Logbook printing with TINE API calls

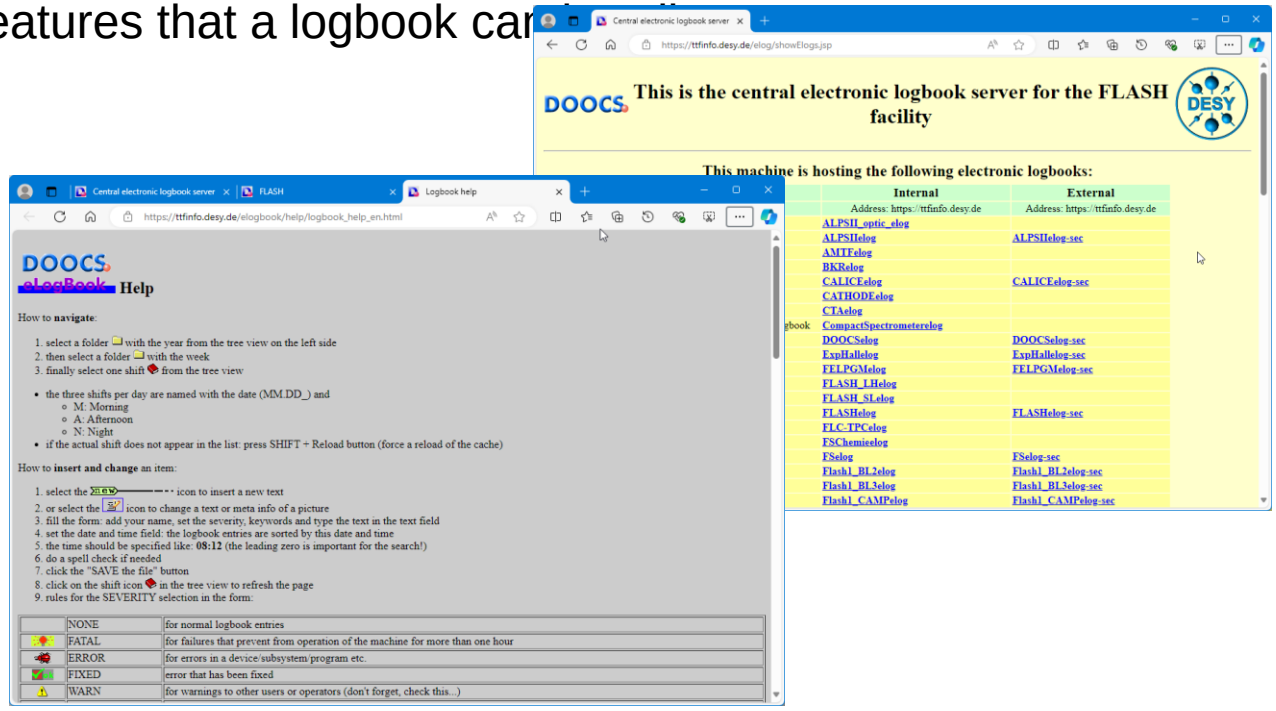
Marcus Walla

Hamburg, November 27 2024

Motivation

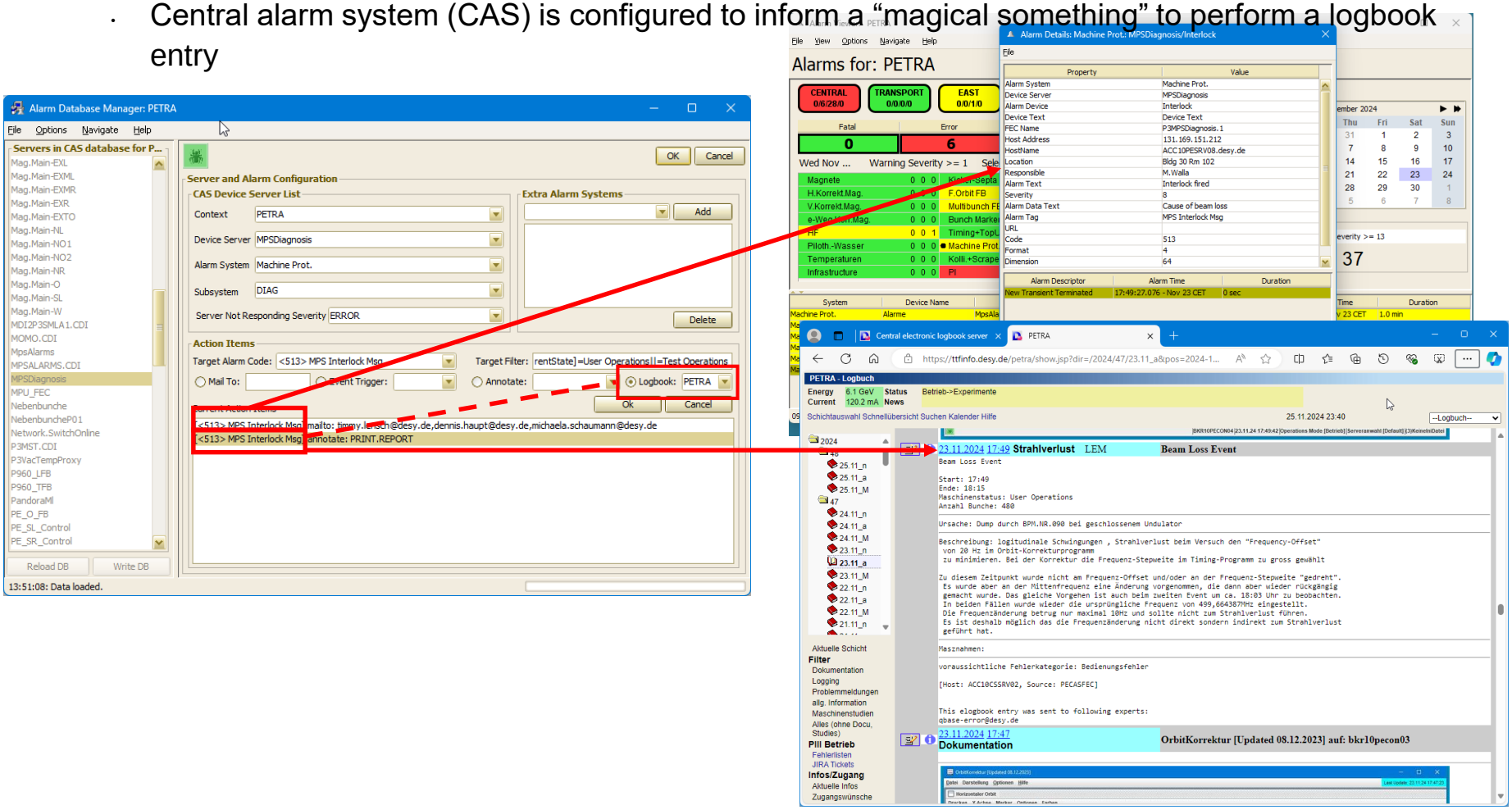
Would be nice to ...

- have a TINE API call to “print” to the logbook
- should use all of the features that a logbook can
  - Author
  - Header
  - Severity
  - Keyword
  - Message
  - Possible an image

