IC@MS — Integrated Critical Alarms Management Software

40th Tango Community meeting at ALBA

Łukasz Żytniak (S2I),
Thomas Juerges (SKAO).

The GSI - Gesellschaft für Schwerionenforschung (Society for Heavy Ion Research)

GSI was founded in **1969**.

- employs around **1520 Staff,**
- ~1200 international user community.

GSI's accelerator complex includes:

1. **UNILAC linear accelerator**,
2. **SIS heavy-ion synchrotron**,
3. **ESR experimental storage ring**,
4. **FRS fragment separator**,
5. **PHELIX high-energy/high-performance laser**,
6. Large spectrometer and detector systems, a medical irradiation facility for cancer-therapy research.

www.zeit.de/news/2026-02/18/gsi-nach-feuer-reparaturen-sollen-forschung-rasch-sichern

5 February 2026, the GSI Helmholtzzentrum für Schwerionenforschung in Darmstadt

0

injuries reported

~06:30

fire brigade alerted

no

hazardous substances detected

~06:00

Fire starts

High-voltage supply area for UNILAC; The fire was caused by a probable technical defect inside a control cabinet.

~06:30

Alarm raised

Darmstadt fire brigade alerted via automatic fire alarm system and 112.

Morning

Response scales

Large multi-agency operation; Campus evacuation and crisis chains activated.

Aftermath

No injuries

No danger to the neighborhood; No hazardous substances detected during the operation.

www.fr.de/rhein-main/darmstadt/brand-im-forschungszentrum-in-darmstadt-hat-enorme-auswirkungen-94157055.html

Even with fast response, the cost of downtime was severe

Alarm systems do not eliminate incidents;
They reduce escalation, casualties, and secondary losses.

Reported impacts

UNILAC building severely affected

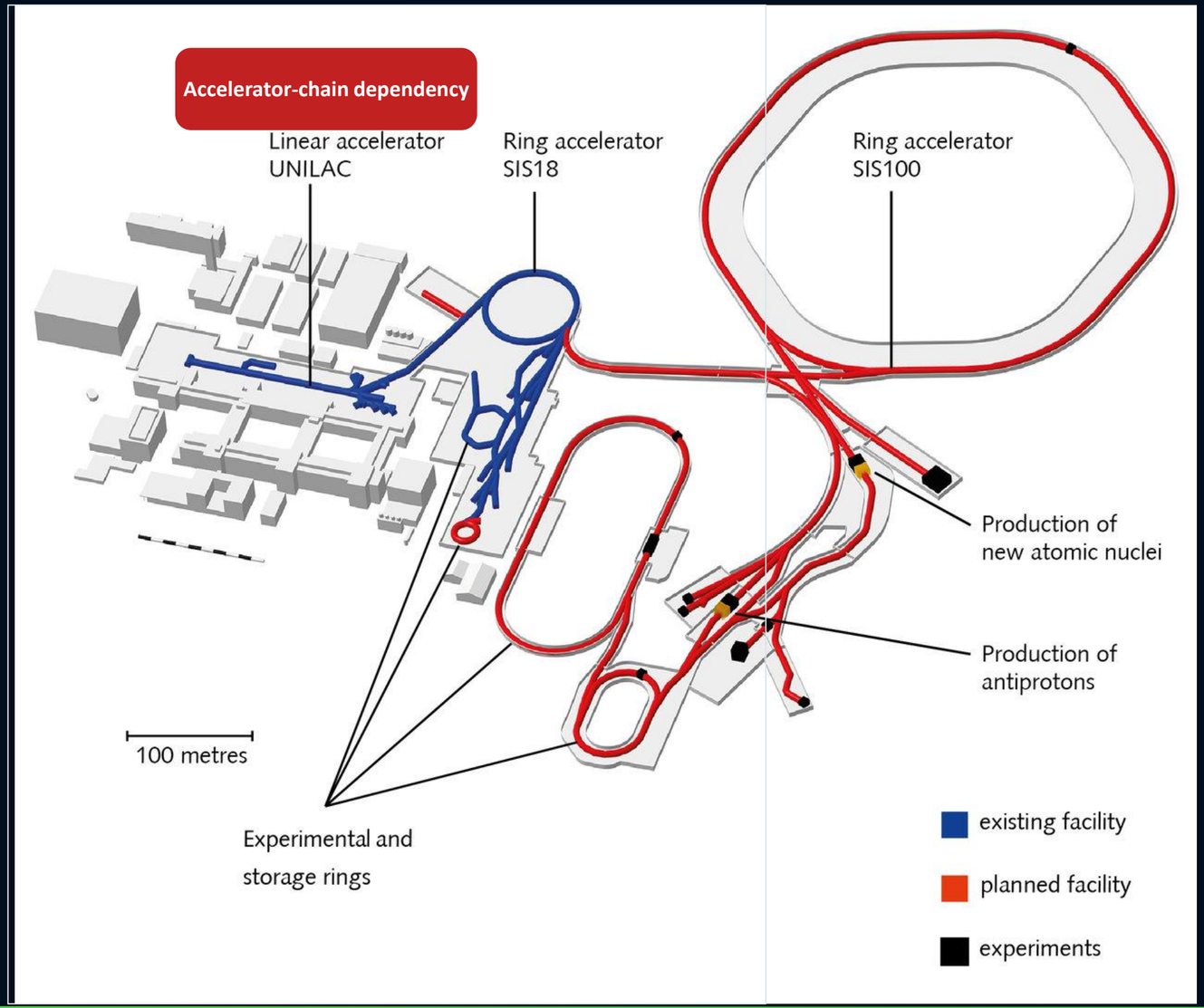
The operation of the accelerator facility was not possible after the incident.

Beamtime disruption

2026/27 beamtime cannot run as planned;
no 2026 beamtime expected at UNILAC-dependent facilities.

Long restoration path

Full restoration of damaged infrastructure can take years; alternative technical solutions are being developed.

