Log analysis in the accelerator sector

Steen Jensen, BE-CO-DO

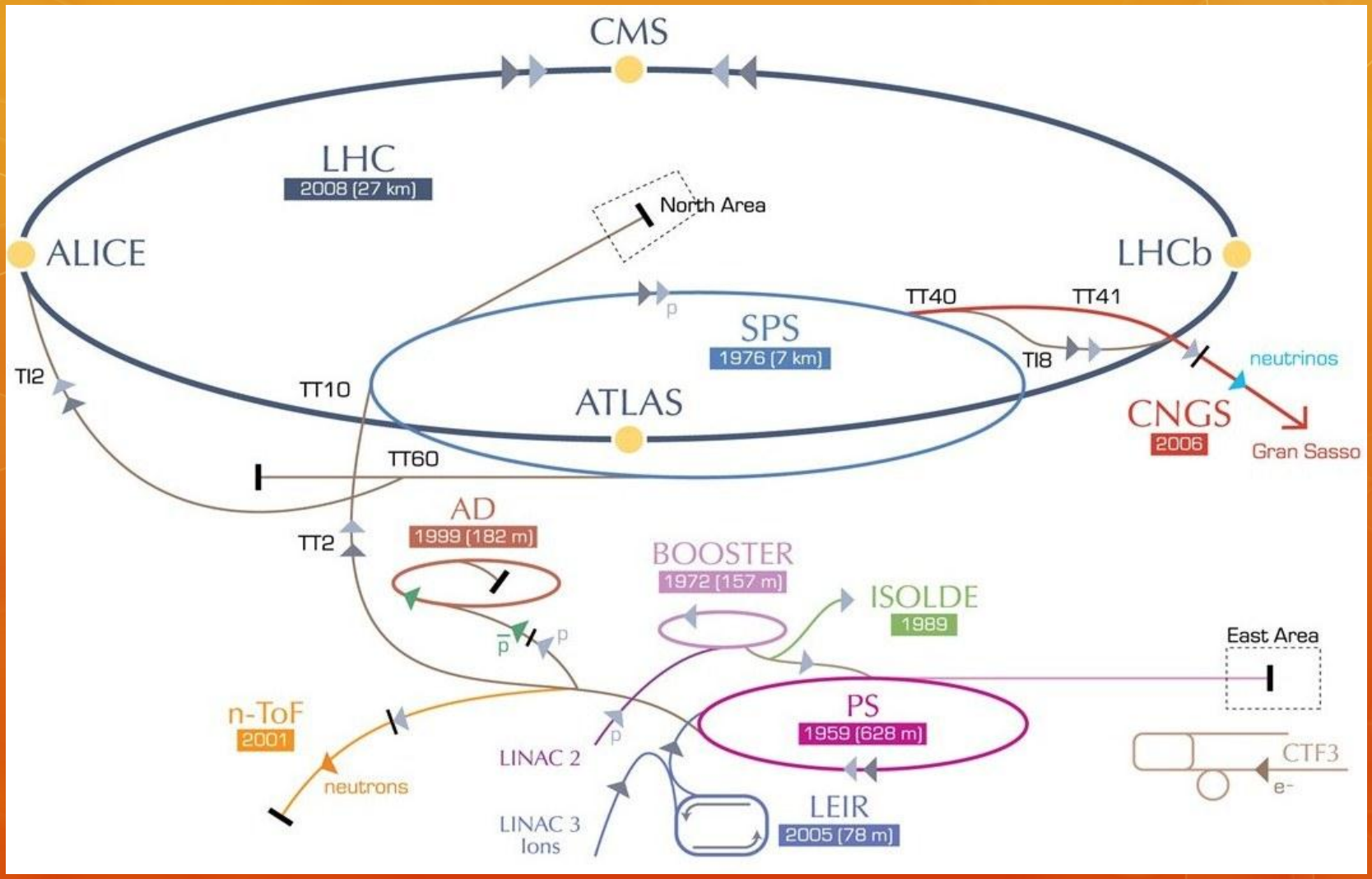
Outline

- Introduction to the controls system
- Our motivation for log analysis
- Requirements
- The current setup
- Use cases
- Conclusions

