Hardware & Configuration DB Status report

Current status

Hardware Database

- A catalog of hardware components that SPD detector consist of.
- It should contain the information about the detectors and the electronic parts, racks, and crates, as well as the location history of all items (optionally)
- It include equipment models, provider, parameters and other (semi)permanent characteristics
- This should help in maintenance of the detector systems and especially helpful in knowledge transfer between team members.
-

Hardware Database

- A prototype system is being developed, including PostgreSQL as a back-end, accessed through the REST API from the web interface
- For each type of device, a set of parameters are defined that are common to all devices of this type,
  - Each parameter has type as well as optional ranges of allowed values
- There can be common values for all devices of the same type
- Each device has unique ID assigned to it, and specific values of the parameters can be specified for it, based on its type.
- The set of parameters is now being specified as JSON
- Alternative solution with sets of dynamically generated tables is being developed

Hardware Database Schema

Tables schema

HWID	type	S/N	Notes	state	parameters
0007-015b2e3488ac	04fd15c3	PG1342		OK	{JSON}
0007-015b2e3488ad	04fd15c3	PG1344		FUBAR	{JSON}
0007-01fe47adf301	0368eba1	164756		FU	{JSON}
0007-01fe47adf301	0368eba1	164756		OK	{JSON}

TypeID	Name	Description	parent	parameters
04fd15c3	PG2T390	PANGOMICRO Titan-2 PG2T390 FPGA Board	000013f0	{JSON}
0368eba1	EQR151110	EQR15 11-1010D-S SiPM	00001134	{JSON}
000013f0	Concentrator			
00001134	SiPM			

```
{
  …
  "ov": "37",
  "dcr": "253",
  …
}
```

```
{
  "rov": {
    "description":"rec. oper. V",
    "value":"38"
  },
  "ov": {
    "description":"oper. V",
    "type":"dec_1",
    "lo":"36",
    "hi":"40"
  }
  …
}
```