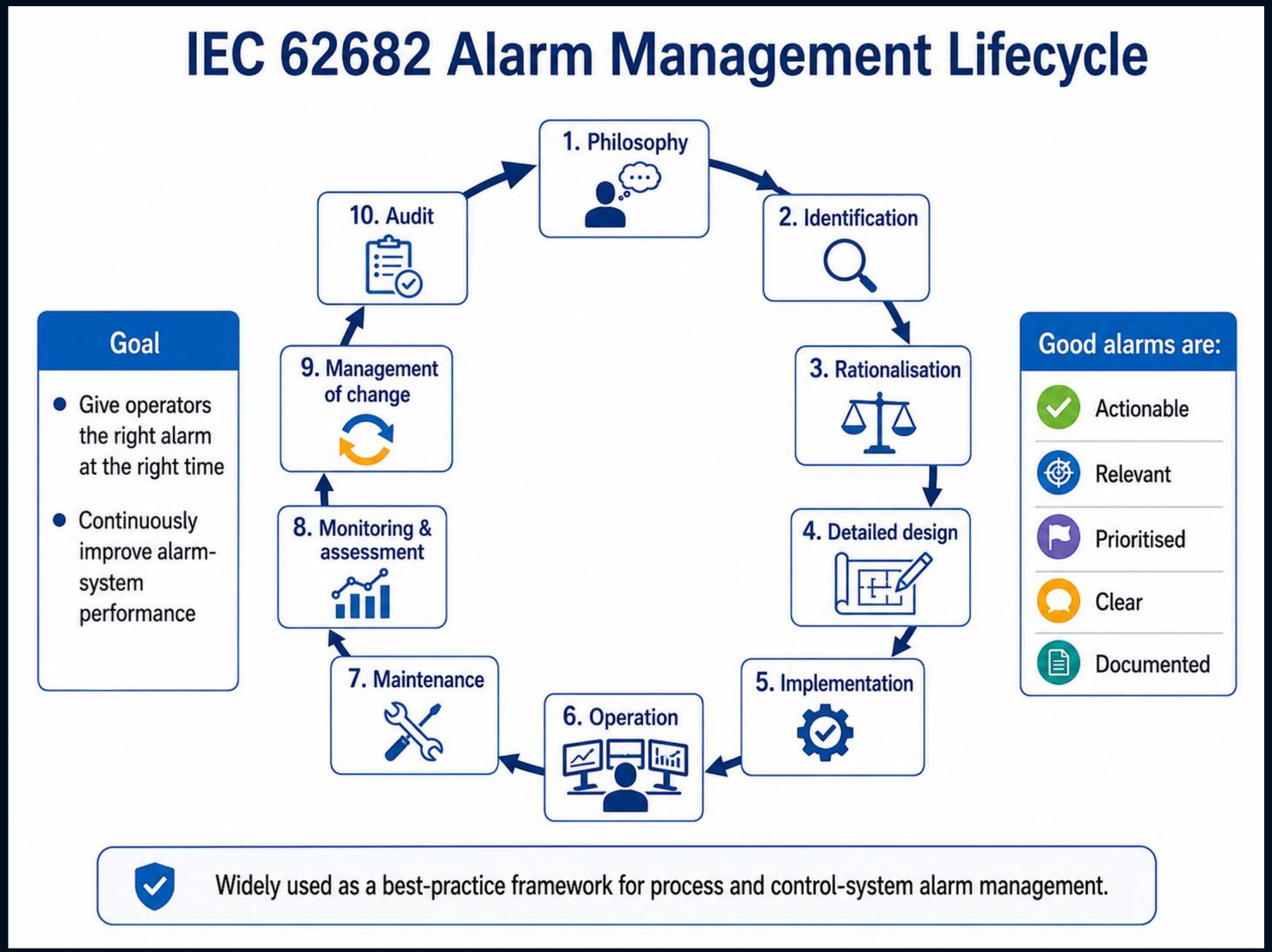


Alarms System ≠ Alarms Software

IEC 62682

How do we ensure that alarms shown to operators are:

1. meaningful,
2. actionable,
3. prioritised,
4. documented,
5. maintainable,
6. not overwhelming?



Integrated Critical Alarms Management Software

icams

How does it work?

Temperature sensor

If temperature is higher than 50 -> trigger alarm.

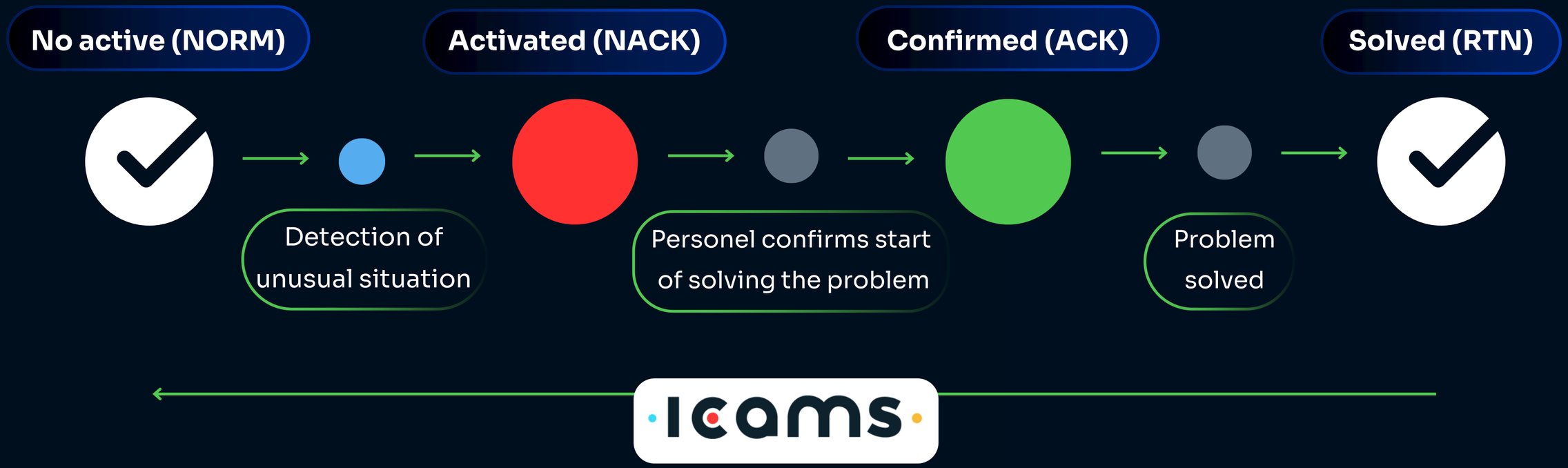
- Web API
- MongoDB CACHE
- Advanced logging
- Speed-up PANIC

- Operated at IC@MS
- Processed by Alarm-Handler/PyAlarm/ Achtung
- notifications
- logging

What IC@MS does?

Area	Functions identified
Alarm monitoring	Active alarms, inactive alarms, alarm details, filters, alarm state, severity, device and tag search
Alarm lifecycle	Add, modify, remove, acknowledge, reset, enable, disable
History and archiving	Alarm history, history by tag, snapshot/archive access
Operator workflows	Comments on actions, saved filters, table column configuration, recent searches
Notification/contact management	Panic phonebook, alarm mail, alarm notification configuration
Datasource/composer layer	HTTP, MQTT and Modbus datasource handling; composer creation and attribute mapping
Administration/security	Login, users, roles, groups, admin panel, password reset, mail configuration
Deployment	Docker Compose, Kubernetes, Helm charts, MongoDB, TangoDB, SNAP/PyTangoArchiving containers

