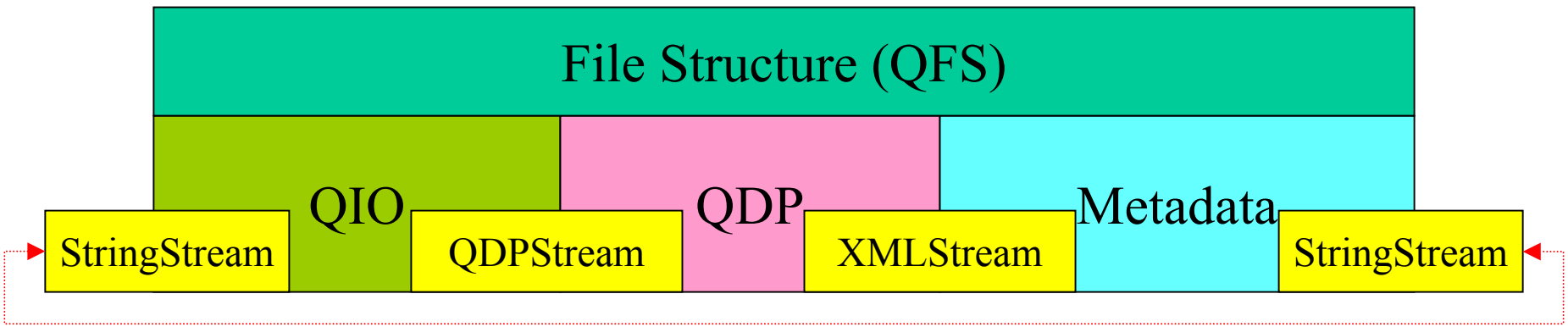
IO Interface

- Metadata
  - name/value pairs
  - direct support for built-in types, streams for generic types
- QDP
  - provides marshal/unmarshal (serialization) routines
  - binary or xml versions of serialization
- QIO
  - single file, parallel file input/output streams
  - DIME support for packaging stream input/output
- QFS - File structure
  - convenience class/routines for packaging metadata and QDP



File Structure (QFS)

```
class QFSInputFile {
MetaReader   reader;  // meta-data container
xmlPath  path;            // possible data file set url
//....
};

// input files tied to a url
QFSInputFile file ( “file://myInputFile” );

// A read brings in the mandated metadata, url, and QDP Object
LatticeReal  QDPfoo;
file.read ( QDPfoo );

// get reader reference or url reference
MetaReader&  reader = file.meta ();
```

XML Example

```
// possible example of xml metadata
<?xml version='1.0' ?>
<objectType typeName="quarkType"  name="quark">
  <stringItem name="description">Wilson-clover-charm-quark</stringItem>
  <doubleItem name="kappa">0.1227</doubleItem>
  <intItem name="color">3</intItem>
  <objectType typeName="sourceType" name="source">
    <stringItem name="name">charmonium-1S</stringItem>
    <doubleItem name="mu">0.350</doubleItem>
  </objectType>
</objectType>
```

Metadata Reader

```
// istreams are used for input
Stringstream xmlstream;

// instantiate reader tied to a stream
MetaReader reader ( xmlstream );

// path is used as a metadata key
xmlPath path ( "/gauge/flavor0/mass" );

// access by path; conversion to compatible type
double qmass = reader [path].as<double> ( );

// move up a level; path will be "/gauge/flavor0"
path.up ( );

// Tree traversal (find) operations provided
```

Metadata Writer

```
// ostreams are used for output
MetaReader reader ( istringstream(my_string) );

// instantiate writer tied to ostream
MetaWriter writer ( ofstream(“file://my_output_file”) );

// copy metadata
writer.import ( reader );

// update number of trajectories
xmlPath path ( "/gauge/trajectory" );
long new_traj = writer [path].as<long> ( ) + 20;

// update metadata
writer [path] = new_traj;

// insert a new object from an xml compliant stream
writer.insert ( “/gauge/complexThing", xmlistream(QDPObject) );
```

DIME

```
// DIME makes multi-record documents
DimeOutput  dg( ofstream( “file://my_file” ) );

// Write metadata
DimeRecord dr1(istringstream(xmlobj), xmlobj.len());
dg.addRecord(dr1);

// Write lattice data
DimeRecord dr2(QDPIStream(fred), fred.len());
dg.addRecord(dr2);

dg.close();

// Form of output
------------- Record mark  -------------
record 1
------------- Record mark  -------------
record 2
------------- End record mark ------------
```