# Axis in FTS

*Zsolt Molnár*

*CERN, IT_DM*

# gSoap

- **Cross-platform C/C++ SOAP implementation**
  - Client stub / service skeleton generation
    - From WSDL
    - Only C/C++ code generation
- **Some problems**
  - Generated code needed to be modified
  - Sometimes complicated implementation
  - Unclear API
  - Missing high-level SOAP fault handling
- **Current C/C++ FTS code is based on gSoap**
- **Do we have better alternative?**

# Axis

- **A SOAP implementation form Apache**
  - Client stub / service skeleton generation
    - From WSDL
    - C, C++, Java
    - Single SOAP platform
- **Clear API, good quality generated code**
  - Object oriented C code
  - Self-describing interfaces
  - Seamless SSL handling
- **AXIOM**
  - The Axis Object Model
  - XML processing engine
  - Incrementally builds the memory model
    - Big messages...

# Axis in FTS

- **Studied its potential application in FTS**
- **FTS: client in C, C++ - services in Java as well**
  - Actually: C/C++ with gSoap, Java with Axis
  - Ideally: single SOAP platform for everything - Axis
- **Created a proof-of-concept client/service**
  - *glite-transfer-channel-list*
  - No high-level fault object support
    - Can be handled in AXIOM level
    - Next Axis release would provide it
    - Situation is line in gSoap case
  - Created an intermediate, specific fault handling API
- **Working with Axis was fast, development was easy**
- **To be continued: performance, robustness, proxy certificates, etc.**

# The FTS WSDL

- **Actual WSDL**
  - Generated from Java interface
  - Not WS-I compliant
    - May be interoperability problems
    - Code generation problems
  - Missing a design
  - Wrong fault handling approach
    - No fault objects, message „unions”
  - WSDL recommendations/standards not always followed
- **Effort to re-write the WSDL**
  - “Manually” - we have design...
  - WS-I compliance guaranteed
    - Validated with Eclipse
  - Python ZSI compliant
    - primary and most active web services toolkit