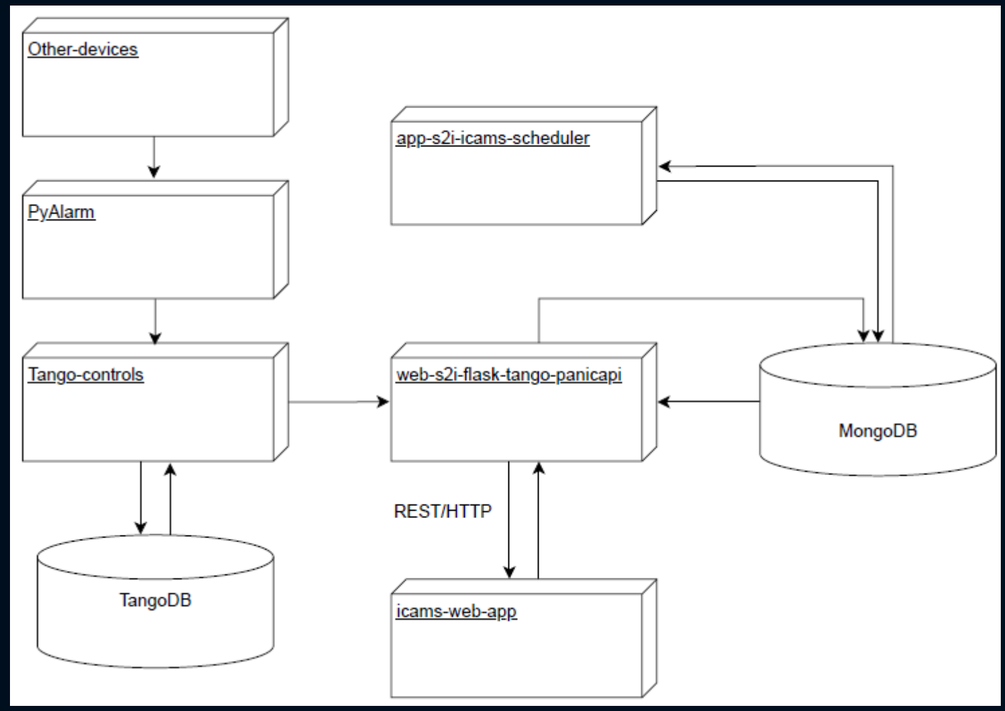
IC@MS is focused on daily operation



IC@MS is **Integrated Critical Alarms Management Software**

A web-based alarm-management and integration layer for Tango-based facilities, **NOT** as a replacement for **PANIC/PyAlarm** or **Elettra AlarmHandler**.

- **Frontend**: React/Redux web application for operators and engineers.
- **Backend API**: Flask/Python REST API exposing alarm, history, phonebook, datasource, composer, mail and user-management operations.
- **Scheduler/integration service**: Python service that synchronizes alarm-provider state and archived snapshots into MongoDB.



Why IC@MS?

Better user interface for everyday use

Access from anywhere through web browser

No need to install on the client’s computer

Web API

Multiple access roles

Integration with Active Directory / MAX-IV login.

Why IC@MS?

Advantage	Why it matters
Brownfield integration	IC@MS can sit above PyAlarm/PANIC, Achtung, and AlarmHandler instead of forcing facilities to replace proven alarm engines. PANIC is already a mature Tango alarm suite with configuration, triggering, acknowledgement, distributed PyAlarm engines and notification actions.
Web-native operations	Operators and engineers access alarms through a browser; no desktop Panic GUI installation is needed. The IC@MS is a web equivalent/extension of Panic GUI.
Open handover model	SKAO received source code, repository history, documentation and deployment artefacts under BSD-3-compatible licensing information, which allows independent extension.
Deployment readiness	Docker/Kubernetes deployment files, build instructions and deployment notes make IC@MS easier to adopt in modern facility infrastructure.
Operator workflow layer	Alarm list, acknowledgement, reset/disable actions, history, filtering, RBAC and APIs create an operational console.
Commercial + community model	SKAO can maintain or extend the code independently, while S2Innovation can still provide engineering support, customisation and long-term maintenance.

Why IC@MS?

IC@MS is the missing web-based operational layer between mature Tango alarm engines and modern facility operations: it keeps PyAlarm/PANIC, Achtung, and AlarmHandler useful, but makes them easier to deploy, operate, integrate and support at scale.

The speed

SKAO test results with the following loads:

- 100 dummy alarms --> nearly instant trigger of the test alarm,
- 300 dummy alarms --> nearly instant trigger of the test alarm,
- 500 dummy alarms --> 1-1.5s latency of the test alarm.




Thank You

For Your Attention

Lukasz Zytniak
(+48) 789 339 875
lukasz.zytniak@s2innovation.com

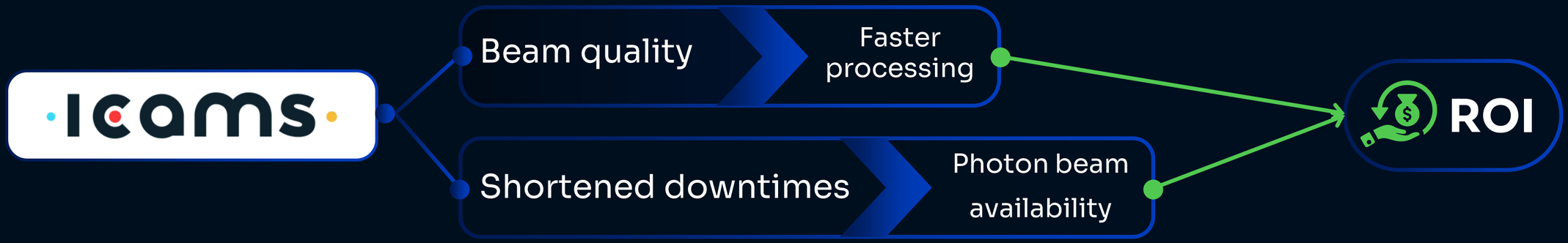
We recognise and acknowledge the Indigenous peoples and cultures that have traditionally lived on the lands on which our facilities are located.

Many thanks: Thomas Juerges, Matteo Canzari, Marco Bartolini, Sonja Vrcic, Ugur Yilmaz

SKAO

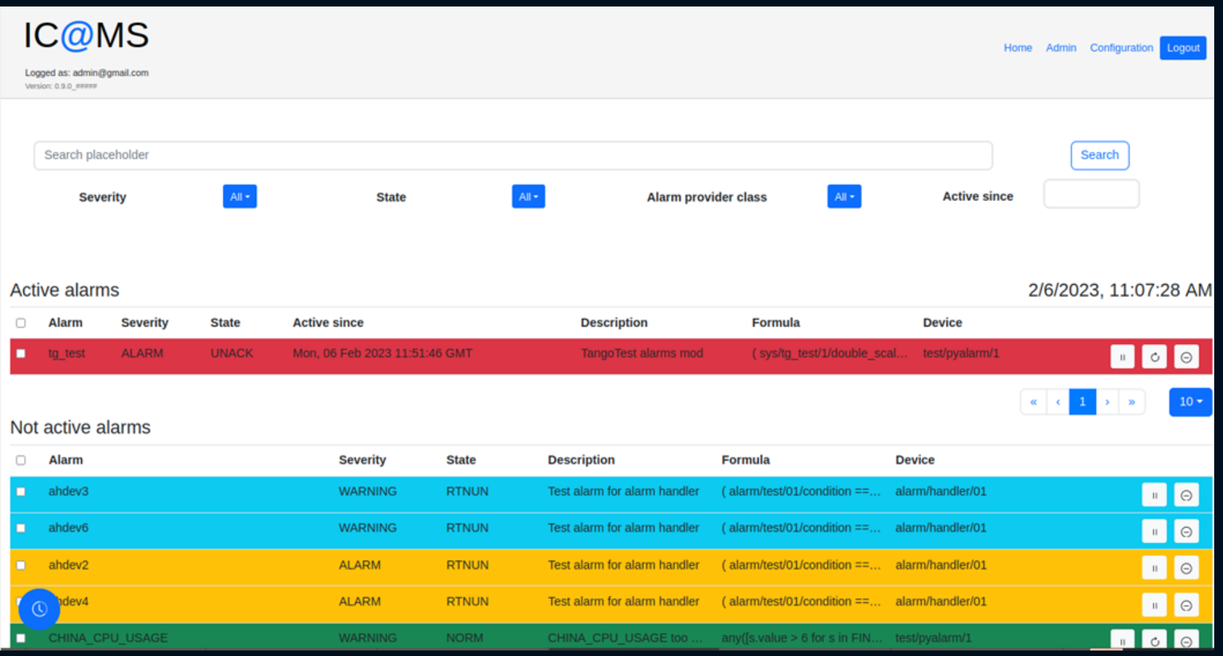
www.skao.int

Accelerators productivity increase



Logging page

- Support for logging using LDAP for different providers
- Forgot password supported
- Permissions by roles



