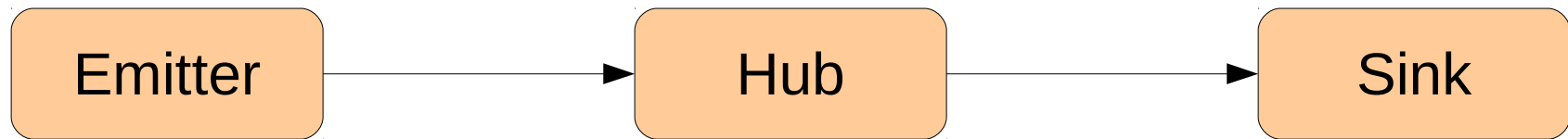


A large, detailed wireframe model of a particle accelerator, likely the FAIR facility, is centered on the page. The model shows a complex arrangement of circular and linear sections, representing the path of the particle beam. The text is overlaid on this model.

# Diagnostic Logging Report

C. Handel, I. Križnar, R. Mueller  
25.07.2011  
BEL Technik Presentation



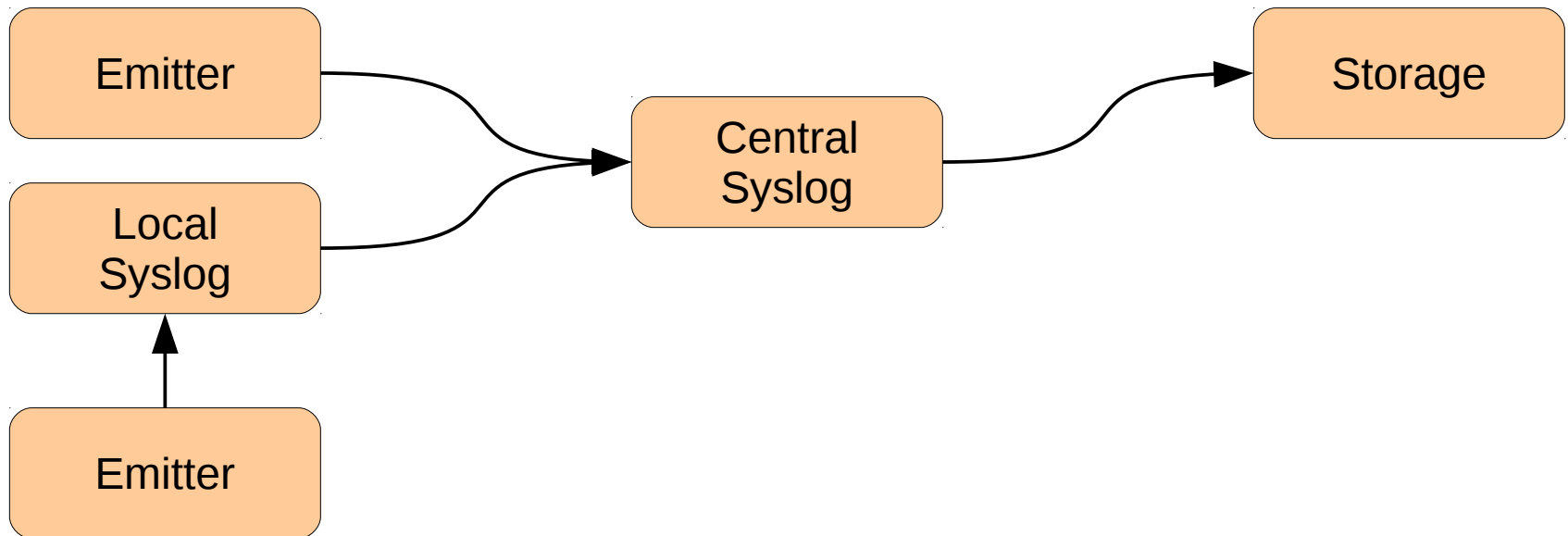
- ◆ Emitters we can influence are written in
  - ❖ Java
  - ❖ C/C++
  - ❖ Fortran
  - ❖ Maybe some scripts (python, shell)
  
- ◆ Emitters we can't influence are Products like
  - ❖ Operating Systems (Linux, Windows)
  - ❖ PVSS/WinCC
  - ❖ Maybe some cern specials (CMW)

- ◆ Files
  - ❖ Various places
  - ❖ On local and shared filesystems
- ◆ Syslog Logsv01
  - ❖ Used by Devman on PowerPC
- ◆ Syslog
  - ❖ Machine local syslogs