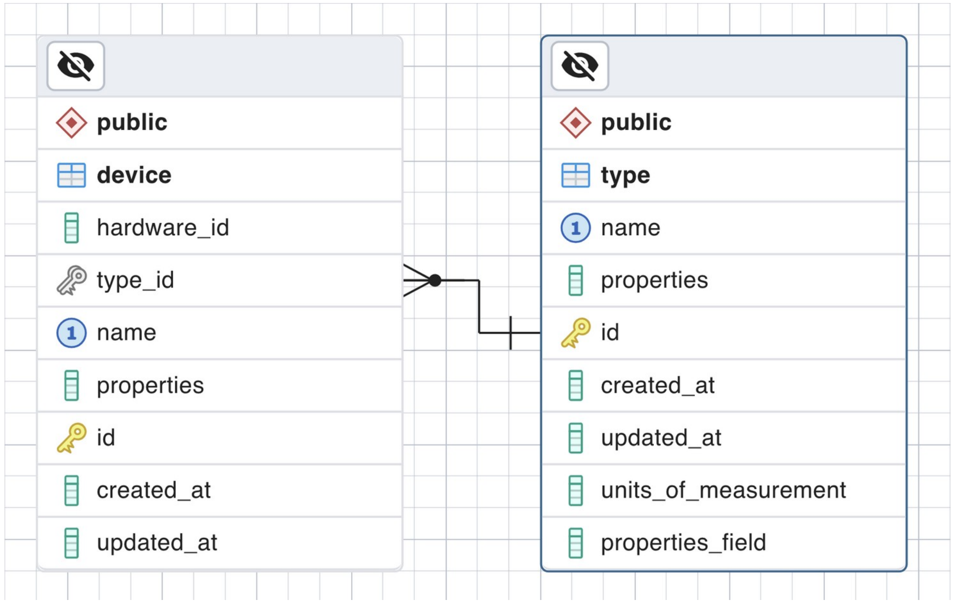
Hardware Database

Types table

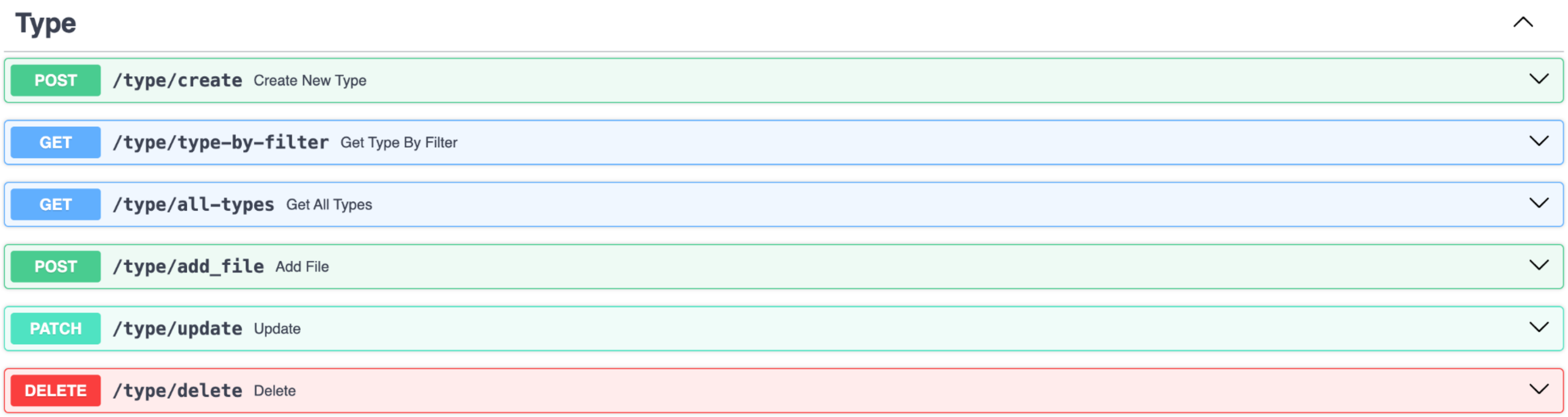
- id – unique type ID
- name – type name
- properties – name and type of the parameters
- properties_fiels – min, max and default values

Devices table

- hardware_id – unique device ID
- type_id – ID of the type of the device
- name – device name
- properties – name and value of the parameters



Types Endpoints



Type

POST /type/create Create New Type

GET /type/type-by-filter Get Type By Filter

GET /type/all-types Get All Types

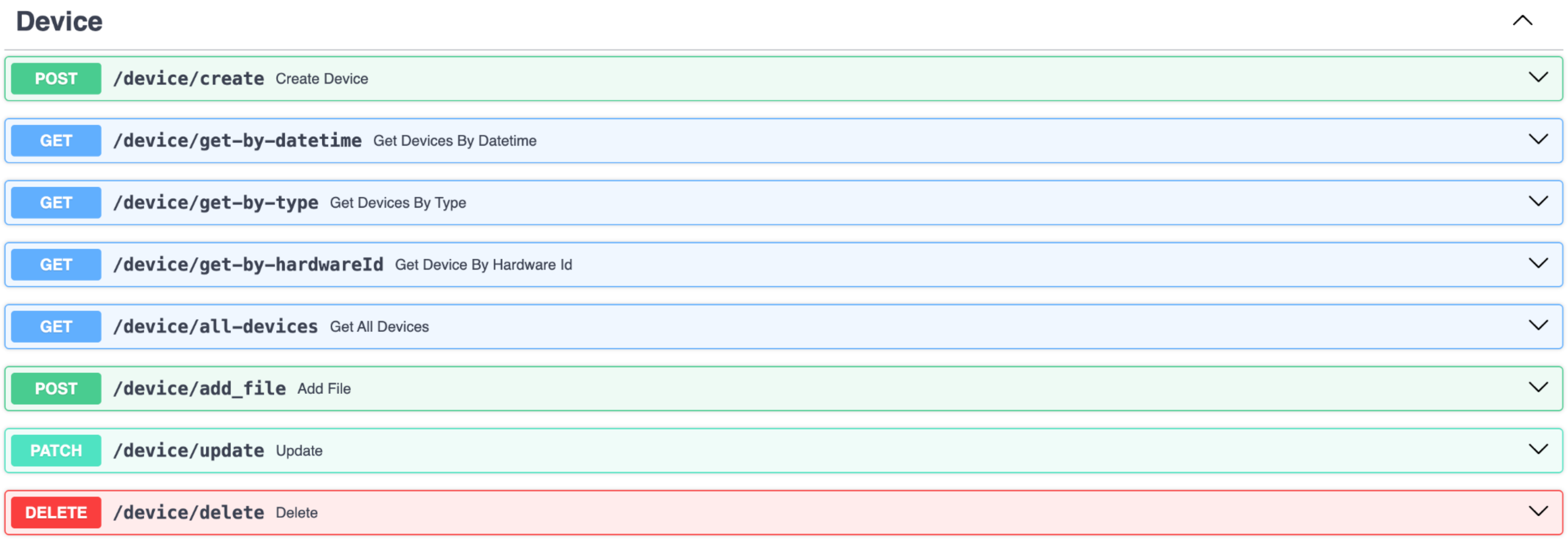
POST /type/add_file Add File

PATCH /type/update Update

DELETE /type/delete Delete

- /create - creating a type
- /type-by-filter - getting a type by name or id
- /all-types - getting all types
- /add_file - creating multiple types
- /update - updating a type
- /delete - deleting a type

