FERARI

Flexible Event pRocessing for big dAta aRchItectures

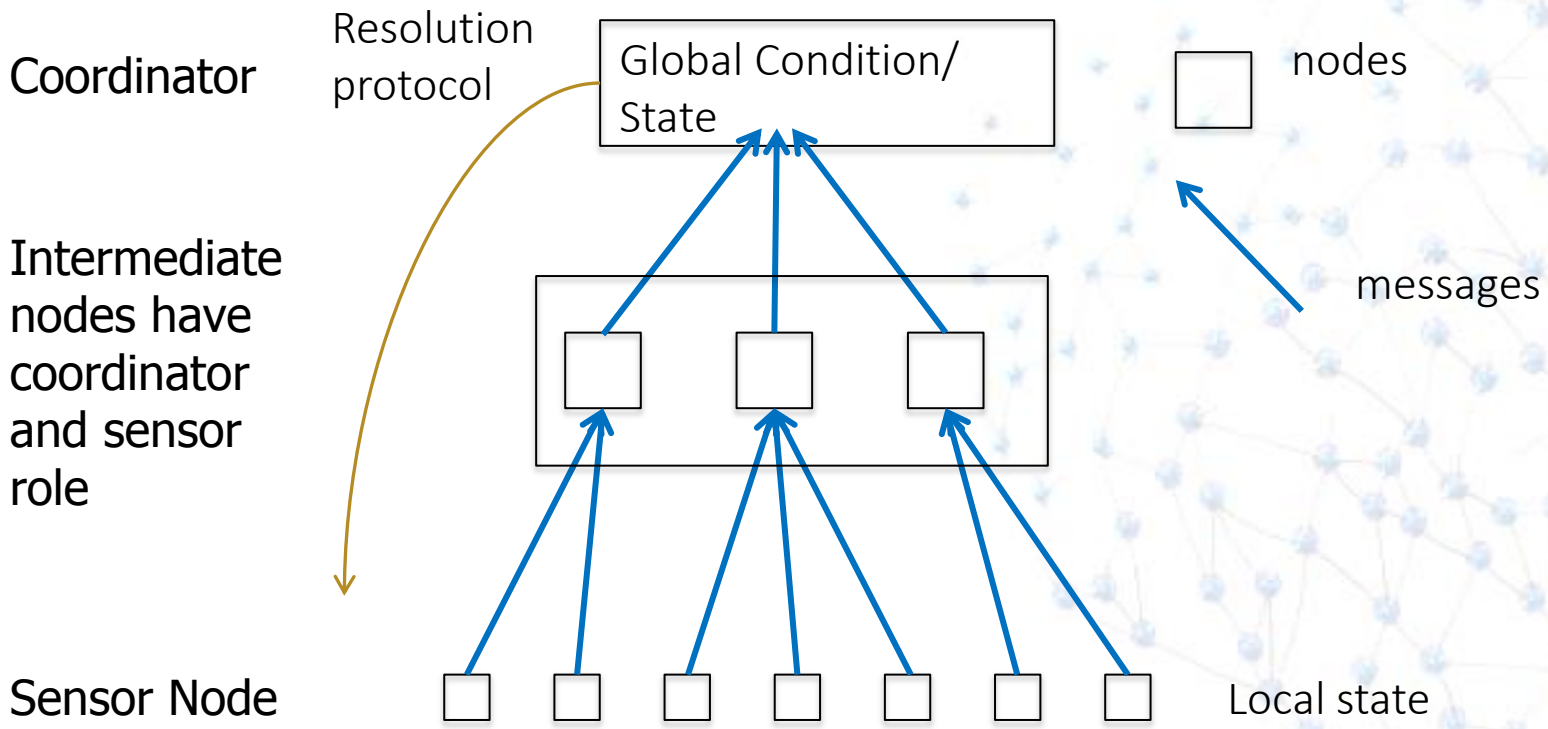
WP2

Big Data Streaming Architecture & Technology Integration

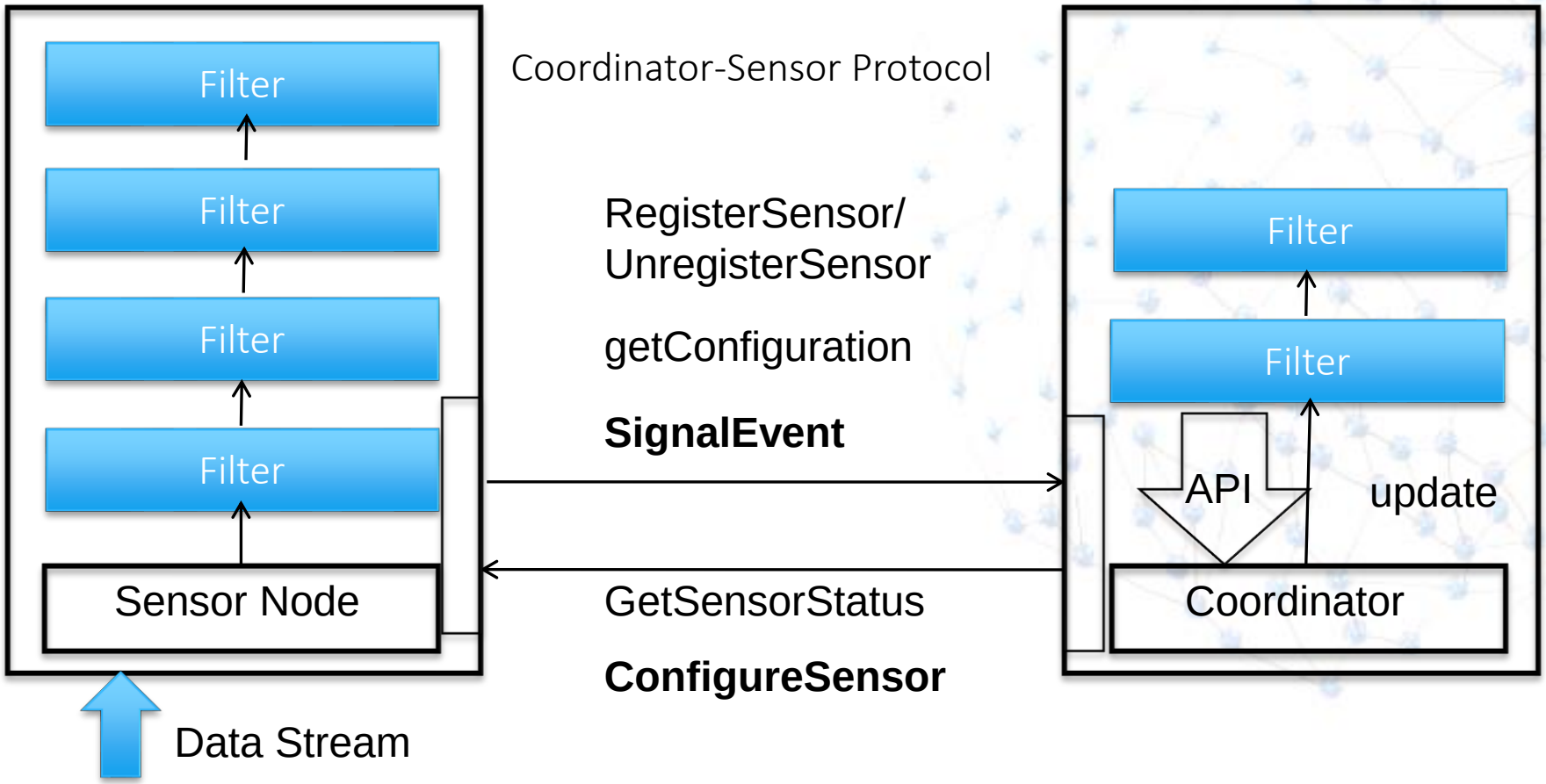
Outline

- Lift Reference Architecture
- Storm Example
- FERARI 1st Architecture Design Proposal
- Missing Pieces
  - Group Communication
  - Dynamic Topology handling
  - Connection to CEP
- Next steps

