XII SPD Collaboration meeting

Hardware DB: DAQ Case

Rolen Magkaev Fedor Prokoshin

JINR, November 2026

Table of Contents

1. Hardware Database (HWDB)
2. Hardware Database (device parameters)
3. Device identification
4. HWDB application: main features
5. Technology stack
6. System architecture
7. IAM (Identity and Access Management)
8. HardwareDB Authentication
9. Authentication and access control
10. Data model: what is stored
11. Catalog hierarchy and inheritance
12. Step 1: prepare the catalog
13. Step 2: register a device
14. Step 3: find and maintain equipment

Hardware Database (HWDB)

- A catalog of hardware components and equipment of the SPD detector
- It should contain information about detector elements, electronic parts, cables, etc. including their position in the setup
- Every item has a unique ID assigned to it and variable set of parameters depending on its type and model
- HWDB should help maintain detector systems and especially helpful in knowledge transfer between team members.
- It should be accessible both through the web-GUI and software API, from the programs and other information systems (DCS, Calibration)
- Filling of the hardware database should take place gradually, and updates will be rare
- The requirements for the speed of recording information are low
- The reading of the information for a large number of devices expected, for example for run preparation

Hardware Database (device parameters)

Devices should have few common parameters not dependent on type

- Serial Number (permanent)
- Component ID, specify in its position or role at SPD
- Current status (Ok, Bad, Problems, Repaired, Unknown etc.)
- Dates related to the lifecycle of device
  - production date
  - acquisition date
  - last status change date
  - or a full history of status changes (TBD)
- Responsible person or division ID
- Inventory/Stock number

Device identification

There are 2 ways to identify a device available in the catalog

- Hardware ID (HWID):
  - Identifies specific item, (like part number + serial number)
  - Permanently assigned to the device, device parameters are linked to it
- Component ID:
  - Assigned to the device depending on its position or role
  - In general can be changed, for example when device is connected to other channel or moved to different subsystem

HWDB application: main features

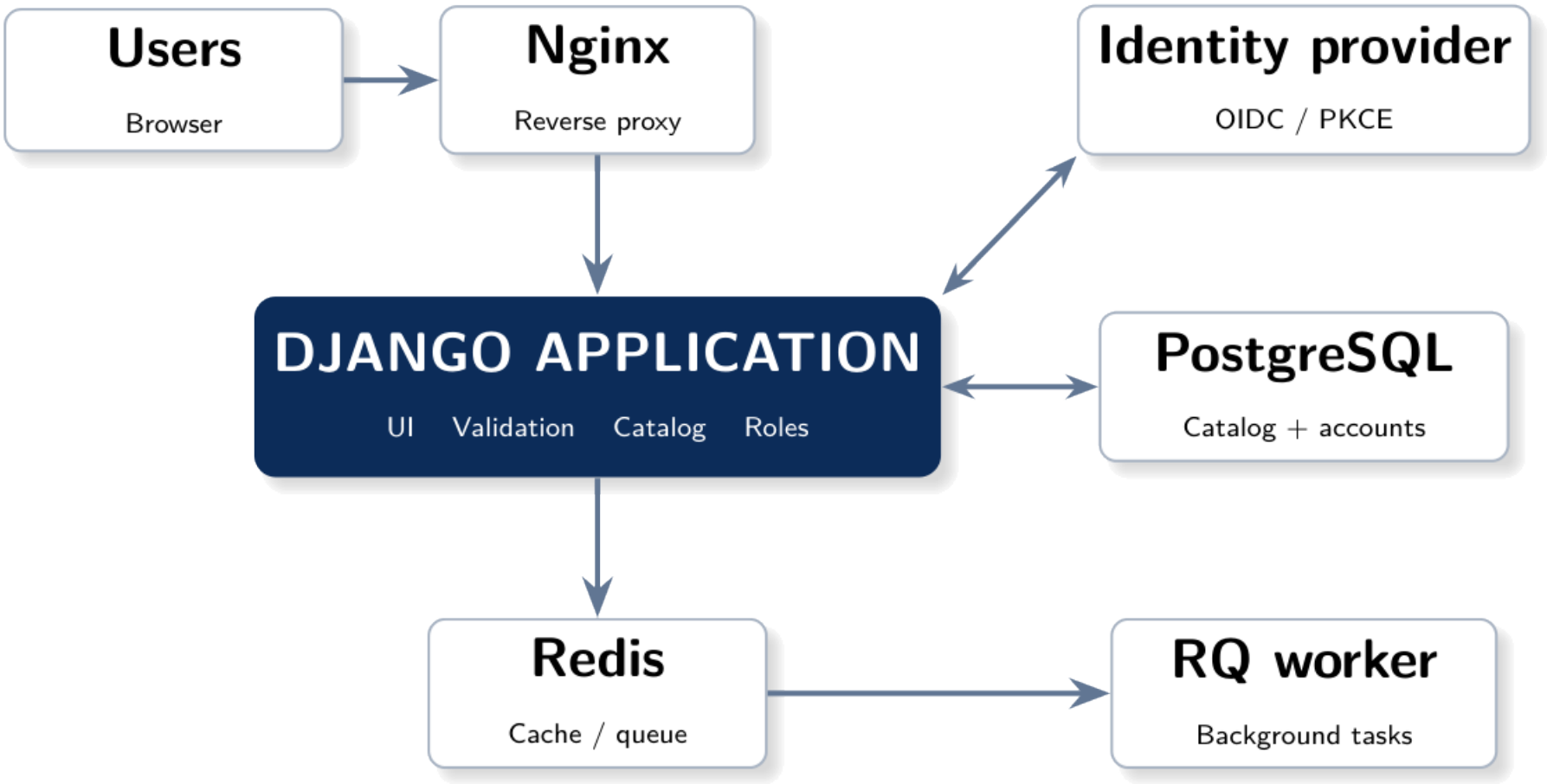
- A shared catalog of physical equipment and components of the SPD detector
- Combines equipment identification, classification and technical properties in a single web interface
- Groups organize equipment by subsystem, category or model; individual devices contain their own values
- The catalog supports gradual data entry, searching, editing and maintenance by authorized users
- RU/EN interface and local-time display support a distributed collaboration