Devices Endpoints


Device

POST /device/create Create Device

GET /device/get-by-datetime Get Devices By Datetime

GET /device/get-by-type Get Devices By Type

GET /device/get-by-hardwareId Get Device By Hardware Id

GET /device/all-devices Get All Devices

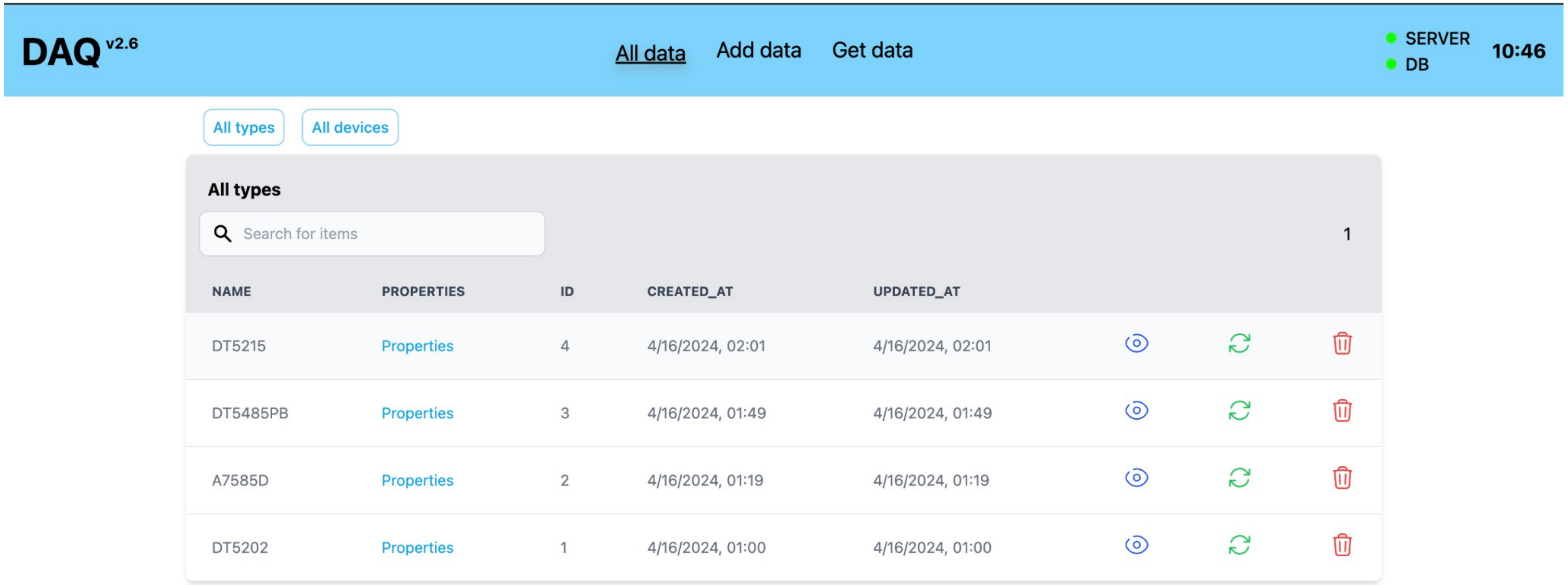
POST /device/add_file Add File

PATCH /device/update Update

DELETE /device/delete Delete

- /create - creating a device
- /get-by-datetime - getting a device by creation time
- /get-by-type - getting a device by type
- /get-by-hardwareId - getting a device by hardware ID
- /all-devices - getting all devices
- /add_file - creating multiple devices
- /update - updating devices
- /delete - deleting devices

