The new FTS – proposal
FTS status

FTS 2.2.6

- Bugs fixed
  - Support an SE publishing more than one SRM endpoint
  - FTS kills transfer while dCache is computing checksum

  More
  https://savannah.cern.ch/patch/?func=detailitem&item_id=4503
- New features
  - Restart / resume failed transfers
    - Restart: GridFTP client build-in plug-in
    - Resume: range markers are kept in memory, use them if transfer fails to resume
      - Precondition: range markers MUST be sent back from the GridFTP server

FTS 2.2.7

- New overwrite files logic
  - If “-o” flag is used
    - Check first if file exists
    - If it exists, delete it
    - Finally, initiate the transfer
      - both srm-copy and url-copy supported
      - Works with srm and gsiftp endpoints

FTS 2.2.8

- New feature
  - Transfer monitoring
    - Instrument FTS instances to report monitoring information
    - ActiveMQ will be used
    - 2 types of messages
      - Transfer events (for every transfer)
      - Queue status reports (sent at regular time intervals)
    - https://twiki.cern.ch/twiki/bin/view/LCG/WLCGTransferMonitoring

Why new FTS?

- Architectural limits reached
- Difficult to maintain and improve
- Based on static channel model
- Uses Oracle backend
- Overlaps with other projects
- Requires complex configuration
- Could handle resources better

Schedule + new features

- Step 1: end of 2011
  - Remove channel model, more transfer protocols (HTTP), replace Java WS, new config, support MySQL?
  - Prototype!
- Step 2: end of 2012
  - Messaging, transfer optimization
  - LHC shutdown starts
- Step 3: middle of 2013
  - Merge FTS and LCG_Util
- We need to set up priorities

Channel model

- Reflects hierarchical Tier model
  - Not scalable in Tier2 domain
- Static properties set by admin
  - SE pairs, mutual agreement
- Channel groups, star channel: still not optimal
- SE and network load not taken into account
- Dropping channel model → dropping FTS architecture
  - Users welcome the idea

Channel model - solution

- Take SE into account
  - Need to discuss it with other SE providers(CASTOR done)
- Take network load into account
  - Collaboration with OSG
- Step 1: no channels
  - SE-s are configured on FTS side
  - Network utilization measured by FTS
- Step 2: using the new interfaces
  - Resource handling: heuristics based on performance of last transfers
  - Ultimately, no configuration needed?
- Open questions. Fair share: how? Control throughput?

