

# ALARM HANDLER TO PHOEBUS

## Migrating the Alarm System at BESSY II

**ABSTRACT:** The BESSY II lightsource has been in operation at Helmholtz-Center Berlin (HZB) for 25 years and is expected to be operated for more than the next decade. The EPICS Alarm Handler (ALH) has served as the basis for a reliable alarm system for BESSY II as well as other facilities and laboratories operated by HZB. To preempt software obsolescence and enable a centralized architecture for other Alarm Handlers running throughout HZB, the alarm system is being migrated to the alarm-service developed within the Control System Studio/Phoebus ecosystem. To facilitate operation of the Alarm Handler, while evaluating the new system, tools were developed to automate creation of the Phoebus alarm-service configuration files in the control systems' build process. Additionally, tools and configurations were devised to mirror the old system's key features in the new one. This contribution presents the tools developed and the infrastructure deployed to use the Phoebus alarm-service at HZB.

### Motivation

BESSY II needs to run for at least 10 more years, ALH has been good, but shows its age:

- Single application that requires a GUI
- Each instance is independent, changes require local restart
- Little to no more development (obsolescence threat)

Phoebus is an attractive alternative

- + Actively developed
- + Supported by multiple institutions
- + Configuration and state shared via Kafka
- + Single alarm-server runs as a headless service

### First Steps

Initial setup is simple enough & well documented [2] But then we faced some challenges:

- BESSYII alarm configuration has about 10,000 PVs → we need scripted help (Phoebusalarm)
- ALH runs locally only and works through PVs → we need to replicate the access control we have through CA-Gateways with Kafka (System-Layout)
- Notifying Operators of every alarm is essential for us + adding an automated action to every alarm is cumbersome → we need a better system (Alarm-Notifier)

### Phoebusalarm

#### 1. Python library

- create an alarm tree programmatically
- export to ALH or Phoebus formats

#### 2. Parse ALH-Files into Python alarm tree

- recurse includes
- warn on problematic/ambiguous sections

Example file U125.alh:

```
$BEEPSEVERITY MAJOR

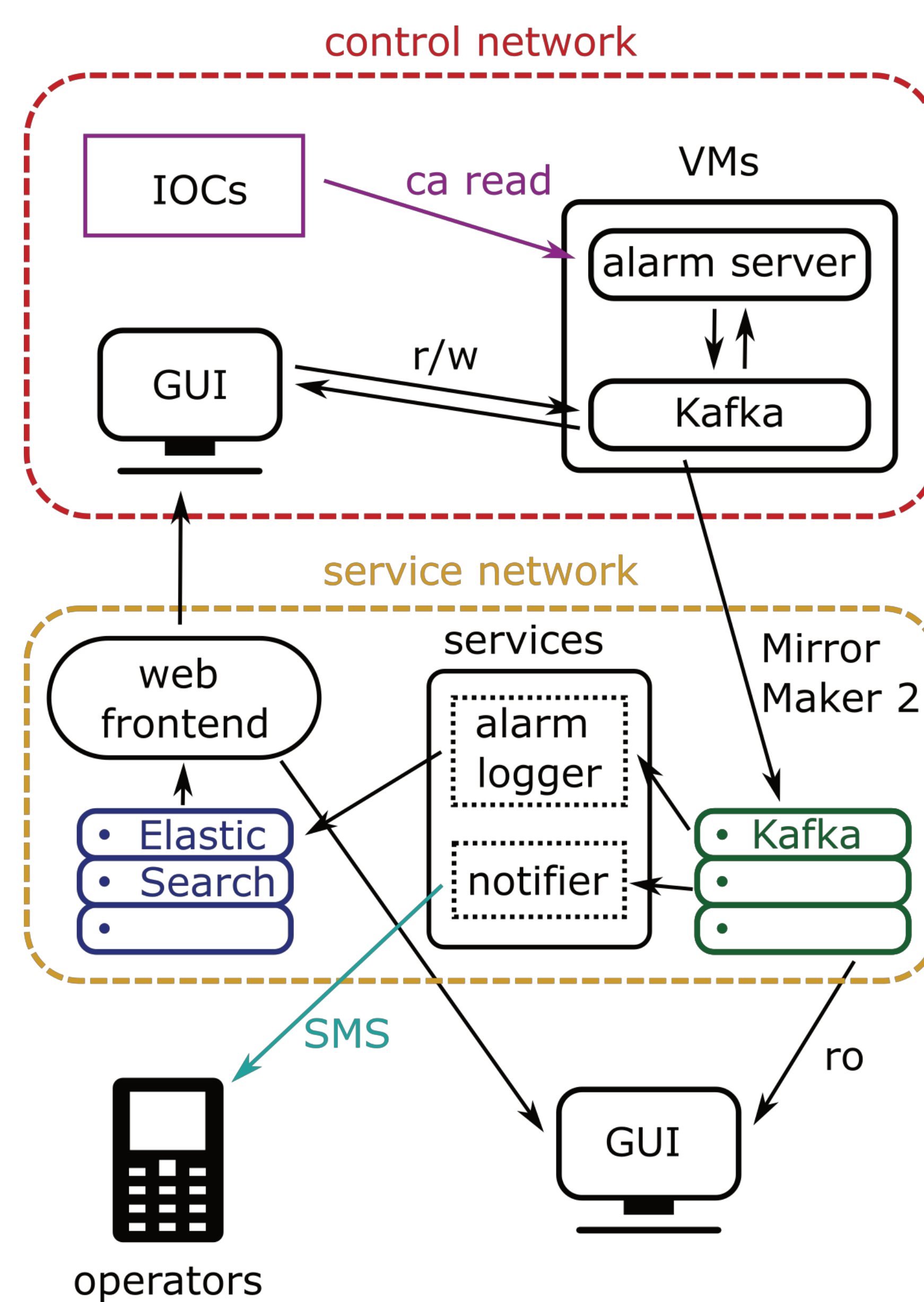
GROUP NULL Insertion-Devices
GROUP Insertion-Devices U125MLS
$COMMAND dm2k -x idcp90.v.ad1
$FORCEPV U125IL2RP:BaseCmdMode CDATL 0 1

CHANNEL U125MLS U125IL2RP:BaseStatHalt
$FORCEPV U125IL2RP:BaseCmdMode CDATL 0 1
$ALIAS Gap-Drive
...
```