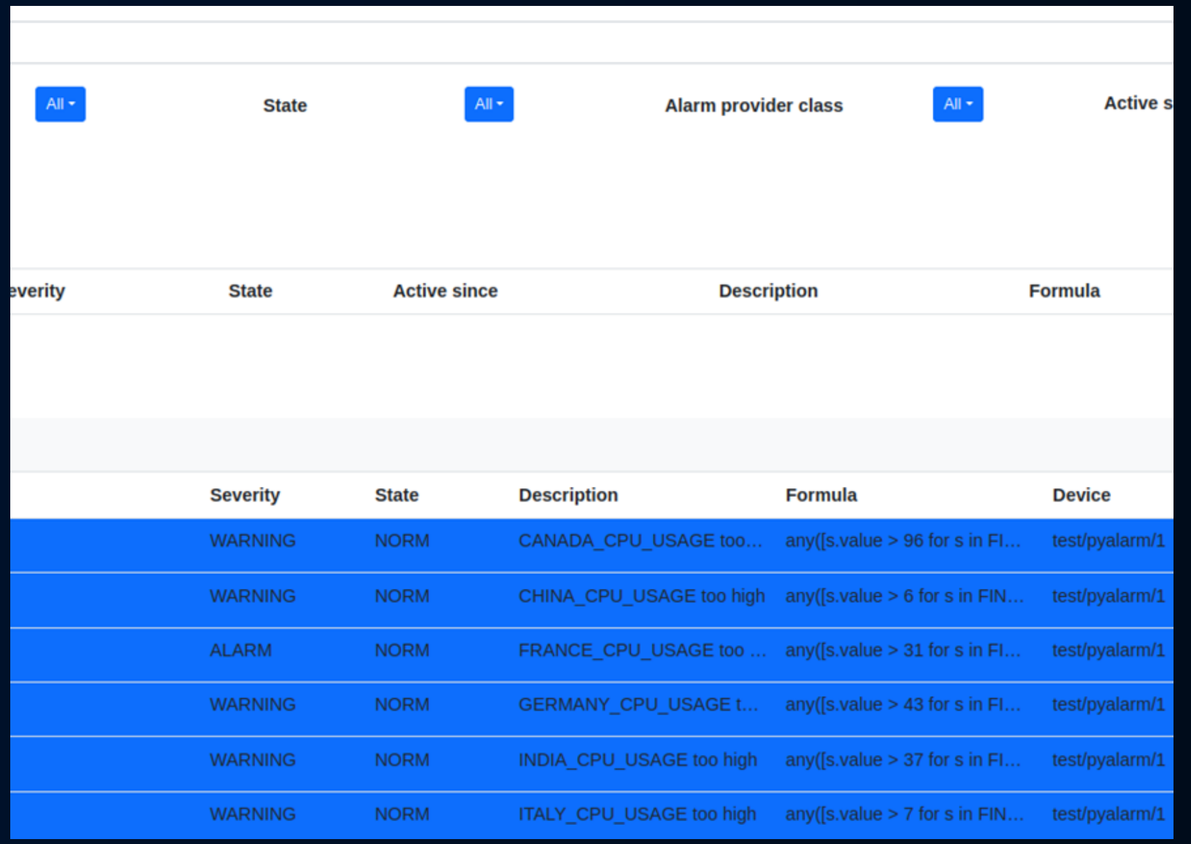
Home page

- Separation for active and not active alarms
- Colors indicating severities
- Search by severity, state, alarm provider class provided (PyAlarm, AlarmHandler)

s²:innovation
Software solutions

Home page

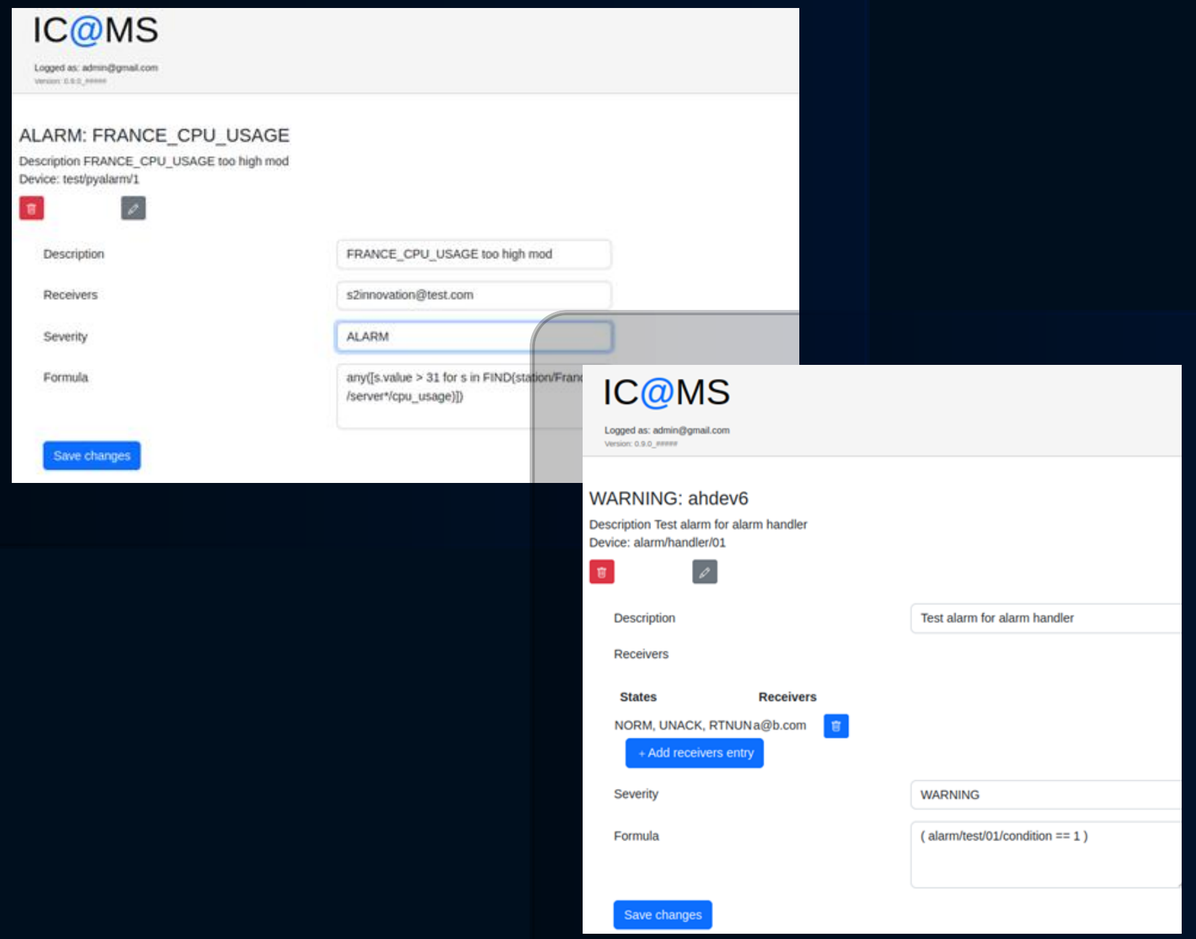
- Perform actions for alarms: acknowledge, reset, disable
- Possibility to choose multiple or all alarms and perform actions
- Pagination allows to choose how many alarms is visible



