FTS & XRootD Workshop

Jozef Stefan Institute, Ljubljana, 27-31st March 2023

FTS

File Transfer Service

FTS Planning Session

CERN

Goal of the planning session

I. Define planning session rhythmicity

II. Go through list of focus areas

III. Discuss impromptu topics

FTS - Steering meetings?

Last FTS Steering Meeting: too long ago

Objective: define the new purpose of the Steering meeting

Current situation:

- Actively involved with the big experiments (DOMA[-BDT], ATLAS/CMS + IT-Storage meetings, ad-hoc meetings)
- Not regularly involved with other communities

FTS - Steering meetings?

Last FTS Steering Meeting: too long ago

Objective: define the new purpose of the Steering meeting

Proposal:

- Try out FTS Community Forum (Discord forum, lightweight account needed)
- Regular, semestrial “check-ups” (not more often than every 6 months)
    - Involve communities for a course-correction meeting
    - Half-day, Zoom?
    - Next one due in September 2023

      (exact date to be announced by e-mail: fts3-steering@cern.ch)

FTS - Next areas of focus?

Service operations

- True microservice model
- Reduce components coupling
- Add built-in service protection
- Add built-in service health monitoring

Community requests

- FTS global config
- Aggressive Optimizer
- User-friendly Cloud configuration
- Improved HTTP(TPC) error reporting

Project evolution

- High Luminosity LHC
- Modernise codebase
- Deterministic + Global view scheduler
- DMC clients evolution
- Transfers for non-WLCG environments
- Activity & Priority discussion

- Need a way to prioritise between service and stakeholder requests

**Lists not exhaustive*

Service Operations

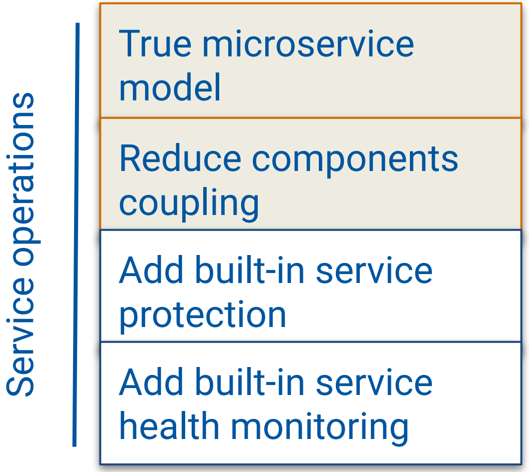
- Current model does not allow for “microservice” model
- Components are coupled by hidden dependencies

Goal:

- Allow each component to be deployed separately
- Permit horizontal scaling of individual components

Plan:

- Each component runs with own config file ✅
- Remove code assumptions that other components are running on same host ⏳
- Handle transfer logs centrally from within the software ⏳


Service operations

True microservice model

Reduce components coupling

Add built-in service protection

Add built-in service health monitoring

Service Operations

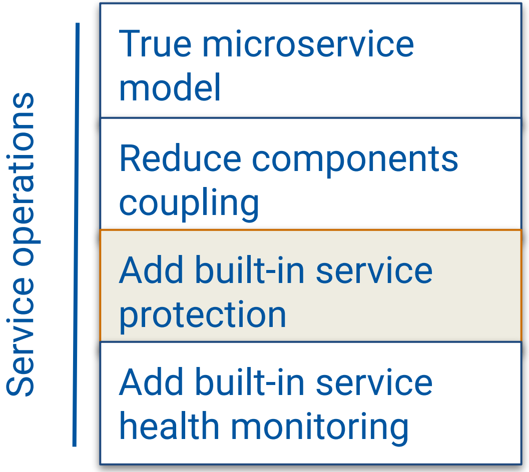
- System has no concept of protecting itself when stressed
- Example: system never refuses submissions

Goal:

- Make system aware of capacity
- Implement service protection when capacity surpassed

Plan:

- Submission limits ⏳
- Improved memory management ⏳
- Database connection robustness ⏳



Service operations

- True microservice model
- Reduce components coupling
- Add built-in service protection
- Add built-in service health monitoring

Service Operations

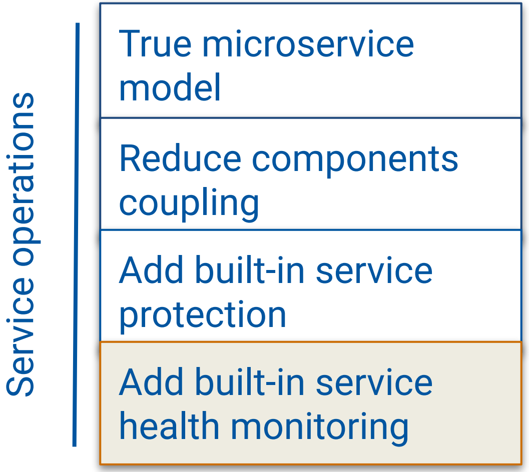
- Service health done indirectly, via logs or database queries
- System cannot report on outcome of last action (scheduling, QoS, etc)