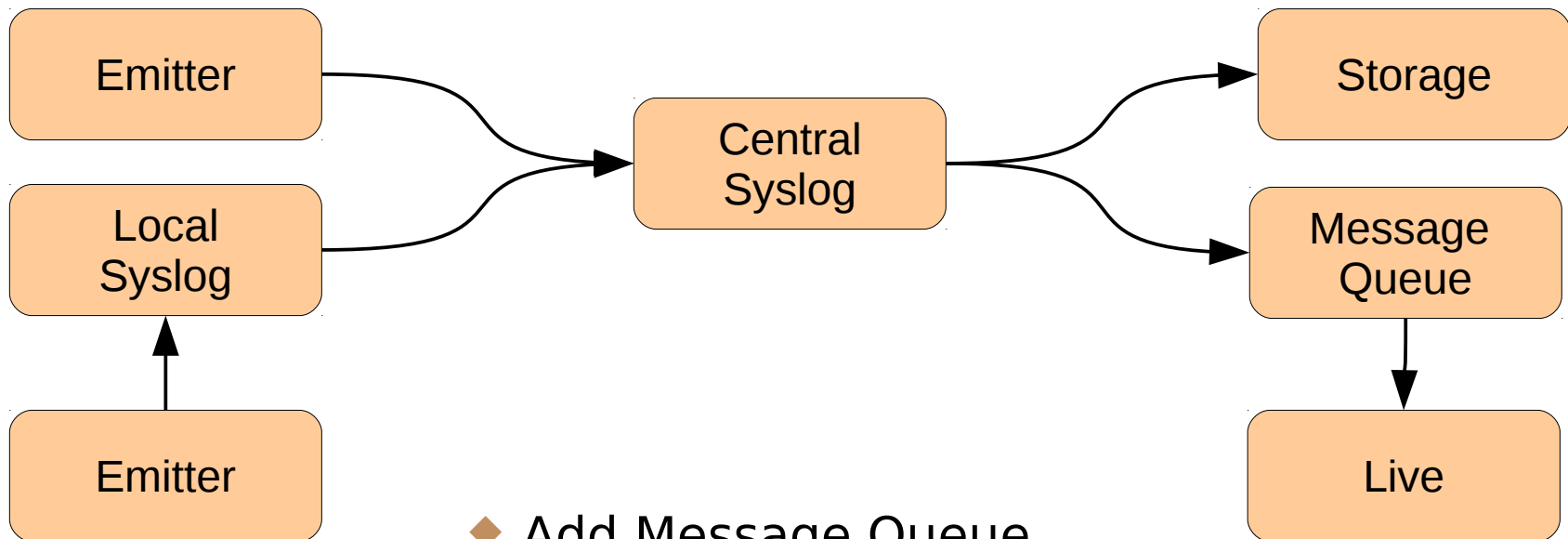
- ◆ Welcome to grep
  - ◆ This is not a negative point. There are various linux tools to search, filter, sort, parse textfiles.
- ◆ Direct acces to logfile

- ◆ Normal Logging
  - ❖ Starting, stopping a service
  - ❖ Beginning, completing tasks
  - ❖ Having problems
  - ❖ Internal errors in actively developed code
- ◆ It's not part of Diagnostic Logging to capture
  - ❖ debug information from components which are not active part of the controls development process, such as:
    - ◆ core dumps
    - ◆ fatal stacktraces
  - ❖ Stderr/Stdout (unfiltered garbage streams)
  - ❖ These features would be nice
    - ◆ But requires a wrapper around the service
      - What happens if the wrapper crashes?
      - May be during debugging, but not default.