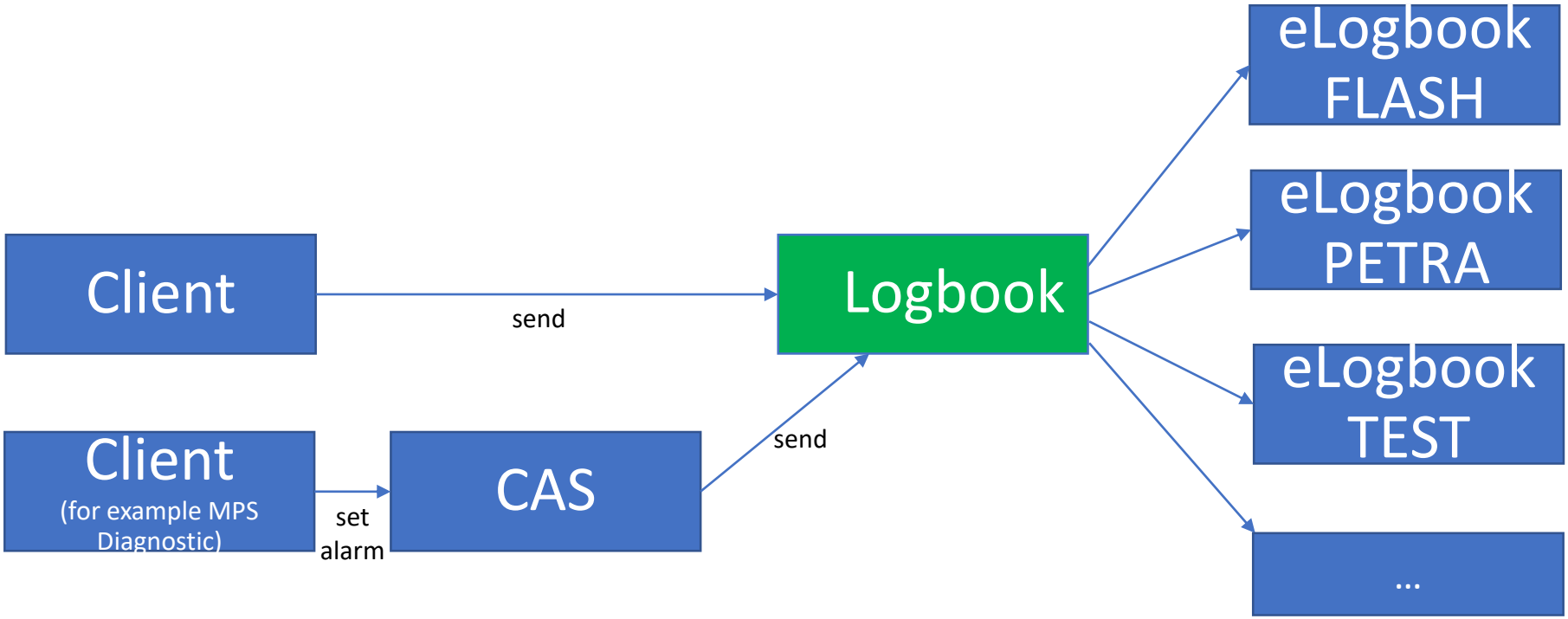


Proof of Concept

- MPS Diagnostic server for PETRA
  - When a beam loss event is triggered, it analysis the cause and sets an alarm to the CAS
  - Central alarm system (CAS) is configured to inform a “magical something” to perform a logbook entry

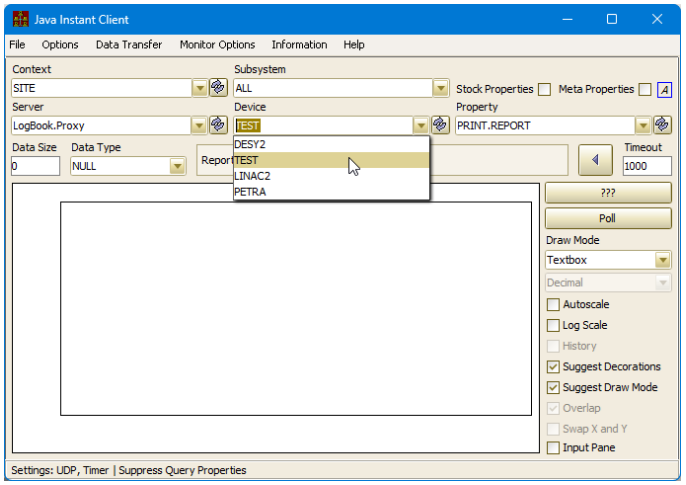


What is behind this ‘magical something’?



TINE Addresses

- Base part is an address from the TINE name space
  - /SITE/LogBook.Proxy/<*queue*>/PRINT.REPORT
    Data type = {TEXT, TLogBookMsg (TINE structure)}
  - /SITE/LogBook.Proxy/<*queue*>/PRINT
    Data type = {TEXT, TLogBookMsg, TAlarmLogBook (TINE structures)}
- <*queue*> = {LINAC2, DESY2, PETRA, TEST, …}
  - Depends on the installed printer queues (ask admins or install it by yourself)
    => server will automatically configure the correct devices as an abstract end-point of the logbook
  - Windows: regex matches:
    \\\\adprint[1-9]\d?(\.desy\.de)?\\\w{3,10}\d?log
  - *nix: similar, but not implemented
- MPS Diagnostic: special parsing of the data type TEXT -> ../PRINT.REPORT