Alarm details page

- Possibility to modify existing alarm
- Show history of alarm from Tango Snap archiving with values used to calculate formula

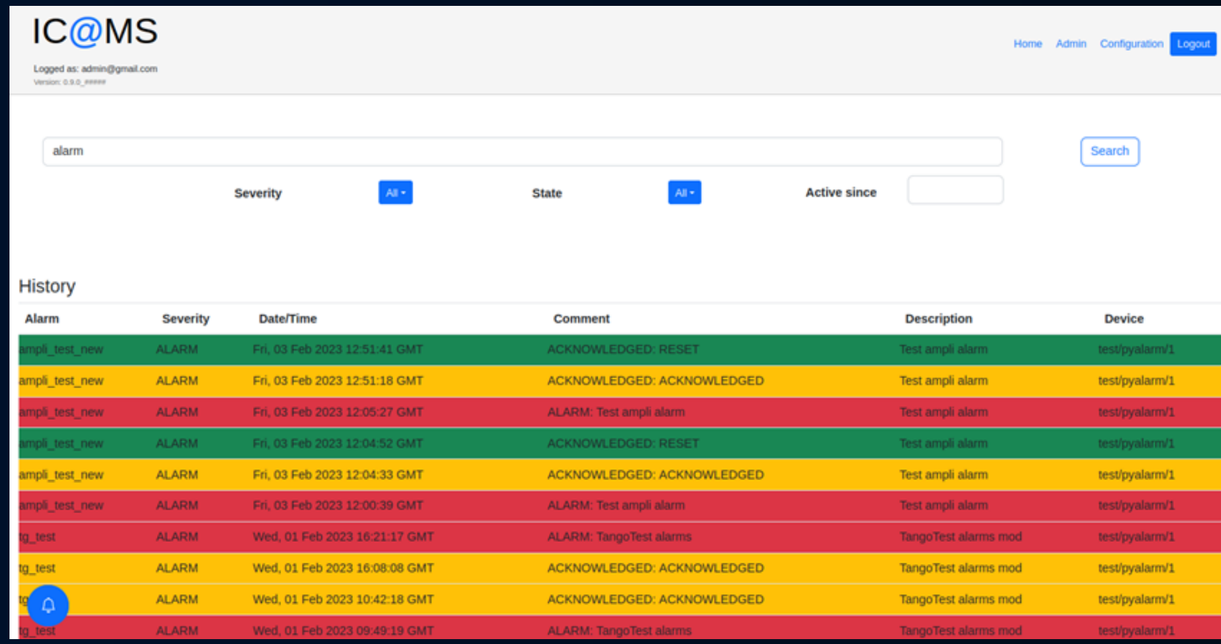
Alarm details page



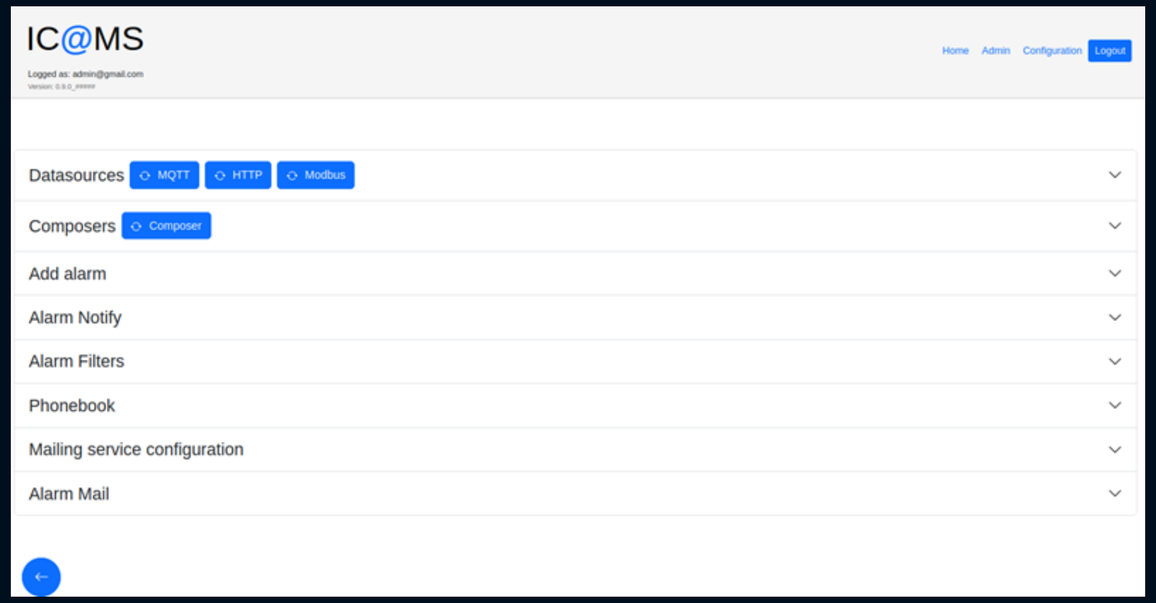
Possibility to modify existing alarm – supported for both PyAlarm and AlarmHandler alarms

Alarms history page

- See history for all alarms
- Search alarm history by name, comment, severity, date...