Outline

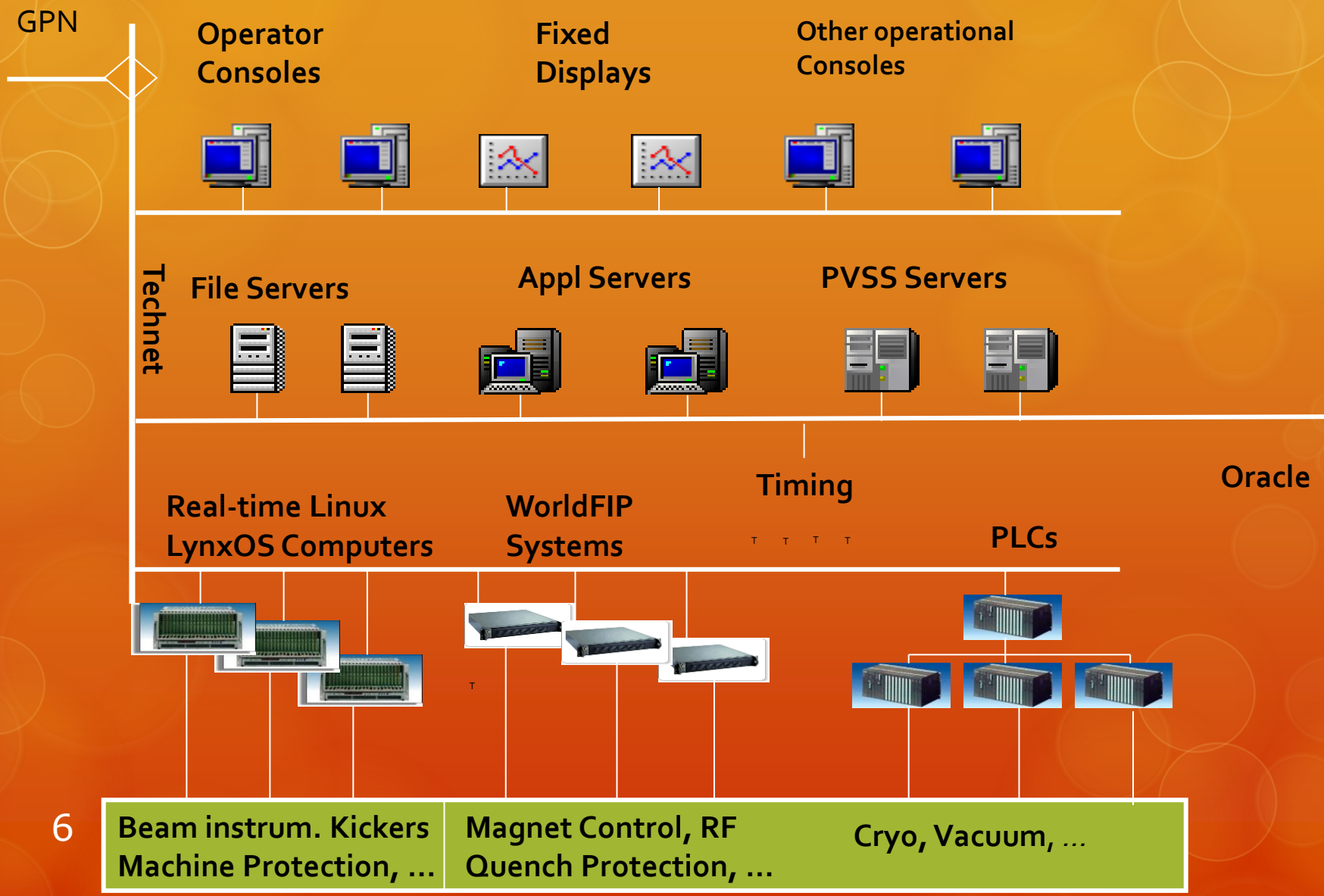
- Introduction to the controls system
- Our motivation for log analysis
- Requirements
- The current setup
- Use cases
- Conclusions