Web Interface


DAQ v2.6

All data Add data Get data

● SERVER
● DB
10:46

All types All devices

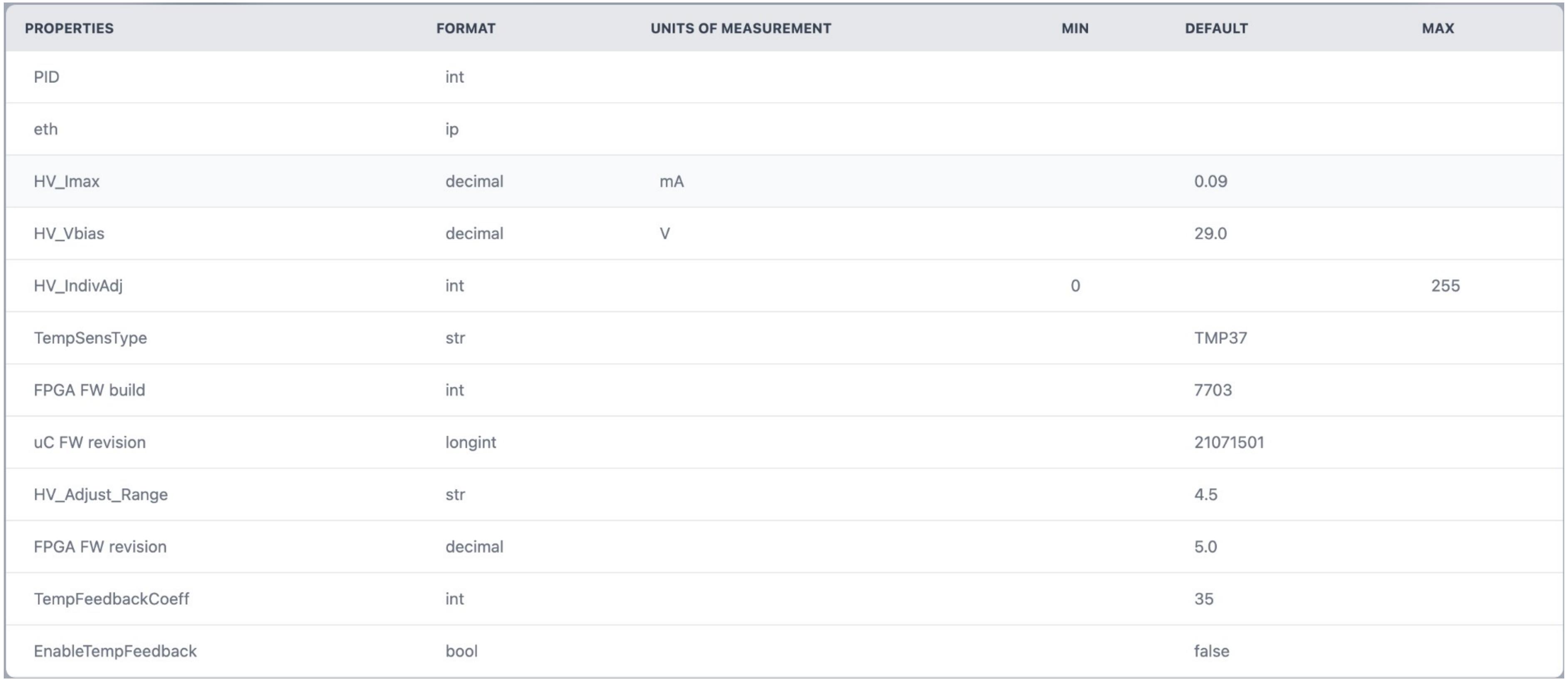
All types

Search for items

1

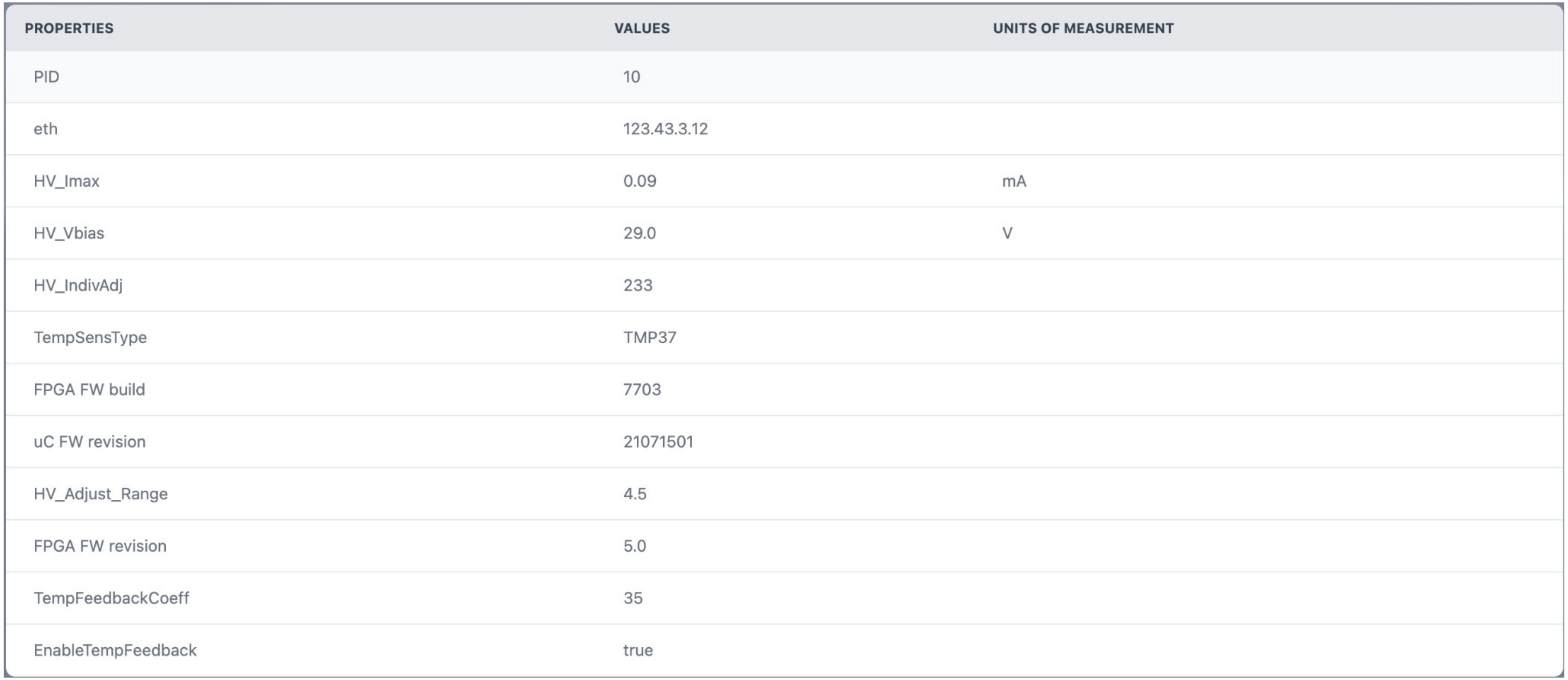
NAME	PROPERTIES	ID	CREATED_AT	UPDATED_AT			
DT5215	Properties	4	4/16/2024, 02:01	4/16/2024, 02:01			
DT5485PB	Properties	3	4/16/2024, 01:49	4/16/2024, 01:49			
A7585D	Properties	2	4/16/2024, 01:19	4/16/2024, 01:19			
DT5202	Properties	1	4/16/2024, 01:00	4/16/2024, 01:00			