s²innovation
Software solutions



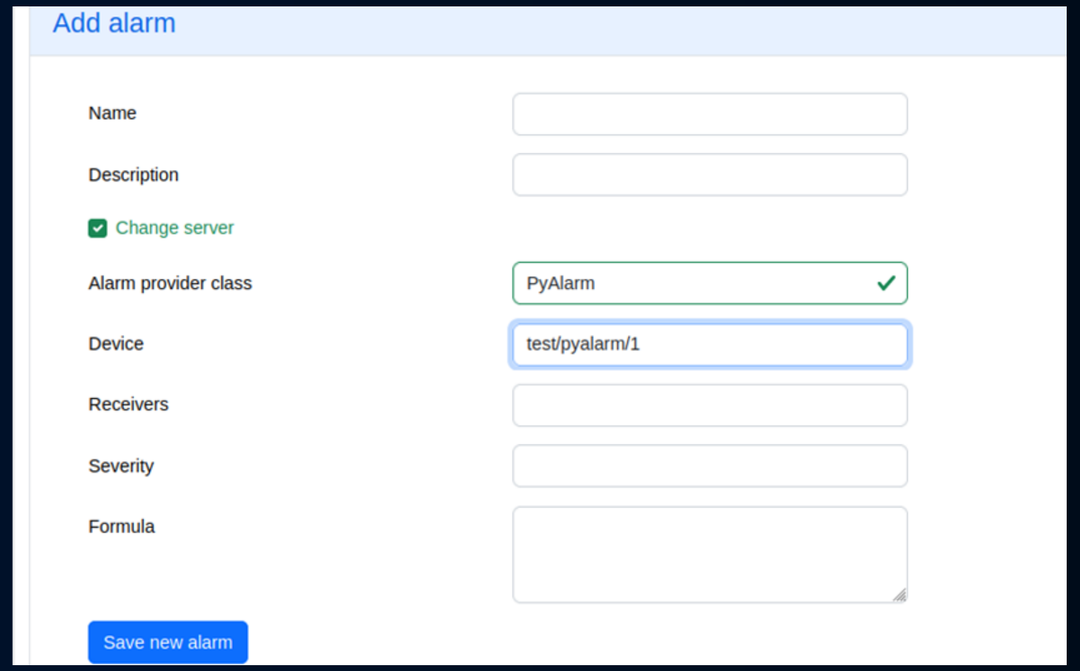
Configuration page

Allows to configure devices and alarms

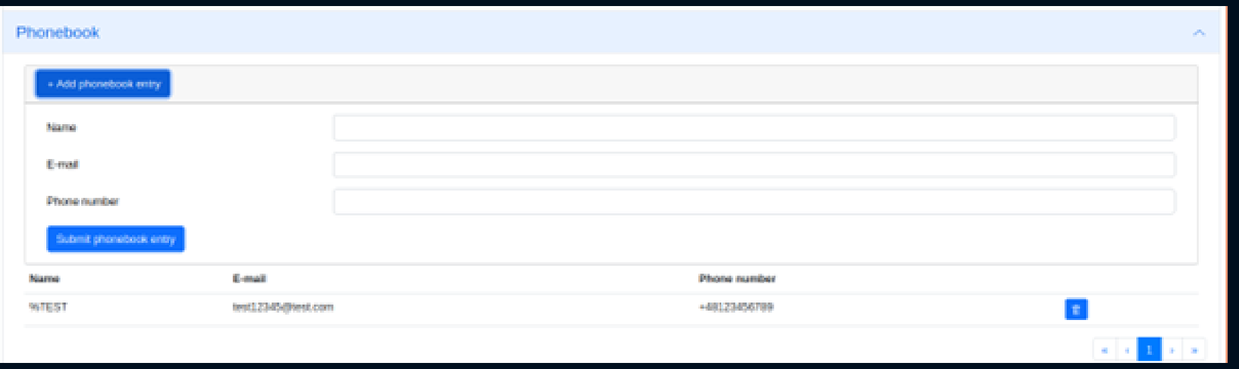
s²innovation
Software solutions

Add alarm form

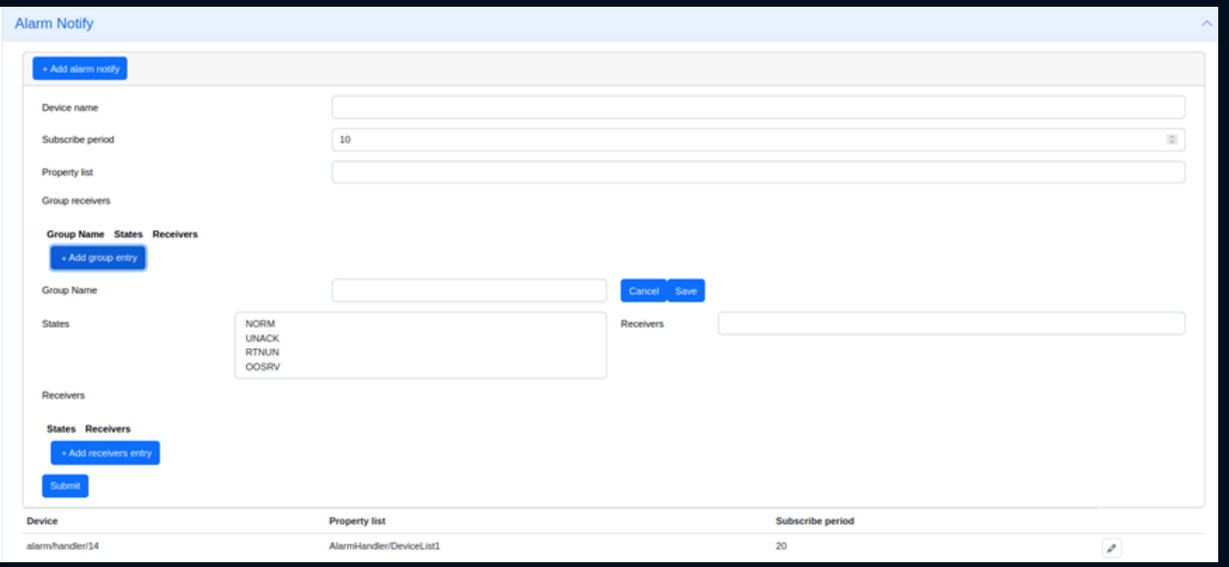
- Allows to create new alarm
- Possible to choose provider class (PyAlarm, AlarmHandler) and device for which alarm will be created



Phonebook



Allows to modify entries in PANIC phonebook
After add, you can use entries to choose receivers in PyAlarm alarms