CERN's accelerator complex

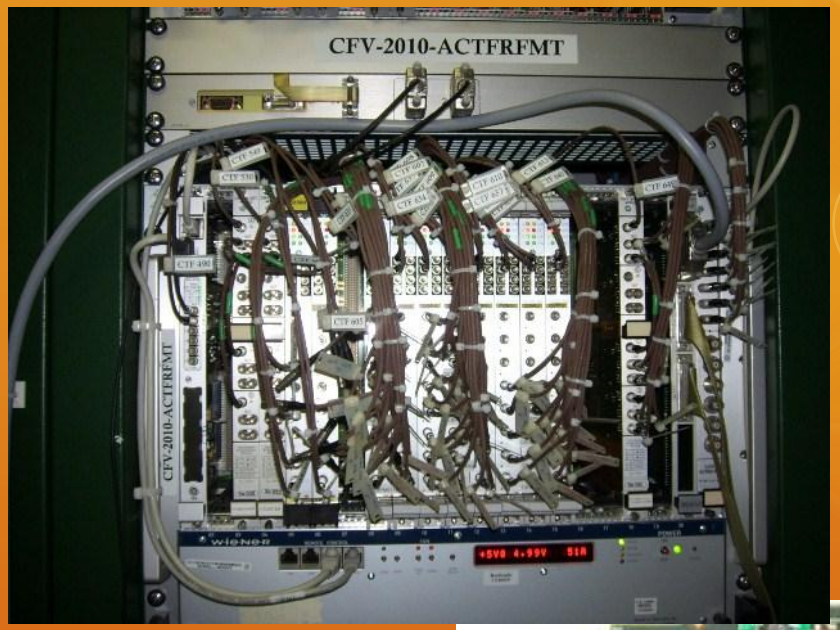


Cern's Control Centre, CCC

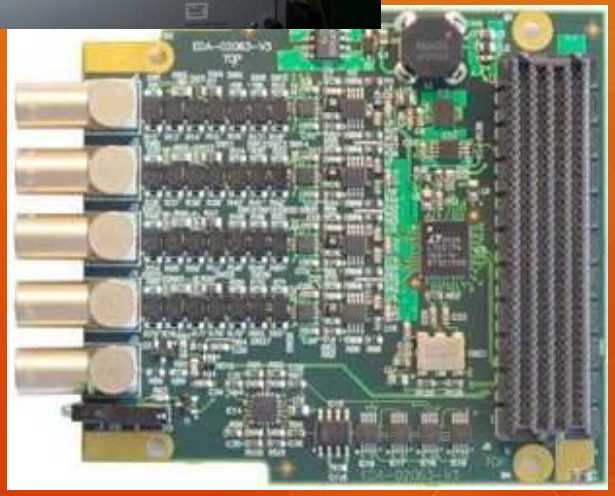
Control system infrastructure



Hardware

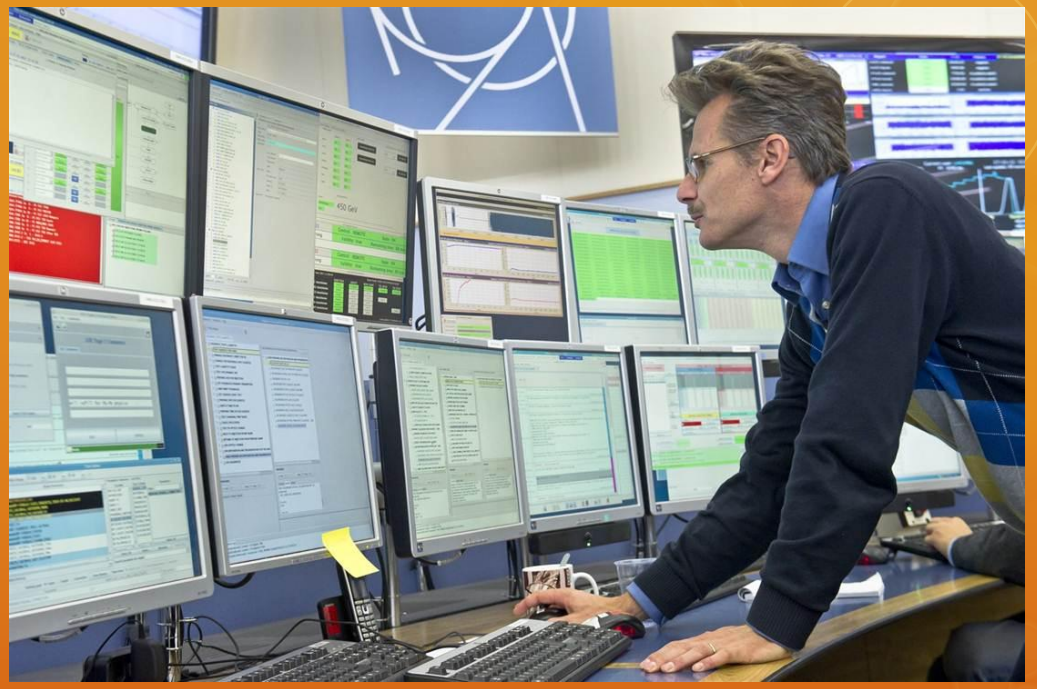


- 425 consoles
- 300 servers
- 1300 front ends
- 600 module types
- ~85.000 device instances