Technology stack

- Backend: Python, Django 6 and Pydantic for structured input validation
- Database: PostgreSQL 18 with Psycopg 3 connection pooling
- Interface: Django templates, HTMX, Alpine.js and Tailwind CSS
- Deployment: Docker Compose, Nginx, Gunicorn and Uvicorn ASGI workers
- Infrastructure: Redis, RQ workers, GitLab CI, health checks and optional Sentry
- Security: IAM [1] OpenID Connect authentication and role-based access control

[1] IAM is a main authentication SPD JINR permission and access control system, which is now integrated in many of our projects

System architecture



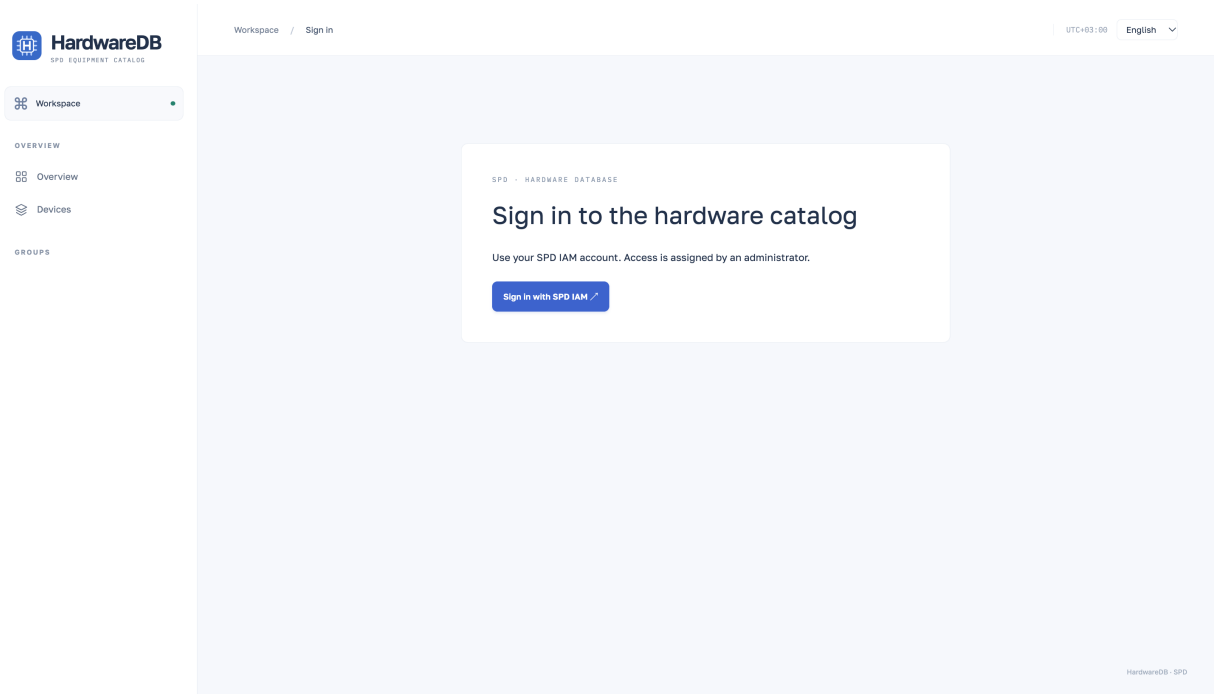
IAM (Identity and Access Management)