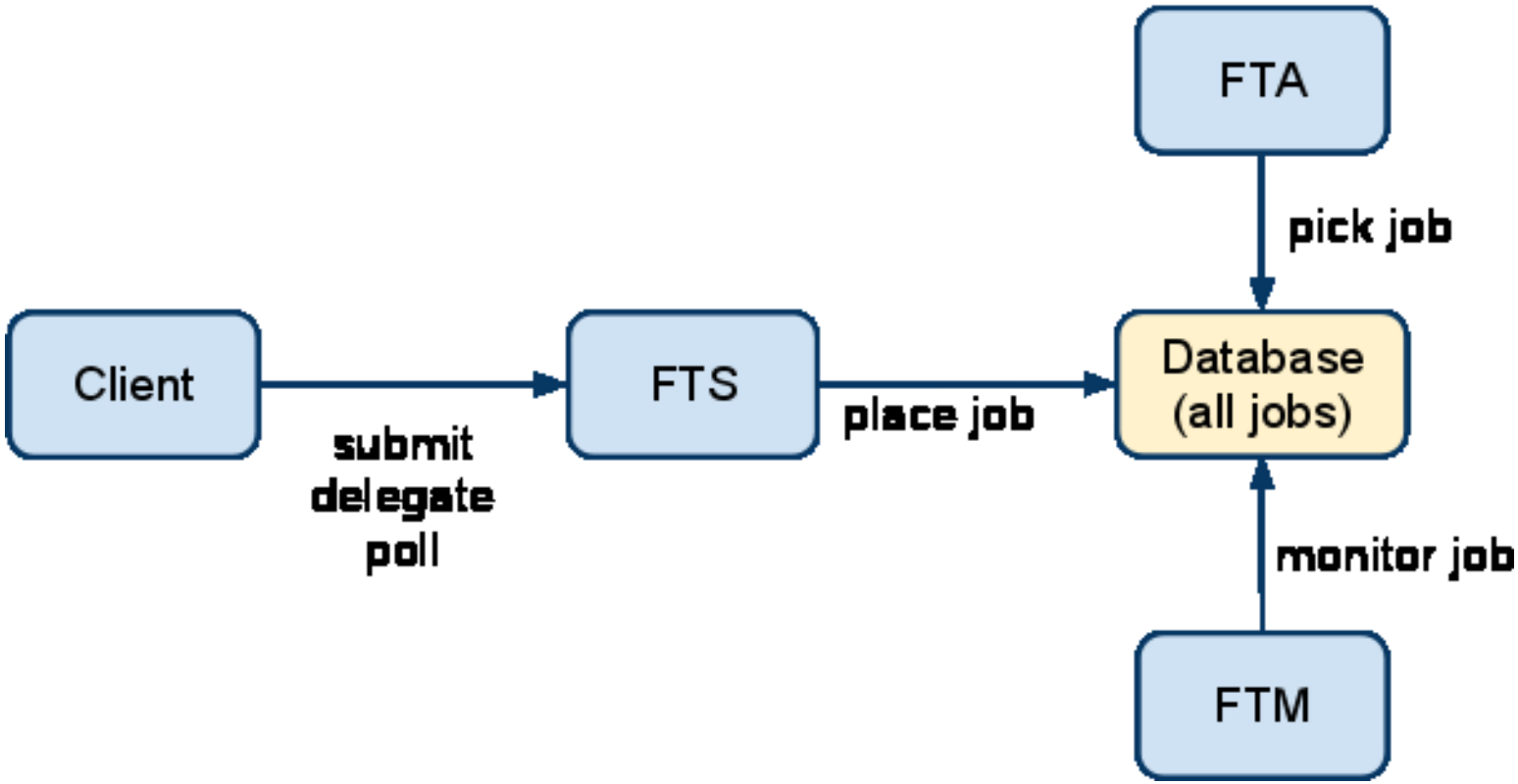
Transfer protocols, WS

- Currently, SRM + gridFTP
  - LCG_Util supports more
  - Step 1: Transfer plugins
    - SRM, gridFTP
  - Step 2: HTTP, XRDCP
- C++ client ↔ Java web service
  - Duplicates, knowledge of Tomcat, etc. needed
- Solution: everything in C/C++
  - Step 1: FTS in C++, based on new generic components
  - Step 2: No web service (decision pending, still balancing drawbacks/benefits)