Software

- Java – operational code
  - 400 GUIs, 200 servers
  - ~8MLOC, > 1000 jars
  - Up to 1000 processes
  - 80 people, 10 groups

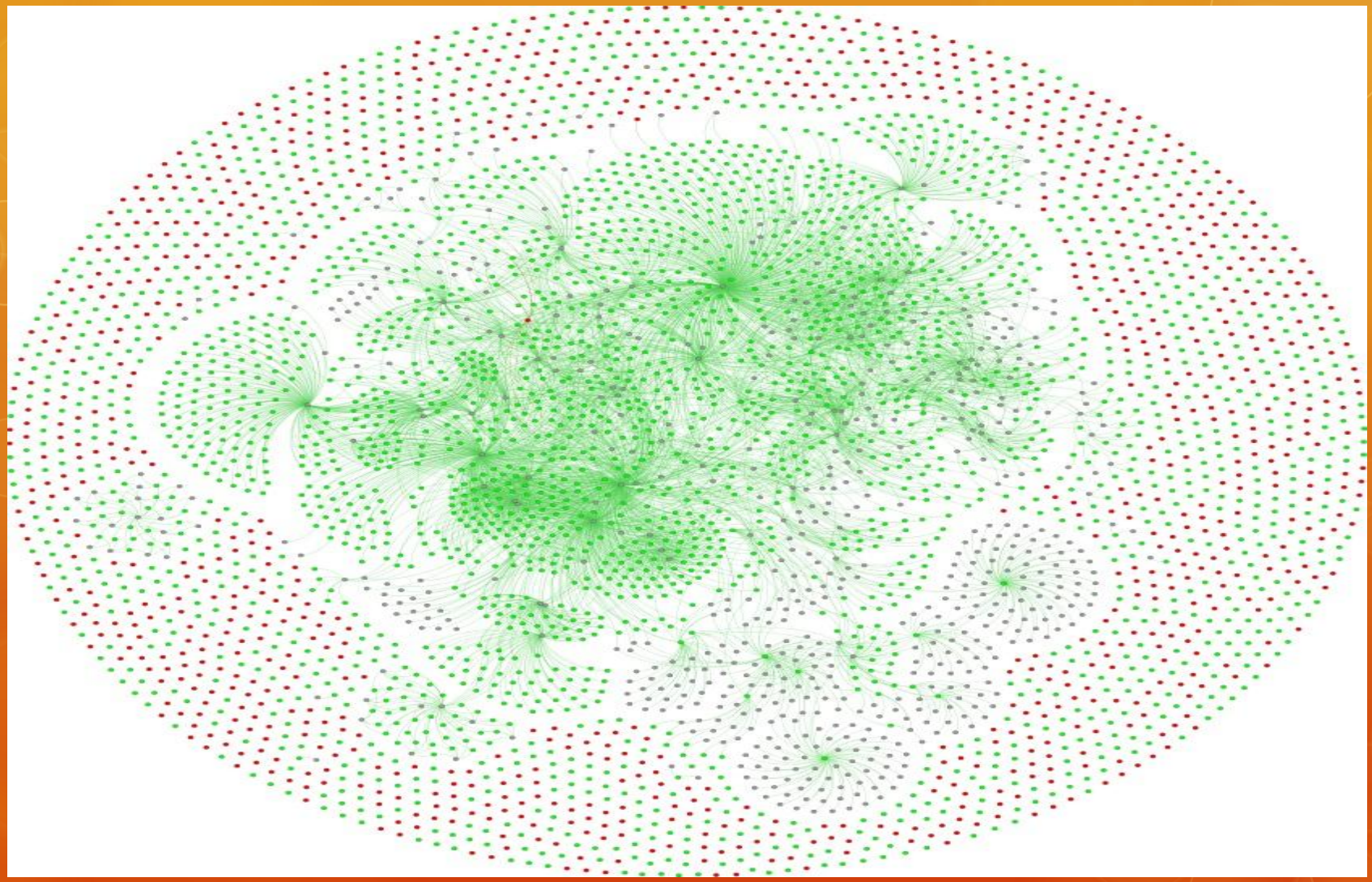


- C/C++
  - > 1300 Front End servers
  - Real-time processes
  - Drivers (Linux/LynxOS)
  - 80 people, 8 groups