Goal:

- System should be able to provide health indicators
- Allow system to be queried on health indicators

Plan:

- Transform current log-based health metrics into system health state ⏳
- Provide mechanism to query the Transfer & QoS daemons ⏳



Service operations

True microservice model

Reduce components coupling

Add built-in service protection

Add built-in service health monitoring

Community Requests

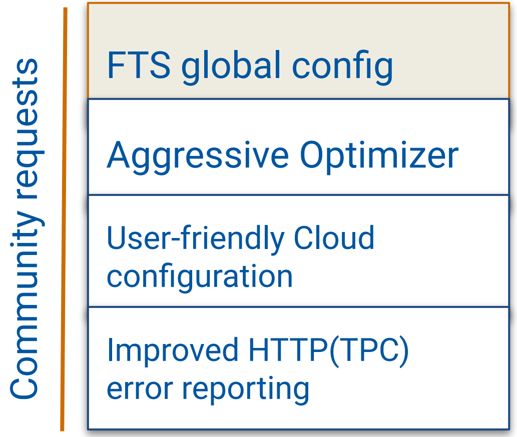
- An FTS instance can orchestrate transfers far from its own location (e.g.: US-based FTS influencing T0 Tape via multihop)
- Site-specific links must be respected diligently

Goal:

- Config publishing server located centrally
- FTS instances can subscribe to receive central config

Plan:

- Must design new central-config publisher ⏳
- Musrt design config subscriber ⏳
- Careful implementation to allow easy manual overriding ⏳


Community requests
FTS global config
Aggressive Optimizer
User-friendly Cloud configuration
Improved HTTP(TPC) error reporting

Community Requests

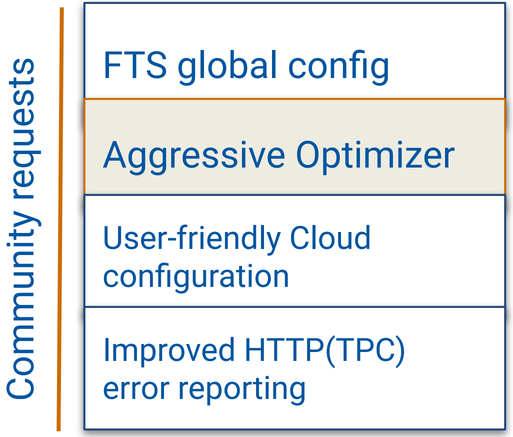
- Optimizer reacts too slowly to abrupt changes
- Optimizer cannot unstuck itself from a bad feedback loop

Goal:

- Optimizer component better models the network throughput and fluctuations
- Optimizer doesn’t require admin intervention to recover

Plan:

- Integrate Optimizer zero-gradient model developed by Richard Yang’s group ⏳


Community requests

FTS global config

Aggressive Optimizer

User-friendly Cloud configuration

Improved HTTP(TPC) error reporting

Community Requests

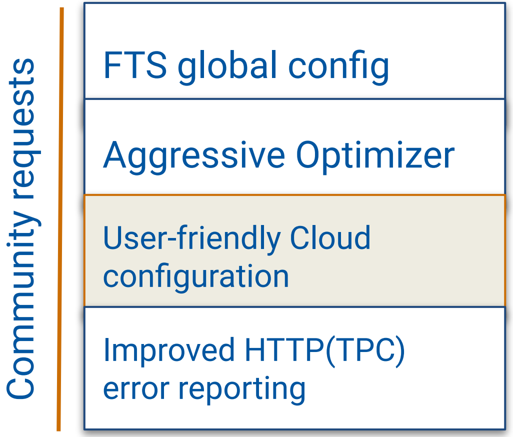
- Cloud transfer configuration is just too confusing
  (see FTS3: Cloud Storage Transfers talk)
- Many support requests on this topic

Goal:

- Cloud transfer configuration becomes intuitive
- Users can do it themselves, without admin assistance

Plan:

- Integrate Eraldo’s Cloud Configuration rework ⏳



Community requests

FTS global config

Aggressive Optimizer

User-friendly Cloud configuration

Improved HTTP(TPC) error reporting

Community Requests

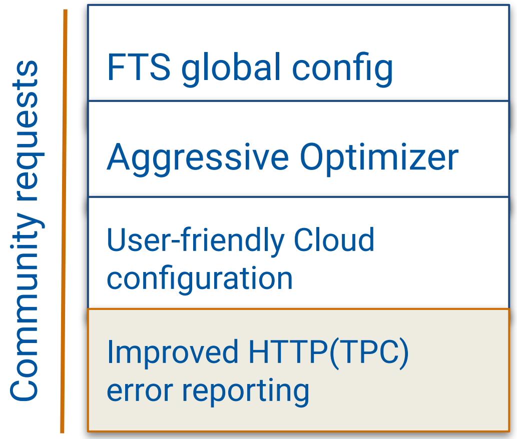
- HTTP-TPC errors are (many times) too obscure
- Great deal of effort spent on this topic (Grid 1st Line Support)