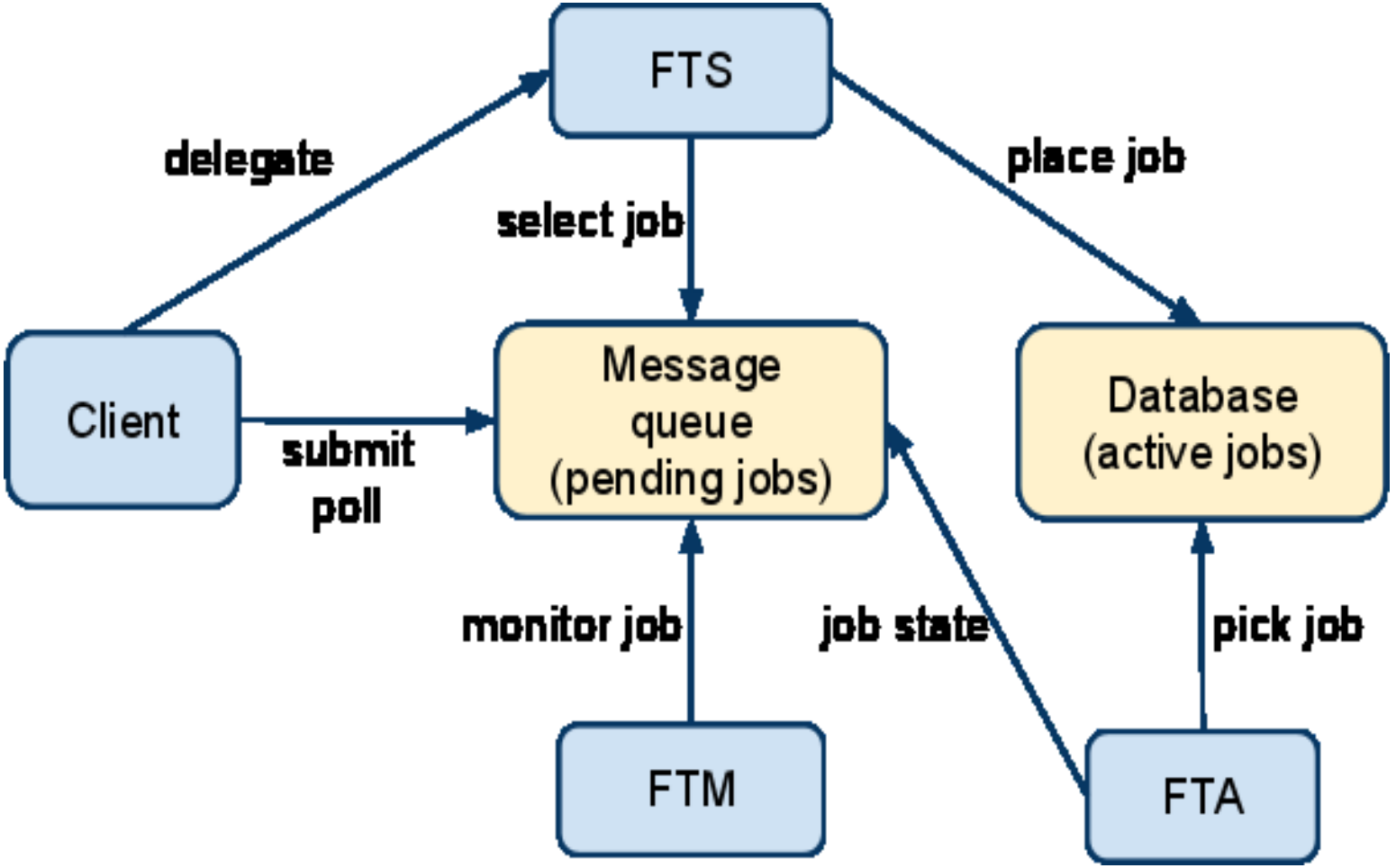
Persistency, configuration

- Using Oracle only
  - Problem for Tier2 sites: no license, etc.
  - FTS deployed in T2-s is not a hot use case now - later?
- Step 1: FTS on top of generic DB access
  - Database access plugins
    - MySQL, ProgreSQL implementation if needed
    - Clear architecture
  - Simpler schema (no channels)
  - Configuration goes to database
    - Simple CLI, query and (re)configure remotely
- Step 2: Job queue based on messaging
  - Global FTS queue? Transparent FTS-es? Monitoring?
  - Problem with delegation

File Transfer Service now



New FTS


FTS
delegate
select job
place job
Client
submit
poll
Message
queue
(pending jobs)
Database
(active jobs)
monitor job
job state
pick job
FTM
FTA

Resources

https://svnweb.cern.ch/trac/glitefts

Jean-Philippe Baud
Oliver Keeble
Zsolt Molnar
Michail Salichos
+ our students and short term visitors