Web Interface



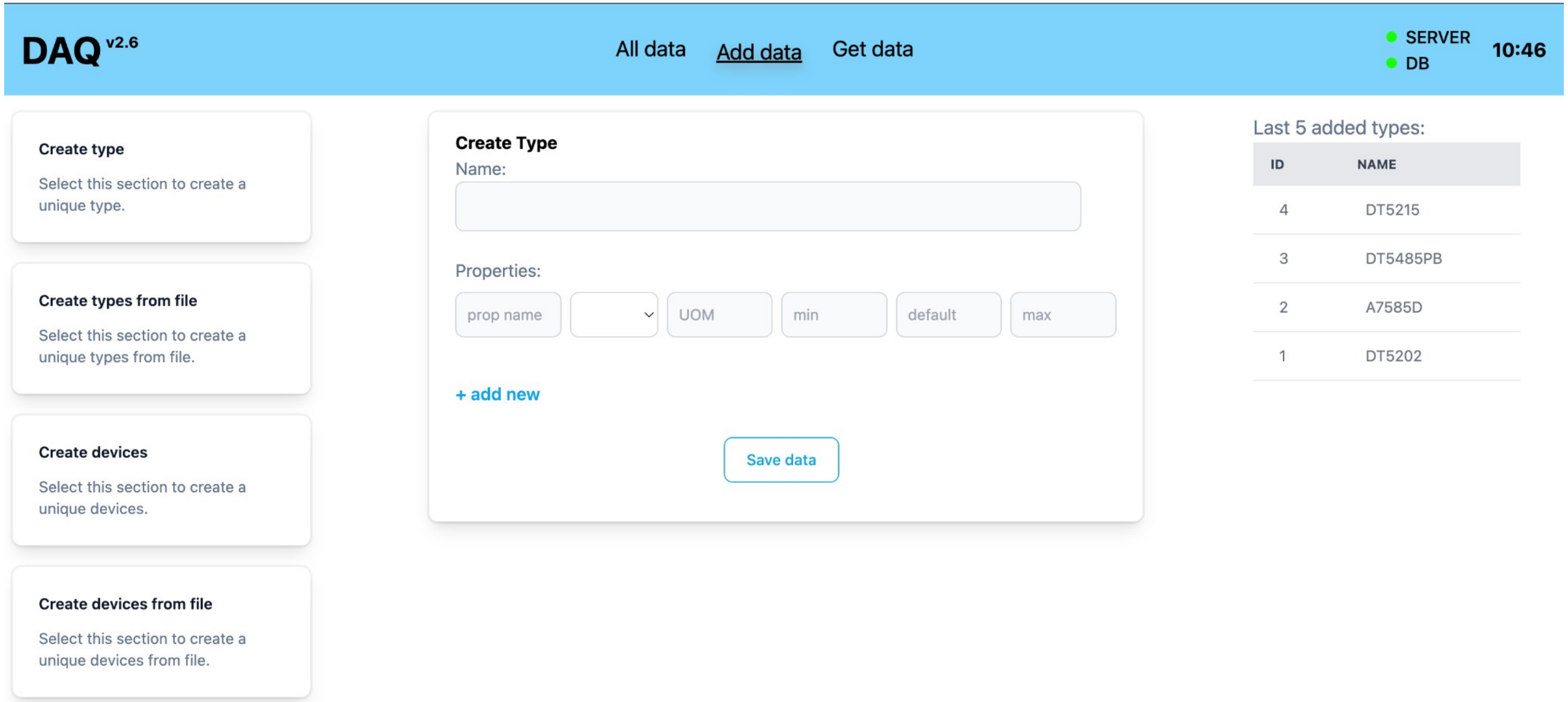
PROPERTIES	FORMAT	UNITS OF MEASUREMENT	MIN	DEFAULT	MAX
PID	int				
eth	ip				
HV_Imax	decimal	mA		0.09	
HV_Vbias	decimal	V		29.0	
HV_IndivAdj	int		0		255
TempSensType	str			TMP37	
FPGA FW build	int			7703	
uC FW revision	longint			21071501	
HV_Adjust_Range	str			4.5	
FPGA FW revision	decimal			5.0	
TempFeedbackCoeff	int			35	
EnableTempFeedback	bool			false	