Goal:

- HTTP-TPC errors are explicit, human-readable and easy to understand
- Allow system to be queried on health indicators

Plan:

- Follow approach of documenting and reporting patterns to storage providers (CMS + DOMA-BDT + FTS) ⏳
- Suggestions welcome !!



Community requests

FTS global config

Aggressive Optimizer

User-friendly Cloud configuration

Improved HTTP(TPC) error reporting

Project Evolution

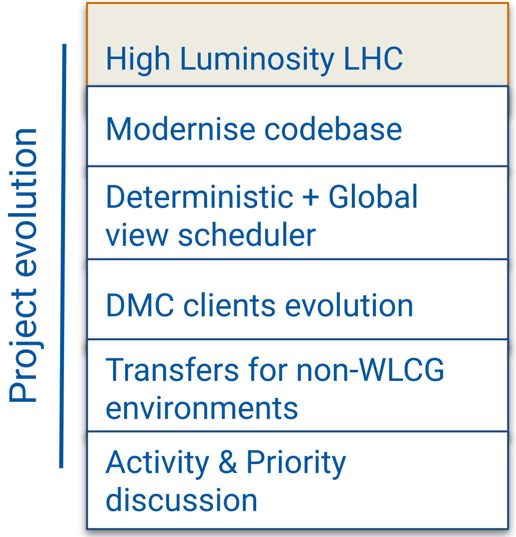
- Scalability objectives for HL-LHC
- System must face 1 → 2 orders of magnitude
- Possibly deal with larger files

Goal:

- Horizontally scalable design, where additional hardware = more gain
- Robust handling of larger file transfers (e.g.: 60+ GB)

Plan:

- Investigate other ways of structuring the data ⏳
- Improve scheduler algorithm ⏳



Project evolution

High Luminosity LHC

Modernise codebase

Deterministic + Global view scheduler

DMC clients evolution

Transfers for non-WLCG environments

Activity & Priority discussion

Project Evolution

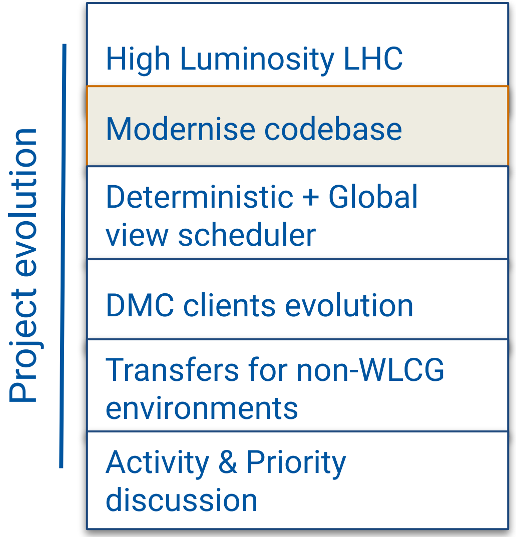
- Code consolidation long needed
- Certain production errors could’ve been avoided with better code scrutiny

Goal:

- Legacy and unused components removed
- Benefit from latest compilers & coding practices

Plan:

- Move compiler to C++17 ✅
- Gradually drop unused components ⏳
- Replace old and risky library dependencies ⏳



Project evolution

High Luminosity LHC

Modernise codebase

Deterministic + Global view scheduler

DMC clients evolution

Transfers for non-WLCG environments

Activity & Priority discussion

Project Evolution

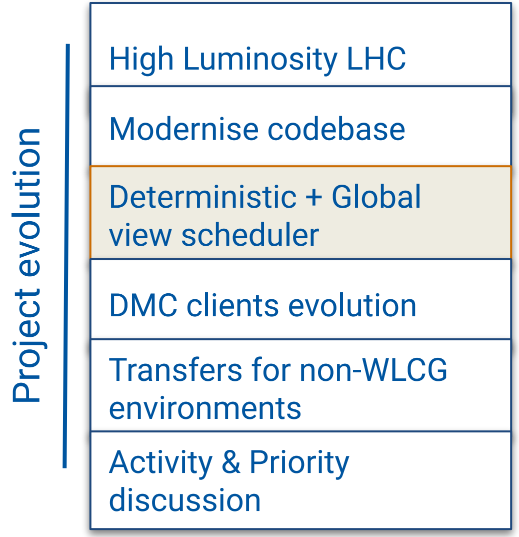
- Current scheduler is stochastic
- Scheduler work partitioning model adds a great deal of complexity

