Uses PostgreSQL 18.1 (as of December 2025)
- Minimum tables, uses JSONB for many of the “important” fields
 - Allows for maximum flexibility, we don’t know what we don’t know going forward
- PostgreSQL is a relational database but there are relatively few actual relations in this schema that warrant the complexity of a 3rd or 4th normal-form design.
- Uses plpgsql functions sparingly to codify certain logic that could be used in multiple places

Deep Harbor Database Schema



Deep Harbor Codebase

- Python
 - Well represented among PS1 members
- PLPGSQL (PostgreSQL stored procedures)
 - Database-specific programs for centralized logic

DHService

- Provides all services to “front end” systems, like the member and administrator portals.
- Uses oauth2 for authorization for the services
 - Applications cannot simply invoke calls without a predefined account and secret stored in the database
 - This is specifically for *applications*, not users
- All API calls are versioned so new calls can be created without impacting existing systems (e.g., “/**v1/member/access**”)

DHDispatcher

- Listens for changes on the **member** table
- Forwards changes to other services (DHAccess, DHAuthorizations, DHStatus)
- Fault-tolerant in both directions
 - Will retry to send messages to services if they're unavailable
 - Will handle outstanding changes when started, in order of when they were made
- Reads endpoints from the **service_endpoints** table.
 - New services can be added without requiring a restart

“Change Services” (a.k.a. “business” rules)

- DHIdentity
 - Handles anything related to a change to the member’s identity, including creating an account for them in Active Directory via DH2AD
- DHAccess
 - Handles updates for things like RFID tags
 - Invokes DH2RFID to communicate with the board hardware
- DHAuthorizations
 - Handles updates for authorizations
 - Invokes DH2AD to update the appropriate groups in Active Directory
- DHStatus
 - Handles status changes (e.g., new member, cancelled membership, etc.)
 - Invokes both DH2RFID and DH2AD to enable or disable all access of a member

“Worker Services”

- DH2AD
 - Only knows how to talk to Active Directory to handle changes to a member account
- DH2RFID
 - Talks to the RFID hardware to enable or disable tags for a member