Software environment

- Operating systems
  - Linux, LynxOS, Windows
- Languages
  - Java, C++, C, Scripts
- Developers
  - Staff, Fellows, Students – as primary or secondary activity
  - CO, OP and Equipment Groups
- Legacy
  - 10+ years

Connections in middleware layer



Outline

- Introduction to the controls system
- Our motivation for log analysis
- Requirements
- The current setup
- Use cases
- Conclusions

Motivation

- Further improve quality and availability of the controls system
- New responsibility model => non-experts must be able to quickly identify who to call
- Increase diagnostic efficiency
- Widen and deepen diagnostic capabilities
- Proactive & preventive maintenance & evolution
- We already have many log files that can be exploited better

Log data characteristics

- Continuous stream of text messages
- ~2Gb per day, bursts > 10Gb per day
- Multiple transport mechanisms
  - syslog, log4J, JMS
- Scattered, system-specific log files
- Diversity in format and length
- Ambiguity in log level interpretation
- Variations in frequency
- Differences in relevance

Outline

- Introduction to the controls system
- Our motivation for log analysis
- Requirements
- The current setup
- Use cases
- Conclusions

Use cases

- Allow people to diagnose systems other than their own
- Facilitate correlation of messages across systems
- Detect what has changed if something does not work
- Observe trends over time
- Easily obtain customized views of one or more systems
- Receive notifications based on custom criteria

Non-functional requirements

- Easy to use and access by (non-)specialists
- Overviews and graphical visualisations
- Efficient data mining
- Allow for knowledge capture, sharing and re-use

- grep/awk/sed does not provide this

- Looking at Splunk, it is unrealistic to develop own system

Outline

- Introduction to the controls system
- Our motivation for log analysis
- Requirements
- The current setup
- Use cases
- Conclusions

Log data pre-processing

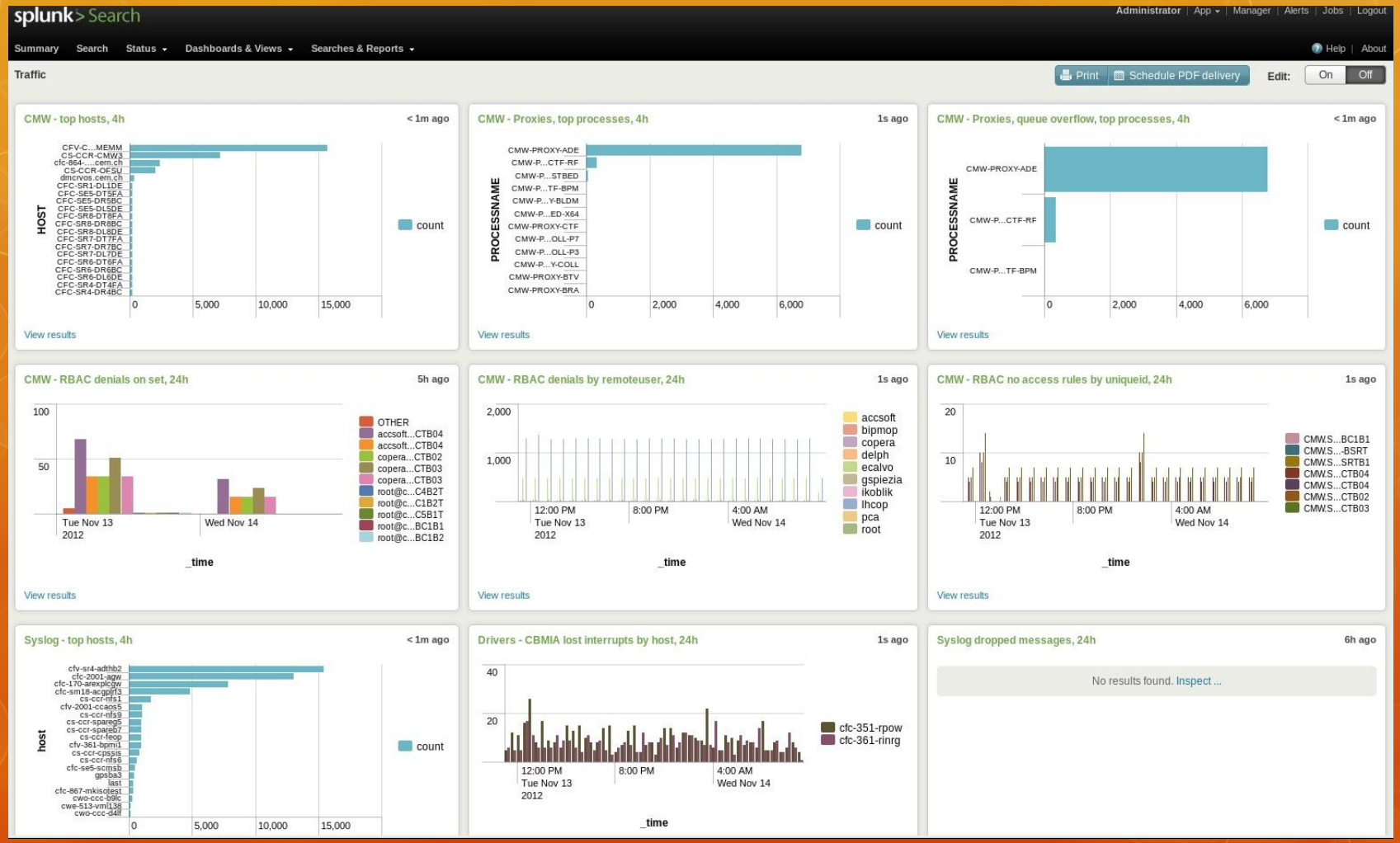
- Centralization
- Filtering of irrelevant messages
- Throttling and burst protection