Web Interface



PROPERTIES	VALUES	UNITS OF MEASUREMENT
PID	10	
eth	123.43.3.12	
HV_Imax	0.09	mA
HV_Vbias	29.0	V
HV_IndivAdj	233	
TempSensType	TMP37	
FPGA FW build	7703	
uC FW revision	21071501	
HV_Adjust_Range	4.5	
FPGA FW revision	5.0	
TempFeedbackCoeff	35	
EnableTempFeedback	true	

Web Interface



Create type

Select this section to create a unique type.

Create types from file

Select this section to create a unique types from file.

Create devices

Select this section to create a unique devices.

Create devices from file

Select this section to create a unique devices from file.

Create Type

Name:

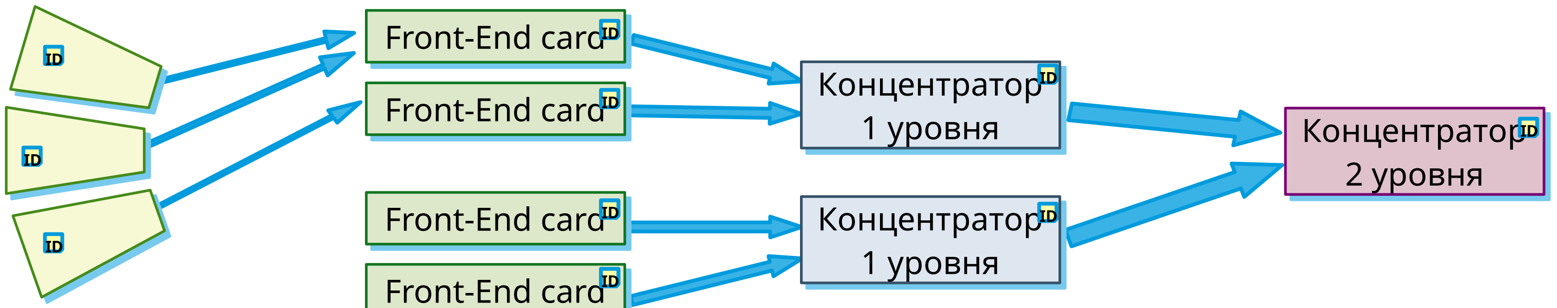
Properties:

Last 5 added types:

ID	NAME
4	DT5215
3	DT5485PB
2	A7585D
1	DT5202

DAQ Mapping

- The number of data collection channels of the SPD installation will be several hundred thousand
- The signals from the detector will pass through several communication devices