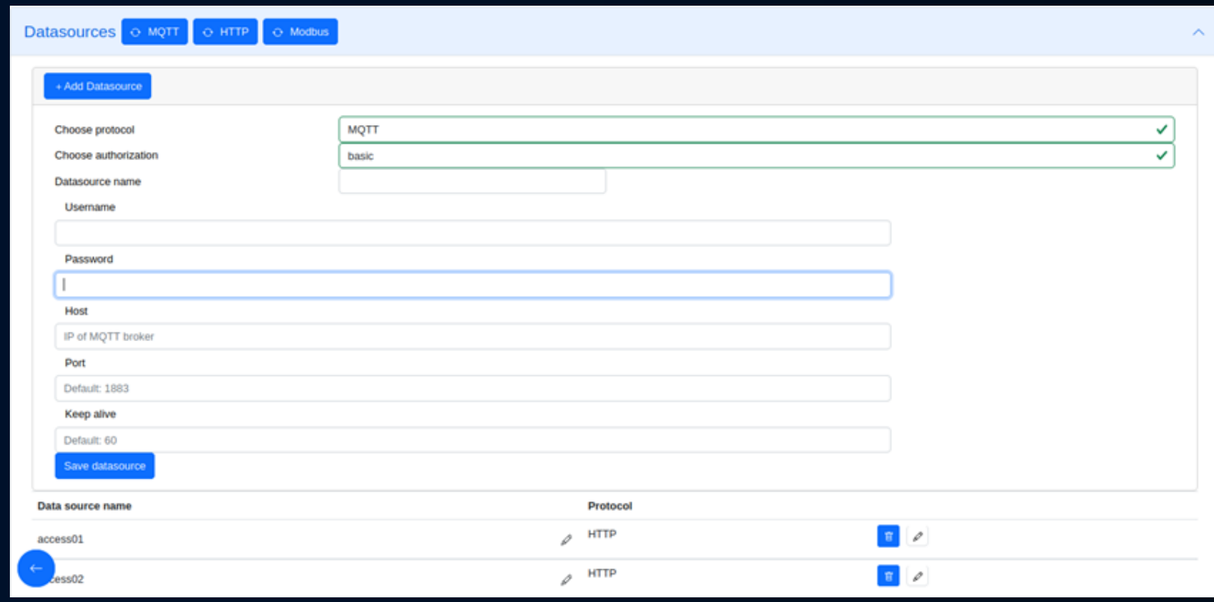


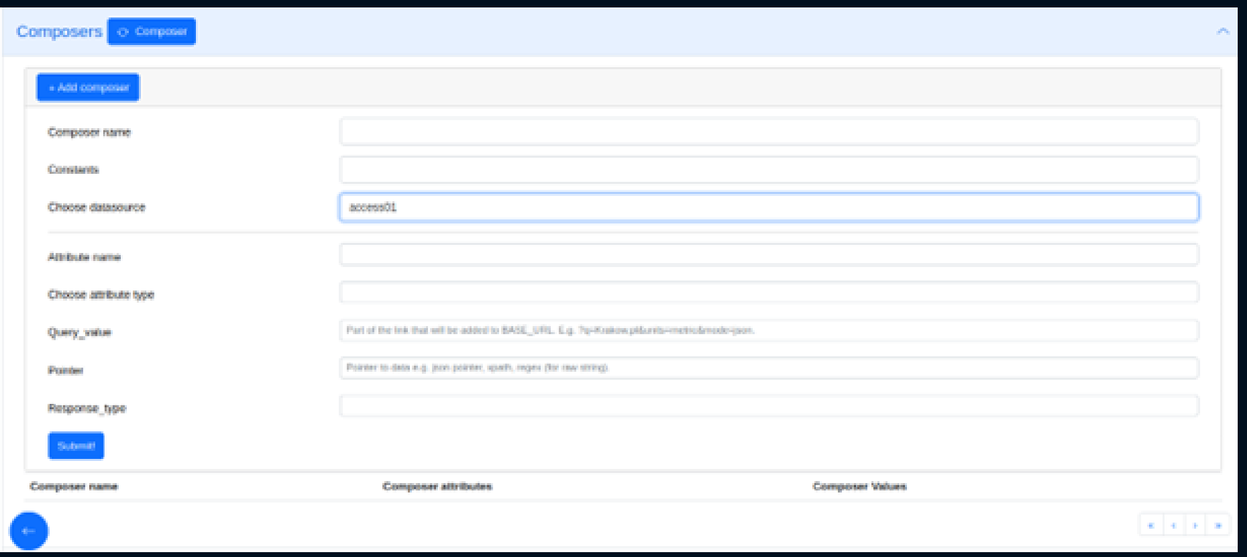
Alarm notify

- Allows to create and modify AlarmNotify devices which are notification devices for Elletra AlarmHandler
- Allows to define groups

Data sources

- Allows to create on-the-fly Tango devices to fetch data from HTTP server, MQTT or Modbus
- Different authorisation types supported (basic, cert, token)



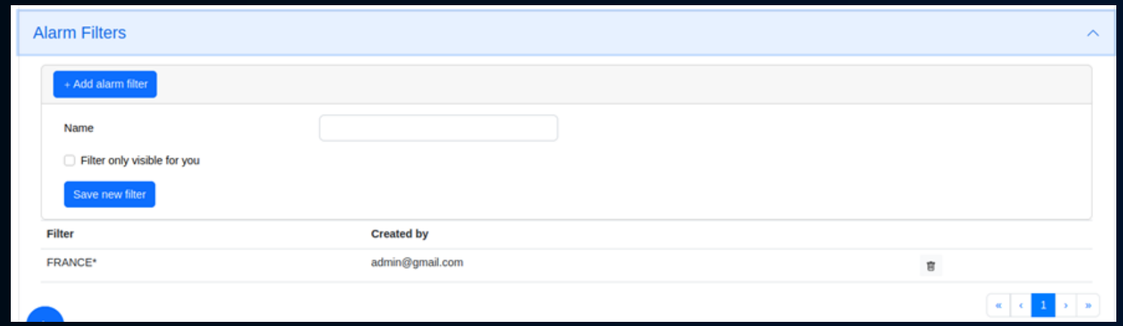


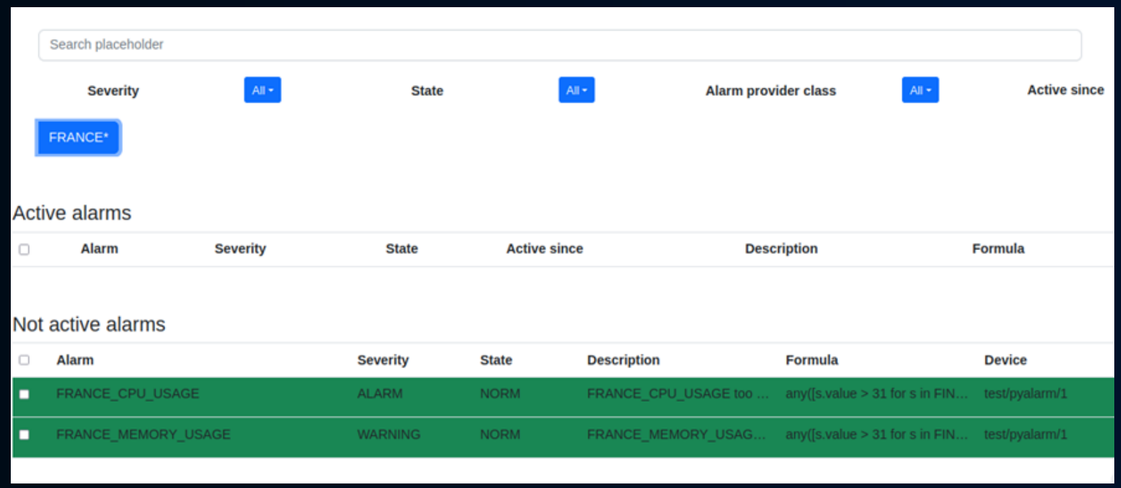
Composer devices

Allows to create on-the-fly Tango devices to compose data from different Tango data source devices (HTTP, MQTT and from Epics PV read from channel access)