Goal:

- Change scheduler algorithm into something that allows better control
- Have a flexible model that can be extended on other scheduling considerations in the future

Plan:

- Investigate a decide-and-dispatch model, with one scheduler and many worker nodes ⏳
- Introduce statefulness into scheduling ⏳



Project evolution

High Luminosity LHC

Modernise codebase

Deterministic + Global view scheduler

DMC clients evolution

Transfers for non-WLCG environments

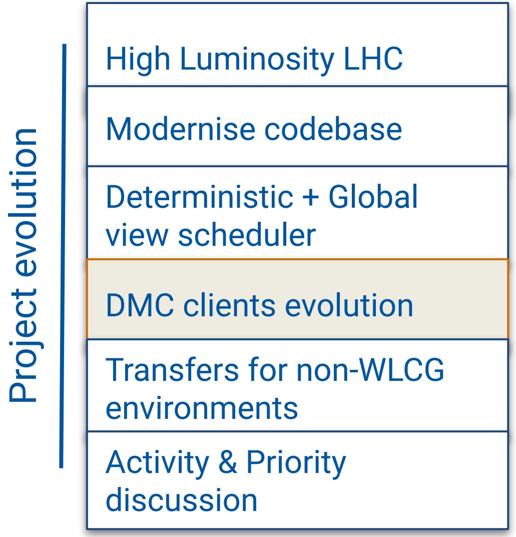
Activity & Priority discussion

Project Evolution

- Hard-to-maintain and risky code hidden in DMC layers

Goal:

- Davix: drop libneon, keep only libcurl backend
- Gfal2: drop GridFTP plugin
- Gfal2: What about SRM?



Project evolution

High Luminosity LHC

Modernise codebase

Deterministic + Global view scheduler

DMC clients evolution

Transfers for non-WLCG environments

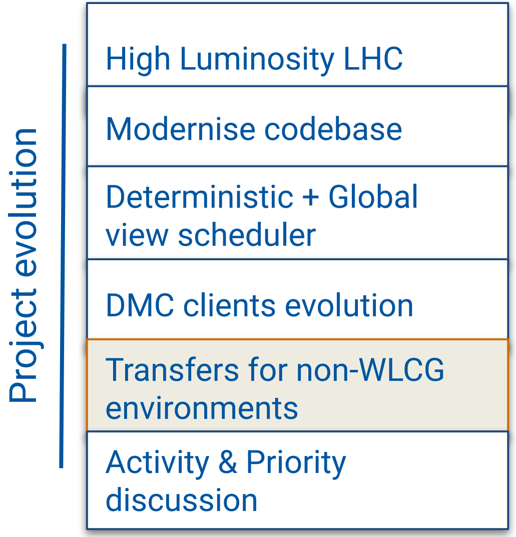
Activity & Priority discussion

Project Evolution

- FTS strives to accommodate also non-WLCG communities

Goal:

- Token implementation not WLCG-bound
- Make FTS appealing to smaller communities as well (those not fortunate enough to run experiment frameworks)



Project evolution

High Luminosity LHC

Modernise codebase

Deterministic + Global view scheduler

DMC clients evolution

Transfers for non-WLCG environments

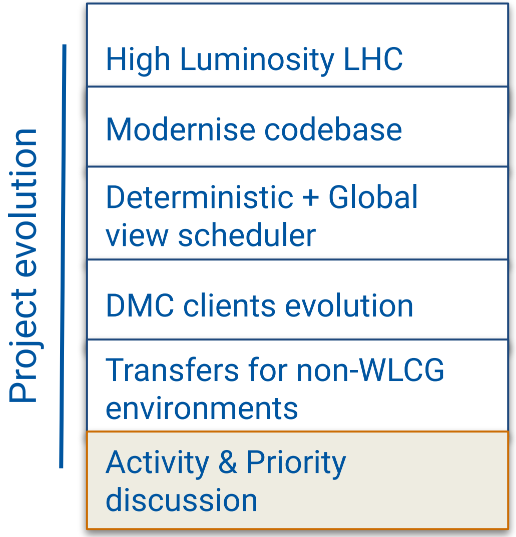
Activity & Priority discussion

Project Evolution

- Discrepancy into what “Activity” and “Priority” means between experiment framework, FTS layer and storage
- Current model only works per-link

Goal:

- Broader prioritization model, which includes not only links but also storage entities
- Make the prioritization definitions clear and FTS-bound



Project evolution
High Luminosity LHC
Modernise codebase
Deterministic + Global view scheduler
DMC clients evolution
Transfers for non-WLCG environments
Activity & Priority discussion

Impromptu Topics

- Tokens?
- Kubernetes?
- Something else?