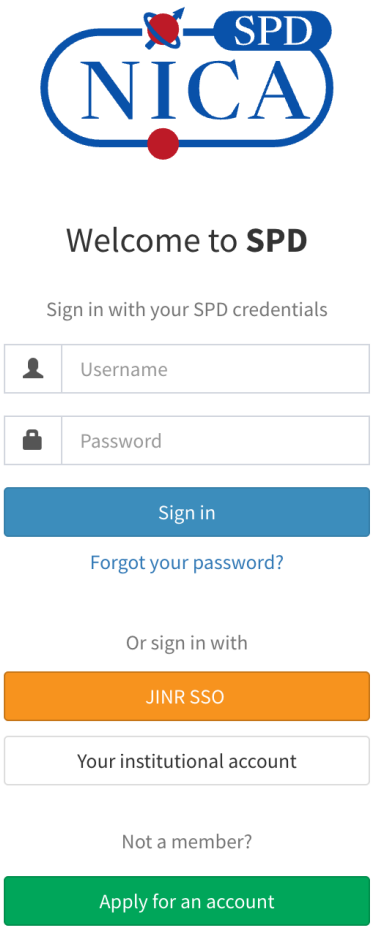
SPD IAM (Identity and Access Management) manages user identities and access to the digital services of the SPD experiment. It allows collaboration members to use a single account to authenticate across systems.

Main components

- **User accounts** contain the identity information of collaboration members
- **Groups and organizational membership** allow users to be organized and membership information to be shared with services
- **Identity server** verifies users during login and confirms their identities to connected applications
- **Connected applications**, such as HardwareDB, trust the authentication result provided by IAM
- **Tokens** are signed data received and verified by an application after a successful login

HardwareDB Authentication via SPD IAM





Authentication and access control

- Users authenticate with OpenID Connect using the Authorization Code flow and PKCE
- The application verifies token signature, issuer, audience, expiration and nonce
- Viewer: browse and search; editor: create, edit and delete catalog records
- Administrator: manage access; the effective role also depends on account status and SPD membership
- Session renewal uses server-side encrypted refresh tokens; tokens are not stored in the browser
- Frozen or revoked access blocks protected operations

Data model: what is stored

- Group: a catalog node with an optional parent; groups form a hierarchy
- GroupProperty: a field definition with name, type, units, limits and optional default
- Device: name, unique Hardware ID, optional unique Component ID and a group
- DeviceProperty: individual values stored in type-specific tables; ActivityEvent records catalog actions
- Each device belongs to one group, while a group can contain many devices