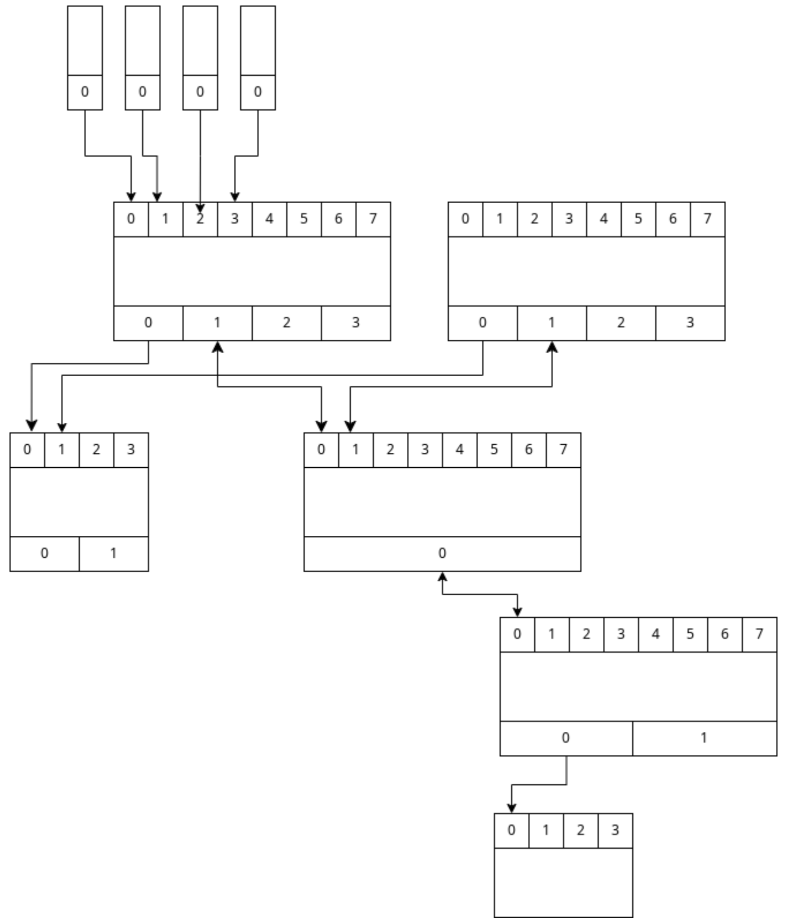


- It is necessary to have a mapping of the data collection system that establishes the correspondence of the channel addresses at the DAQ outputs with the devices from which this signal came

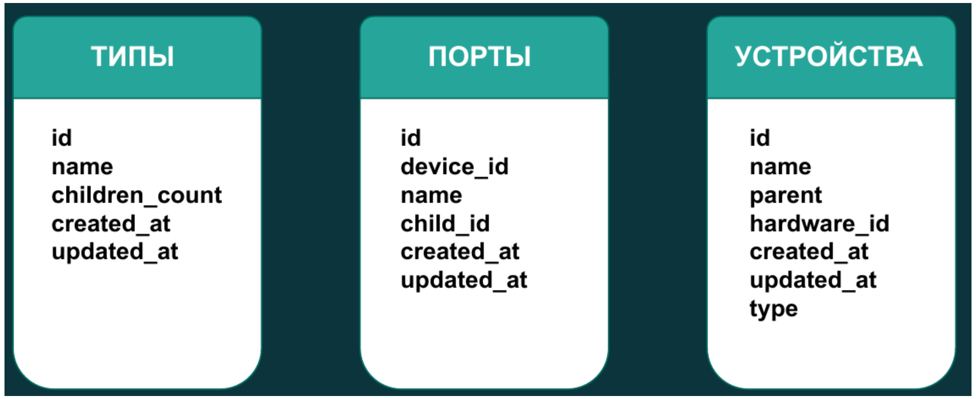
Building of mapping

- Due to the large number of elements in the system, it is almost impossible to build mapping manually
- For the elements involved in the transmission of digital signals, an automatic mapping procedure should be implemented
  - The element must issue a HW ID over the data channel in response to a special signal
- For parts of the system that are not equipped with automatic source ID recognition, an interface must be provided that allows data entry by groups.

Mapping model



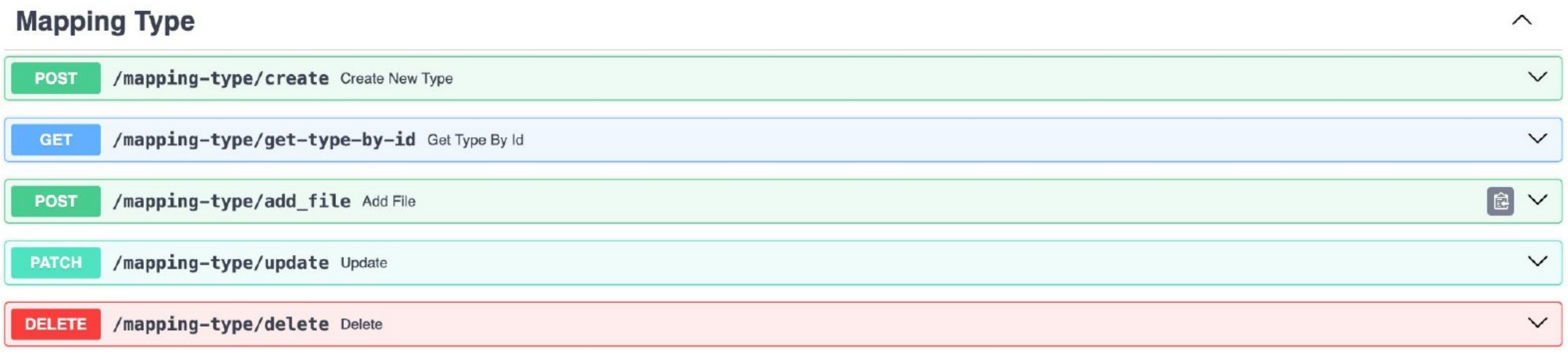
Tables



ТИПЫ	ПОРТЫ	УСТРОЙСТВА
id	id	id
name	device_id	name
children_count	name	parent
created_at	child_id	hardware_id
updated_at	created_at	created_at
	updated_at	updated_at
		type