Alarm Filters

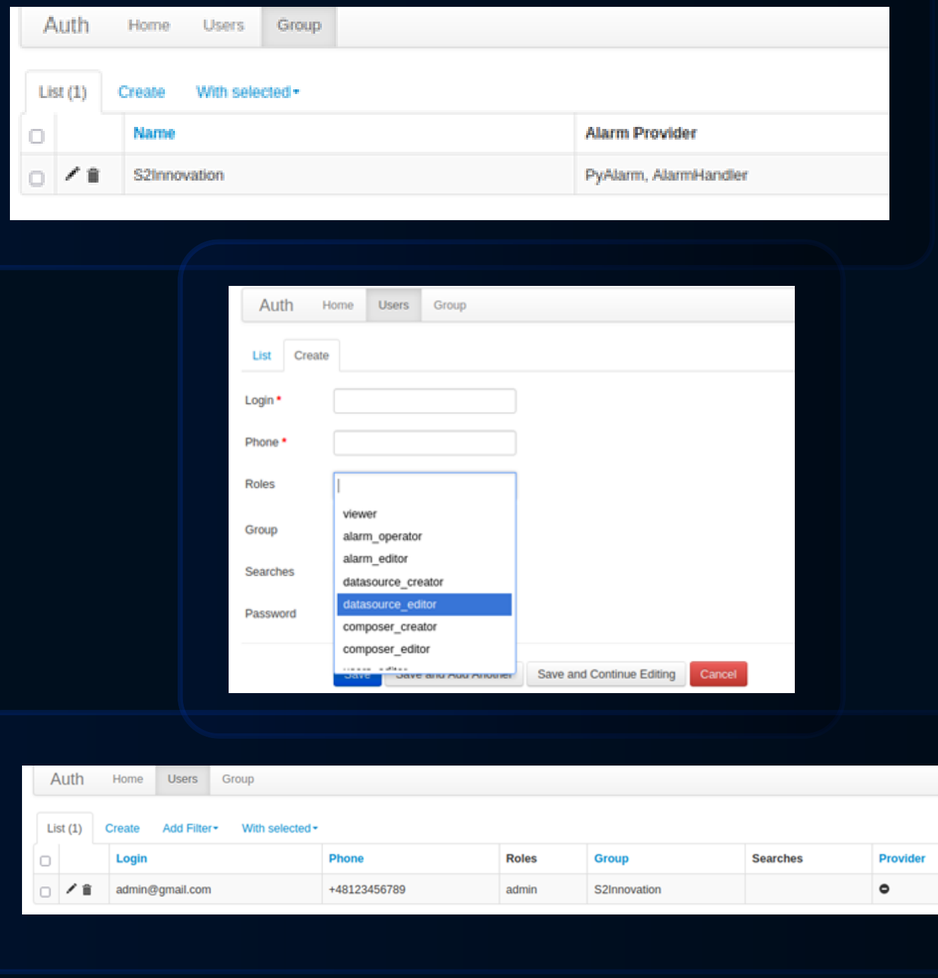
- Allows to create filters used to easily search alarms in history or home page
- Possible to choose visible level

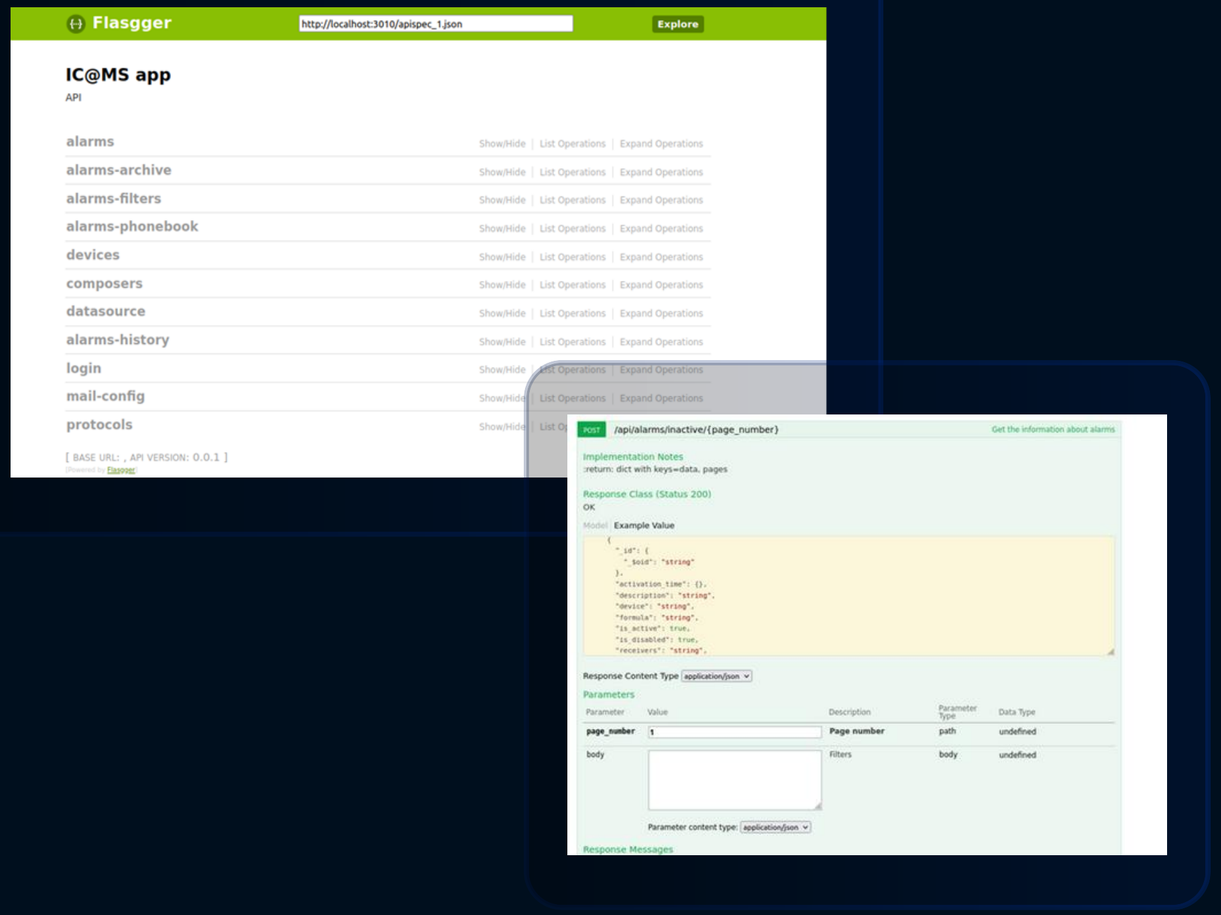




Admin panel

- Implemented using Flask admin
- Allows to manage users and groups (e.g. roles for users)
- Each group has provider classes



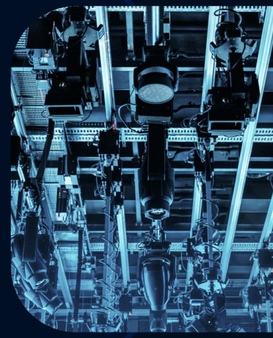


Swagger

- We are using Flask framework for backend, we use REST communication
- Swagger is tool to document and endpoints
- Endpoints are groupped and well defined to simplify development and testing

Thank You

For Your Attention



Lukasz Zytniak- COO
(+48) 789 339 875
lukasz.zytniak@s2innovation.com