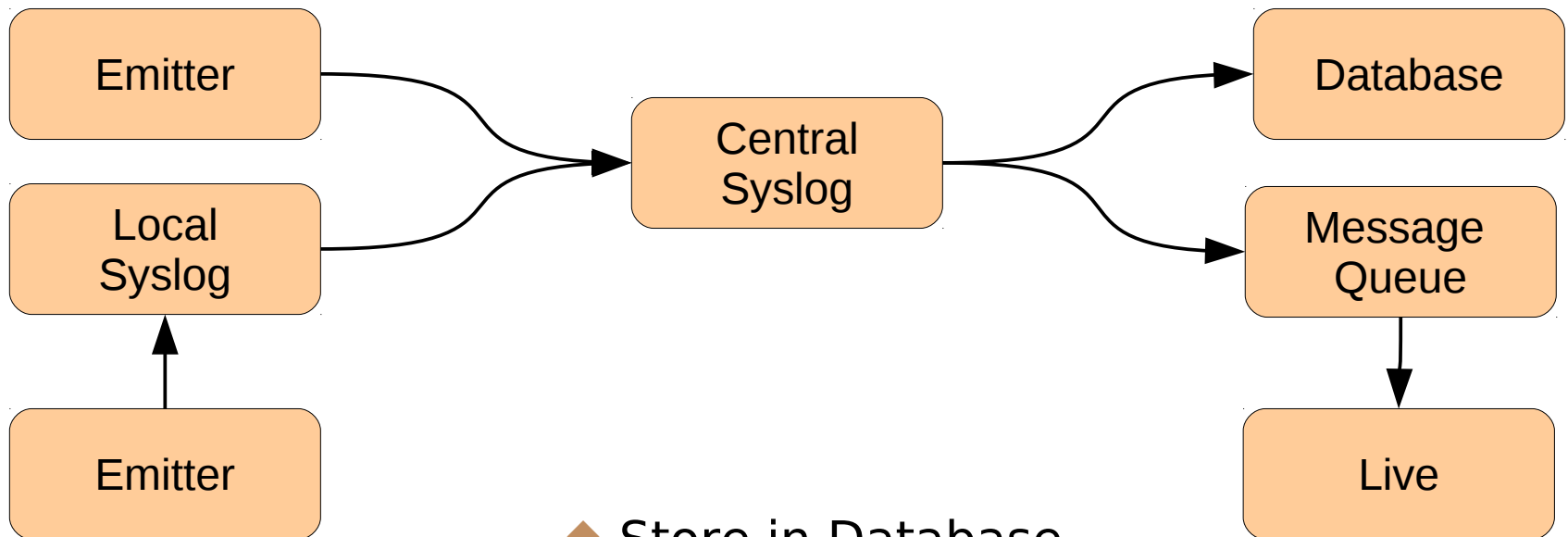
- ◆ Diagnostic
  - ❖ What happened last week?
  - ❖ How many times did a service crash?
  - ❖ Statistics/Trends (number of changes, crashes)
- ◆ Live
  - ❖ What is happening at the moment?
  - ❖ Watching/Monitoring of applications freshly released.
- ◆ Operational
  - ❖ Who is to blame for screwing the beam?
  - ❖ What values were supplied to equipment.
  - ❖ Which parameters changed at what point in time.







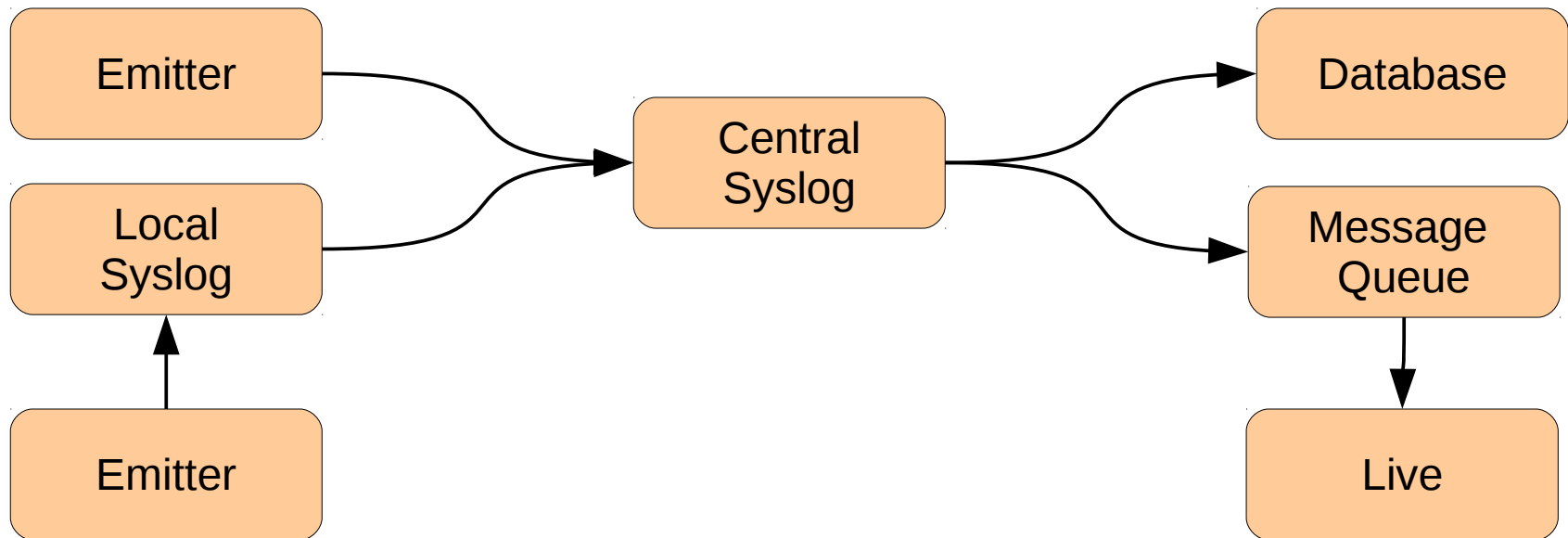
- ◆ Add Message Queue
  - ❖ Message Push
  - ❖ Dynamic Consumers