Catalog hierarchy and inheritance

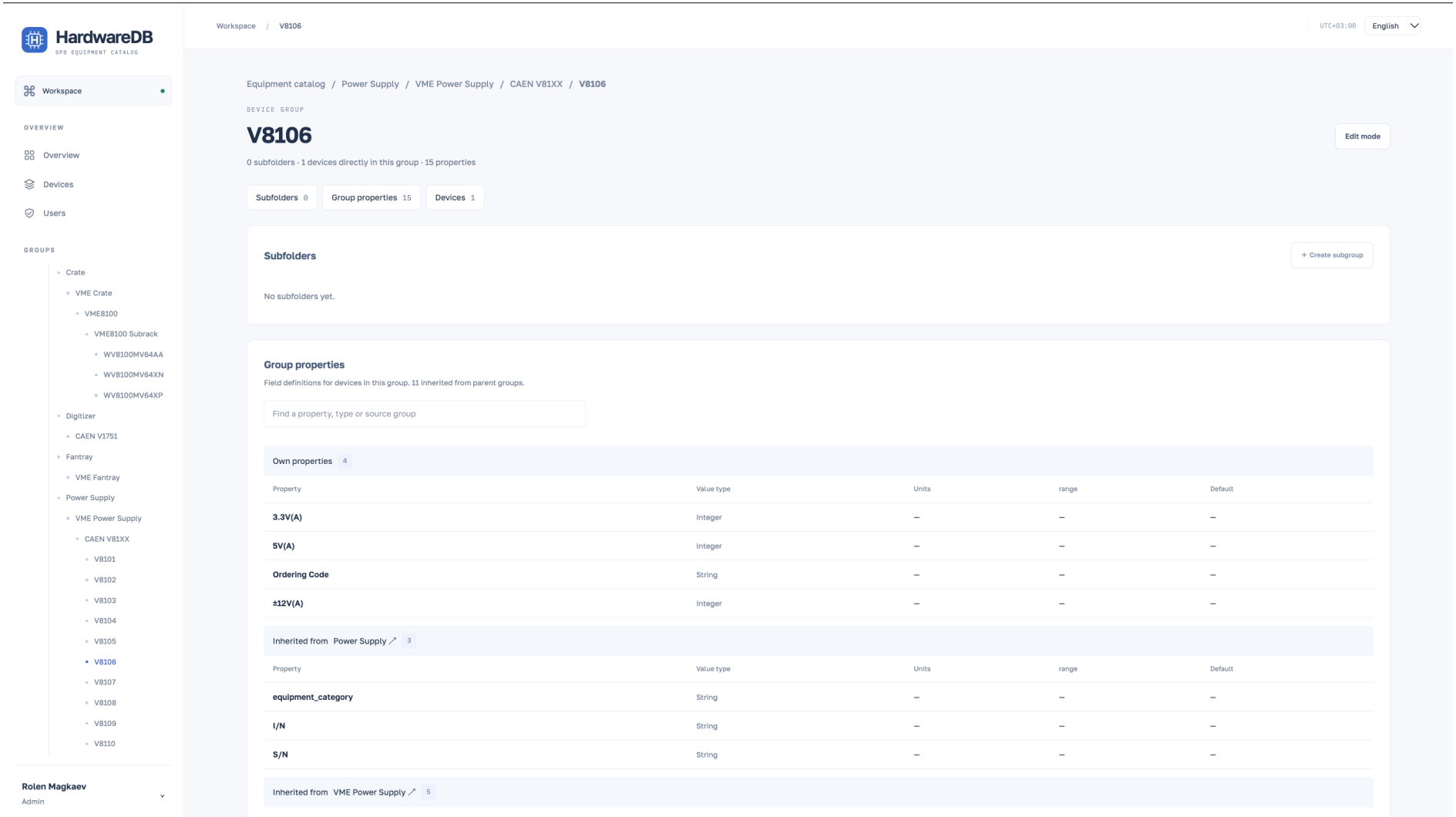
- The catalog groups equipment into categories. For example: Crate, Digitizer, Fantray and Power Supply
- Groups are organized from general categories to specific models, for example: Power Supply → VME Power Supply → CAEN V81XX → V8101
- Common properties can be defined in a parent group and inherited by its descendants
- Each subgroup can add its own properties; a device uses both inherited and local definitions



Step 1: prepare the catalog

- An editor creates a root group or selects an existing group and adds a subgroup
- Choose meaningful levels: subsystem $\rightarrow$ equipment category $\rightarrow$ device model
- Define fields using integer, long integer, float, text, boolean, date, datetime, IP or enum
- For each field, specify units, numerical bounds and a default when appropriate
- For enum fields, define allowed options; the default must be one of those options
- The group can be created first and its property schema extended later

Catalog example: V8106 group



Step 2: register a device

- Select the device group; the form displays its inherited and local properties
- Enter a device name and unique Hardware ID; enter a unique Component ID if assigned
- Fill in individual property values, or use defined defaults where applicable
- The application validates formats, numeric bounds, IP addresses and enum options
- Unknown fields and duplicate identifiers are rejected; errors are returned to the form
- The device and its typed property records are saved in a single database transaction

Device example: V8106

← All devices

V8106

Unknown

Edit mode

Hardware ID: 001-ps · Component ID: 1065.831.200.001

Device properties 15

Group settings

Find a property or type

Property	Value type	Value	Units
3.3V(A)	Integer	110	–
5V(A)	Integer	220	–
I/N	String	Unknown	–
Ordering Code	String	WW8106AAAAAA	–
S/N	String	Unknown	–
control_system	String	Microprocessor Controlled	–
crate_compability	String	VME64/64X, 21 Slot, 8U	–
equipment_category	String	Power Supply	–
equipment_type	String	VME	–
manufacturer	String	CAEN	–
output_noise_level	String	Extremely Low	–
power_module_replaceable	Boolean	Yes	–
power_supply_technology	String	Switching	–
product_series	String	V81XX	–
±12V(A)	Integer	20	–

Created
12.11.2026 16:39

Updated
12.11.2026 16:39

Step 3: find and maintain equipment

- Browse the group tree or search by device name, Hardware ID, Component ID or group
- Filter by group including its subgroups; results are paginated
- Open a device card to inspect identifiers, effective properties, values and their source group
- An editor switches to edit mode to update the device name, Component ID or property values
- Hardware ID remains the stable identifier in the standard editing workflow
- Interface form drafts remain in the current browser until saved or discarded

Thank You for Your Attention!

Questions? / Comments?