Example with image

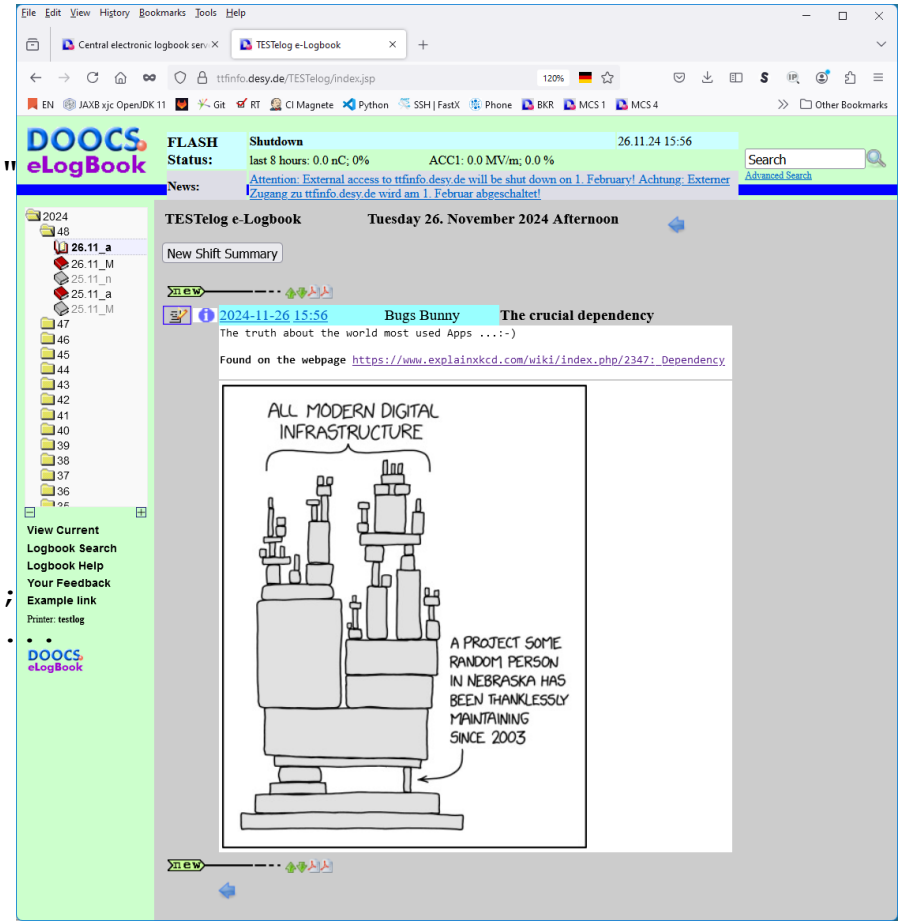
Tested: PNG, TIFF, BMP, GIF (not animated!)

Will be added soon to the TINE lib!
(C and Java)

```
IMAGE img = new IMAGE();
byte[] image = enocde(new File("H:/Projects/LogBookProxyServer/resources/images/
                                dependency.png"));
img.fillFrameBuffer(image, 0, 0, image.length);
IMAGE.FrameHeader header = img.new FrameHeader();
header.appendedFrameSize = image.length;
header.sourceFormat = IMAGE.IF_BASE64;
img.setFrameHeader(header);

TDataType din =
     new TDataType(new TLogBookMsg(..., img));
TLink lnk = new Tlink("/SITE/LogBook.Proxy/TEST"
                      , "PRINT.REPORT"
                      , TDataType.nulltype
                      , din
                      , TAccess.CA_WRITE);
lnk.executeAndClose();

static byte[] encode(File file) throws ...
{
   if (!file.exists())
      throw new FileNotFoundException(...
   byte[] bytes = new byte[(int) file.length()];
   try(FileInputStream fileInputStreamReader = ...
   {
      fileInputStreamReader.read(bytes);
   }
   return Base64.getEncoder().encode(bytes);
}
```



But be warned:

it is preliminary and still needs some improvements!