Type table: (id) unique type identifier, (name) type name, (children_count) number of child devices, (created_at) type creation time,(update_at) type update time.
Port table: (id) unique identifier,(device_id) id of the device to which the port belongs, (name) port name, (child_id) id of the device connected to this port, (crated_at) port creation time, (updated_at) port update time.
Device table: (id) unique identifier,(name) device name, (parent) device parent, (hardware_id) unique device identifier, (created_at) device creation time, (updated_at) device update time,(type) device type

Mapping types Endpoints



Mapping Type

POST /mapping-type/create Create New Type

GET /mapping-type/get-type-by-id Get Type By Id

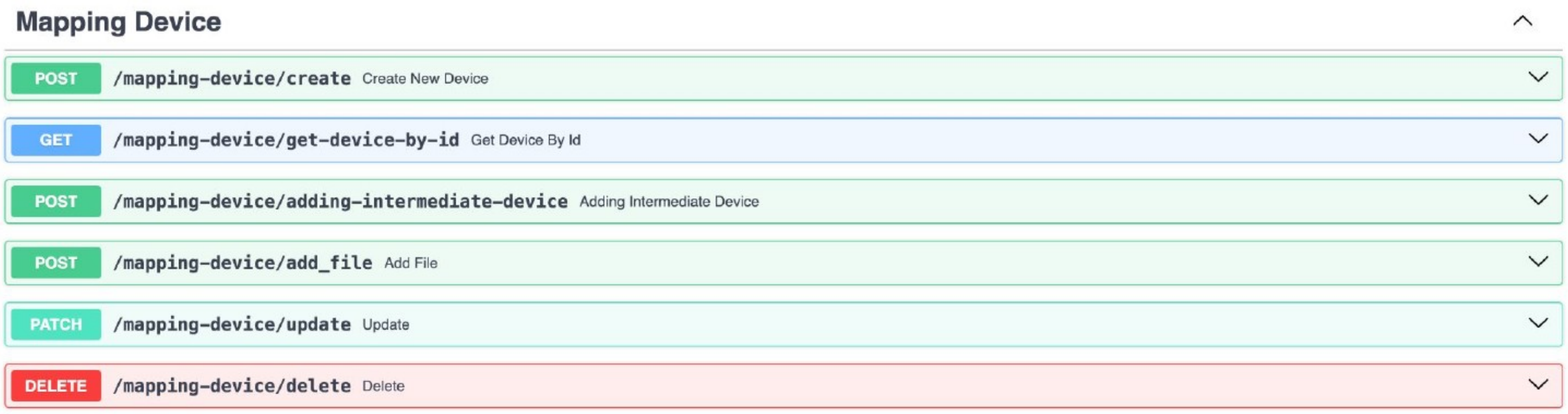
POST /mapping-type/add_file Add File

PATCH /mapping-type/update Update

DELETE /mapping-type/delete Delete

- /create - creating a type
- /get-type-id - getting a type by id
- /add_file - creating multiple types
- /update - updating a type
- /delete - deleting a type

Mapping devices Endpoints



Mapping Device

POST /mapping-device/create Create New Device

GET /mapping-device/get-device-by-id Get Device By Id

POST /mapping-device/adding-intermediate-device Adding Intermediate Device

POST /mapping-device/add_file Add File

PATCH /mapping-device/update Update

DELETE /mapping-device/delete Delete

- /create - creating a device
- /get-device-by-id - getting a device by id
- /adding-intermediate-device - adding intermediate devices
- /add-file - adding multiple devices
- /update - updating a device
- /delete - deleting a device

Building of mapping

- Due to the large number of elements in the system, it is almost impossible to build mapping manually
- For the elements involved in the transmission of digital signals, an automatic mapping procedure should be implemented
  - The element must issue a HW ID over the data channel in response to a special signal
- For parts of the system that are not equipped with automatic source ID recognition, an interface must be provided that allows data entry by groups.

Performance requirements

- It is expected that the filling of the hardware database will take place gradually, and updates will be rare
- The construction of the connection diagram and its changes will also be performed rarely (no more than once a week)
- The requirements for the speed of recording information in the database are low
- Mapping information may be required when processing each file. It is possible that tens of thousands of processes will try to simultaneously access the system.
- It is necessary to ensure their processing, avoiding database overload due to too high frequency of requests

Input required

- A Input for the further development of the HWDB is required
- Most of the subsystems are in the very early stage of the development
- For some detectors, like Range System, design of the system and components are already defined to some extent
- The development of the HWDB can be based on the input from them