- ◆ Store in Database
  - ❖ Remote Access
  - ❖ Minimal Structured Access

# (A) Evolutionary Concept

- ◆ Syslog is well established
- ◆ Bindings in nearly any language, usually by default
- ◆ Free form message body
- ◆ Support in closed/foreign products
- ◆ Limited payload
- ◆ No guaranteed delivery
- ◆ Limited structure
- ◆ Limited filtering Capabilities



# Why not?

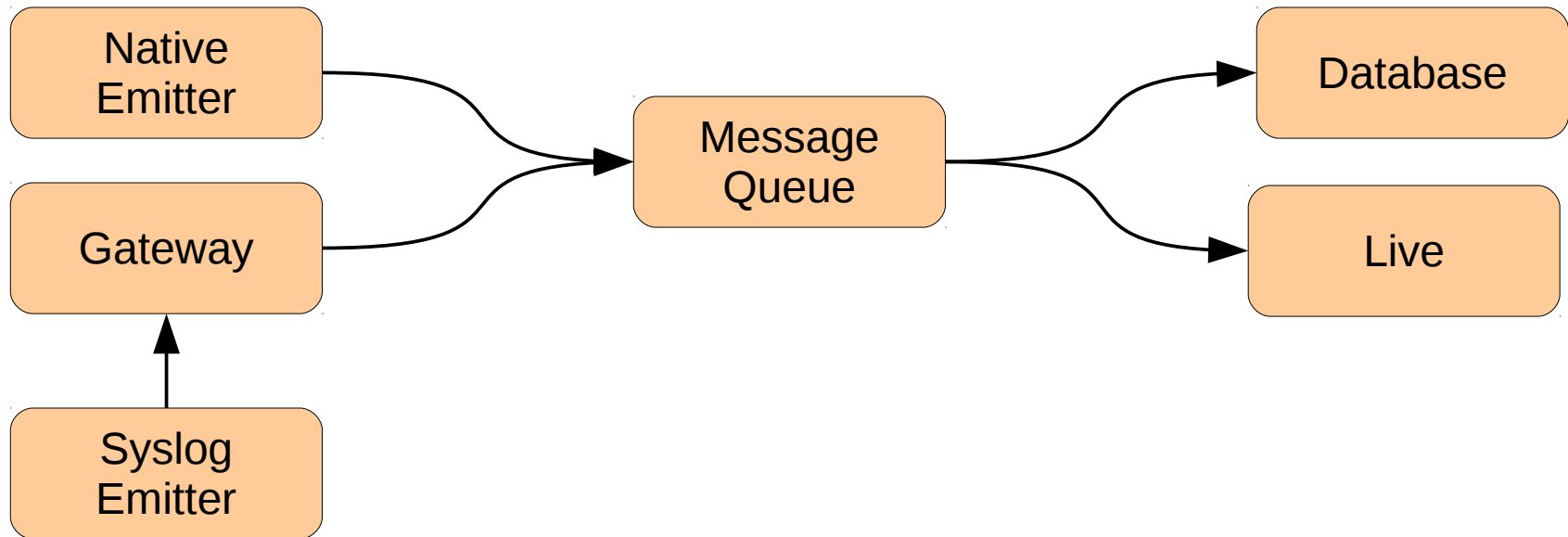
use the Message Queue as core?

It's a

Revolutionary Concept ;)

## (B) Message Concept

- ◆ Structured messages
- ◆ Flexible (e.g. sink attachment, properties)
- ◆ Separate system from application logging
- ◆ one-to-many publishing
- ◆ Language bindings (by message queue library)
- ◆ Not a complete solution (set of tools)
- ◆ Heavier (more complex messages)



## Syslog

- ◆ Well established
- ◆ Performance
- ◆ Existing Bindings
  
- Explodes on multiline messages
- Limited Filtering
- Static Config

## Message

- ◆ Flexibility
  - ◆ Fields
  - ◆ Filtering
  - ◆ Sink Attachment
- ◆ Structured
  
- Language Bindings
- Complexity
- Heavyweight
- Implementation Effort

- ◆ Client API
  - ❖ Toggle Debugging
  - ❖ Configuration (productive Central, development Local)
  - ❖ Simple Usage
  - ❖ Non-Blocking
- ◆ Live Viewer
- ◆ History Researcher
- ◆ Component Integration