Conversion with many warnings, but we know what to fix.

```
> alh-to-xml -v U125.alh
WARNING: Ignoring $BEEPSEVERITY for Accelerator, severity based annunciation filtering not possible in phoebus
WARNING: PV Accelerator/Insertion-Devices/U125MLS uses resetValue 1 for force, phoebus filter will reset immediately once forcePV != forceValue
WARNING: PV U125IL2RP:BaseStatHalt uses resetValue 1 for force, phoebus filter will reset immediately once forcePV != forceValue
WARNING: Replacing inherited filter/ForcePV for U125IL2RP:BaseStatHalt
...
```

### System-Layout



Layout of the components for the Phoebus alarm system

Setup achieves several objectives:

- Control network can work in isolation
- Fairly simple access controls to Kafka: Inside control network everything can write, outside only the mirror process
- Logging and notifications can be unified for multiple control networks

### Alarm-Notifier

**Basic idea:** Use the Phoebus "Talk"-topic (used for annunciation) to send out messages to operators. Can be controlled with the annunciate property.

Pseudo-code like Python-script with confluent\_kafka:

```
from confluent_kafka import Consumer

consumer = Consumer(
    {
        "bootstrap.servers": args.servers,
        "group.id": args.name + str(uuid.uuid4()),
        "security.protocol": "PLAINTEXT",
    }
)
consumer.subscribe(f"{args.name}Talk")

while True:
    msg = consumer.poll(1.0)

    if msg is not None:
        if msg.error():
            logging.error("Consumer error: %s", msg.error())
        else:
            send_sms(f"{msg.desc}: {msg.path}")
```

### EXPERIENCES & CONCLUSION

- While initial installation is simple there is an increased complexity in Phoebus vs ALH
- Going to a full production system may be a challenge (depending on existing expertise)
- Phoebusalarm works very well for us
  - part of our build for a year
  - finds regressions and misconfigurations of ALH as well
- Mirroring Kafka-Clusters allows independent operations and simple access management
  - running for half year
- We can implement our needs and will complete the migration in the next year

#### REFERENCES

- [1] "EPICS alarm handler." (2014), <https://epics.anl.gov/extensions/alh/index.php>  
 [2] Phoebus Developers. "Phoebus alarms." (2017), <https://control-system-studio.readthedocs.io/en/latest/app/alarm/ui/doc/index.htm>

#### ACKNOWLEDGEMENTS

We thank the members of the accelerator control system group for their support. In particular, D. Engel, B. Franksen, V. Laux and S. Heise for their support in the setup of the various systems.

#### MORE INFORMATION

Malte Gotz

[malte.gotz@helmholtz-berlin.de](mailto:malte.gotz@helmholtz-berlin.de)

<https://github.com/hz-b/phoebusalarm>