- Down from 2+ Gb / day to ~100 Mb / day, not counting Java, i.e. likely to increase considerably

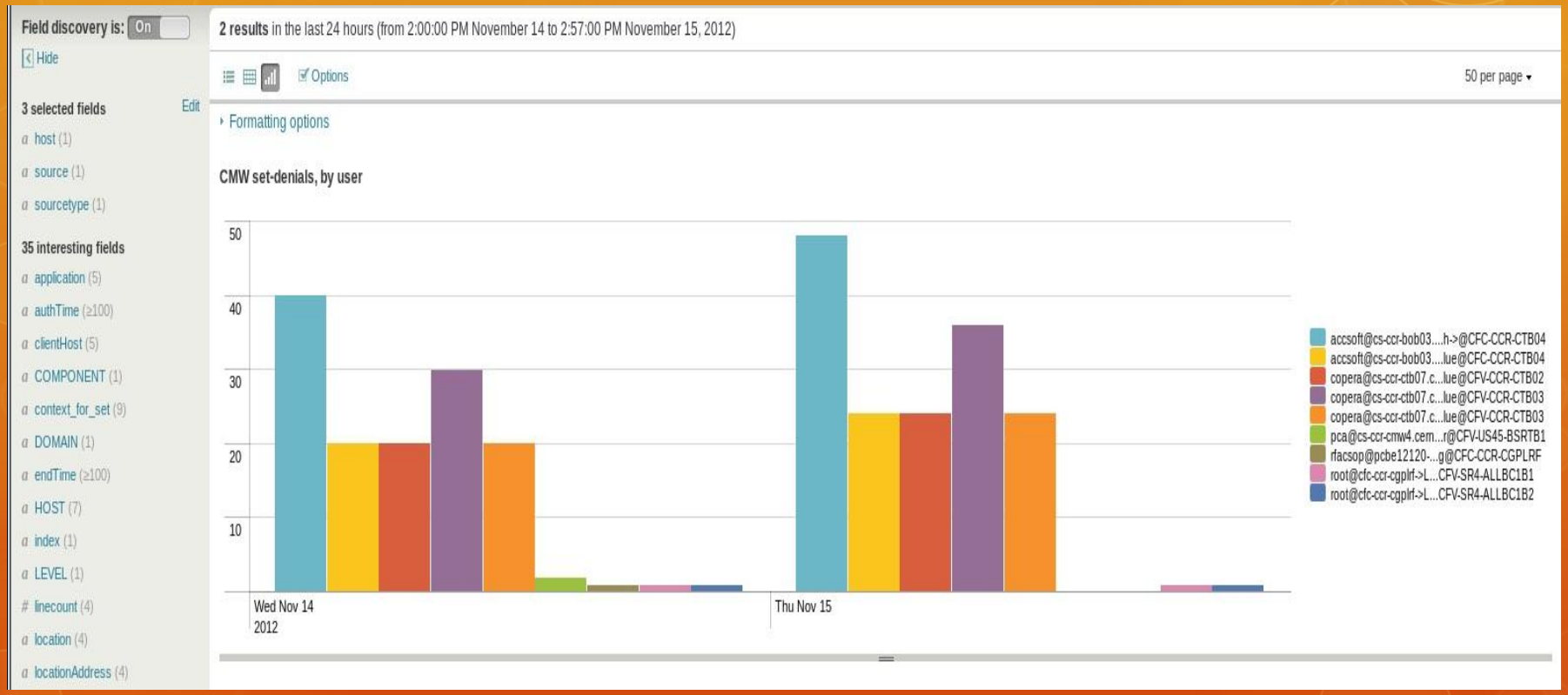
Splunk setup

- Central instance receiving pre-filtered messages via syslog and JMS
- Automated alerts
- Dashboards and saved searches
- Manual monitoring and follow-up

Splunk dashboard



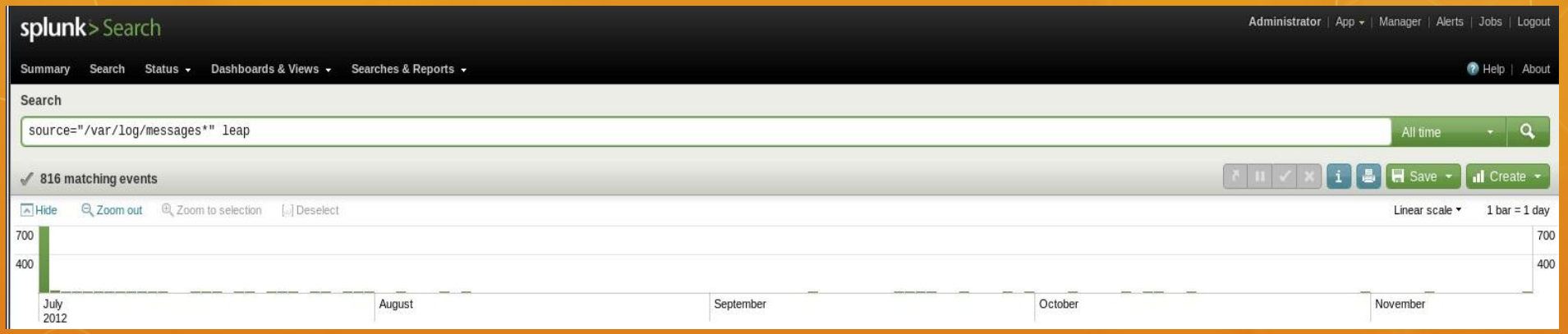
RBAC denials on SET



Outline

- Introduction to the controls system
- Our motivation for log analysis
- Requirements
- The current setup
- Use cases
- Conclusions

Detailed case: Leap second



- On July 1st, 01:59:59 a 60th leap second was added globally
- Using Splunk, a system administrator discovered that
  - Some computers failed to add the leap second on time
  - Certain types of computers added the leap second at a later moment
  - There seems to be no pattern over time
  - No computer added a leap second more than once

Use cases

- RBAC security
  - Tokens missing, malformed or expired
- CMW Client issues
  - Slow consumer, zombies, incorrect accesses, error handling
- Timing system
  - telegram layout issues
  - Performance testing of new system
- CO Frameworks
  - Improvement: Token check prior to access
  - Bug: applying wrong token in certain cases
  - Bug: Loop on timing error

Use cases, continued

- Design considerations
  - Architectural decisions based on usage info
- Driver error reporting, missing module
- System issues
  - yum updates misbehaving due to race condition
- Separation between operational and test environments

Outline

- Introduction to the controls system
- Our motivation for log analysis
- Requirements
- The current setup
- Use cases
- Conclusions

Conclusions

- Log analysis is very valuable – we learned a lot using Splunk
  - Knowledge about control system usage
  - Pro-active detection, trimming, improving
  - Re-active diagnostics
- It is a continuous discover-learn-improve process
- It involves a mix of
  - automated alerts
  - manual monitoring
  - ad hoc data mining
- Experience allows improving log data quality
  - Culture: Developer usage
  - Structure: Key-value pairs
- Splunk is a very promising tool in CO's context, and projects show strong interest