LIFT Reference Architecture



LIFT Filter Model
Data Stream driven



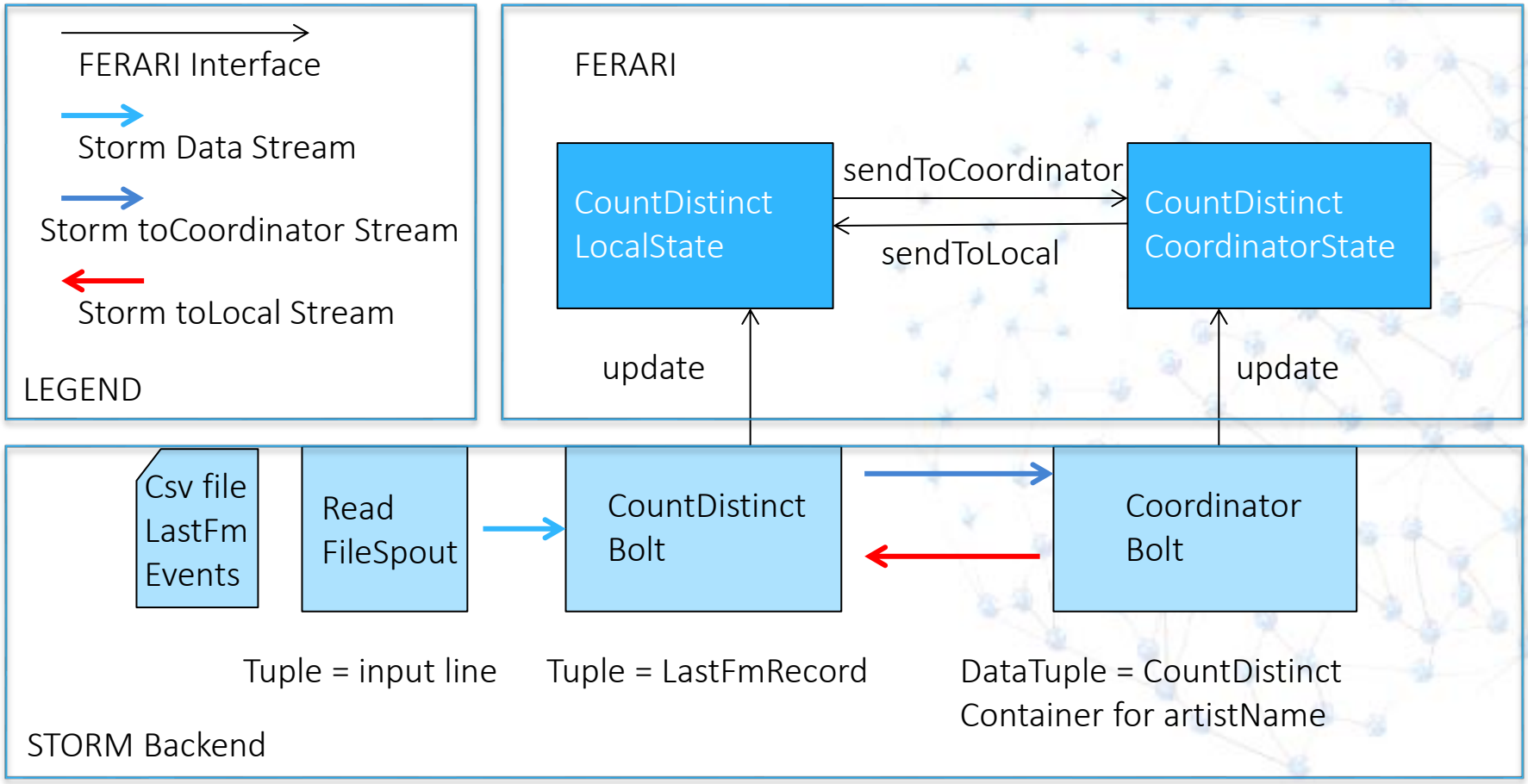
FERARI

Application Example| Count Distinct

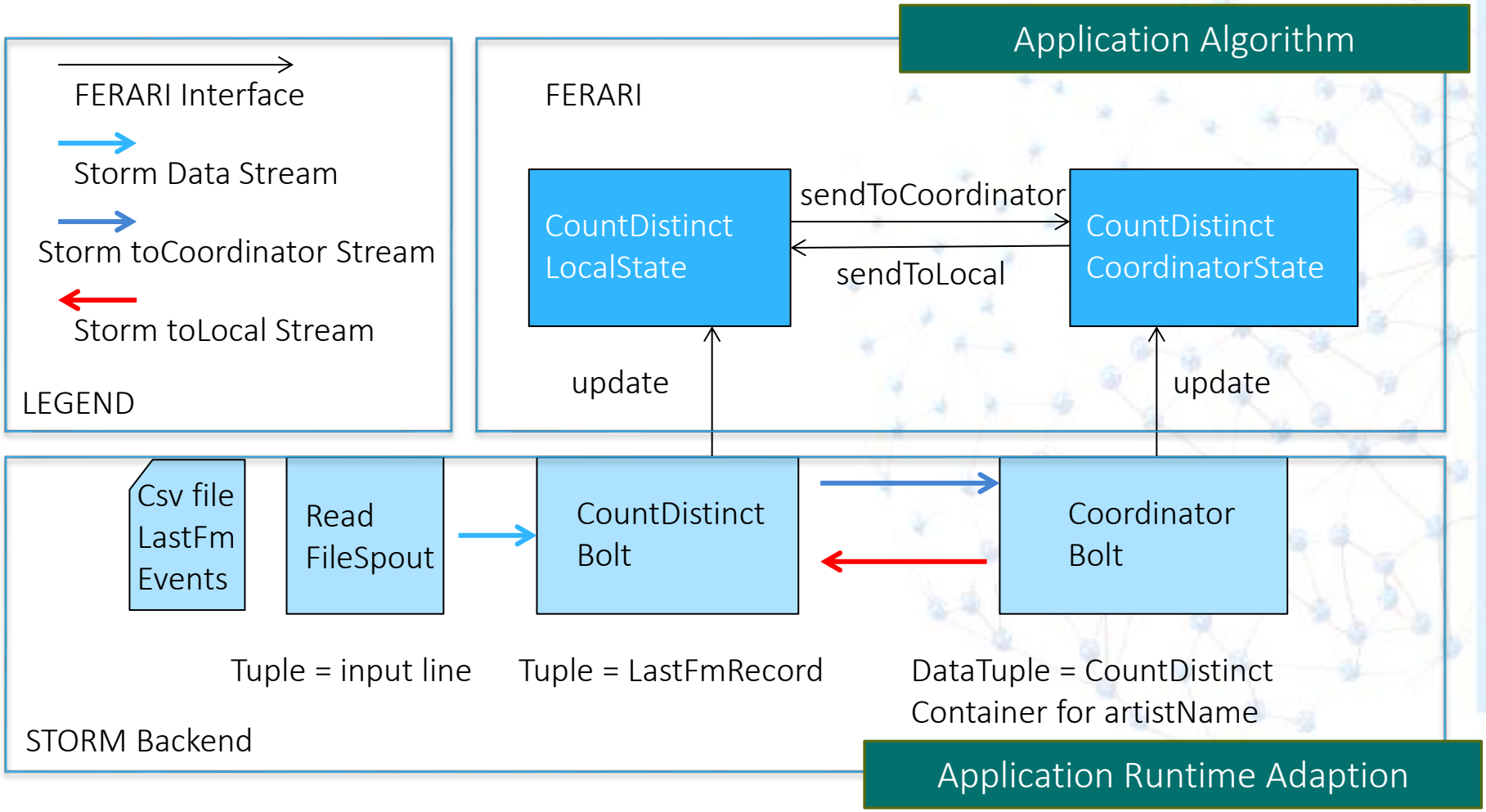
* Last.fm dataset
  * Each record represent a user listening to an artist
* Application:
  * Count the number of distinct users listening to an artist in the stream of listening events
  * Restart counting when global condition is met, i.e., >50 Users
* Algorithm: LinearSketch
* Platform adaption: Storm Topology

Fraunhofer IAIS

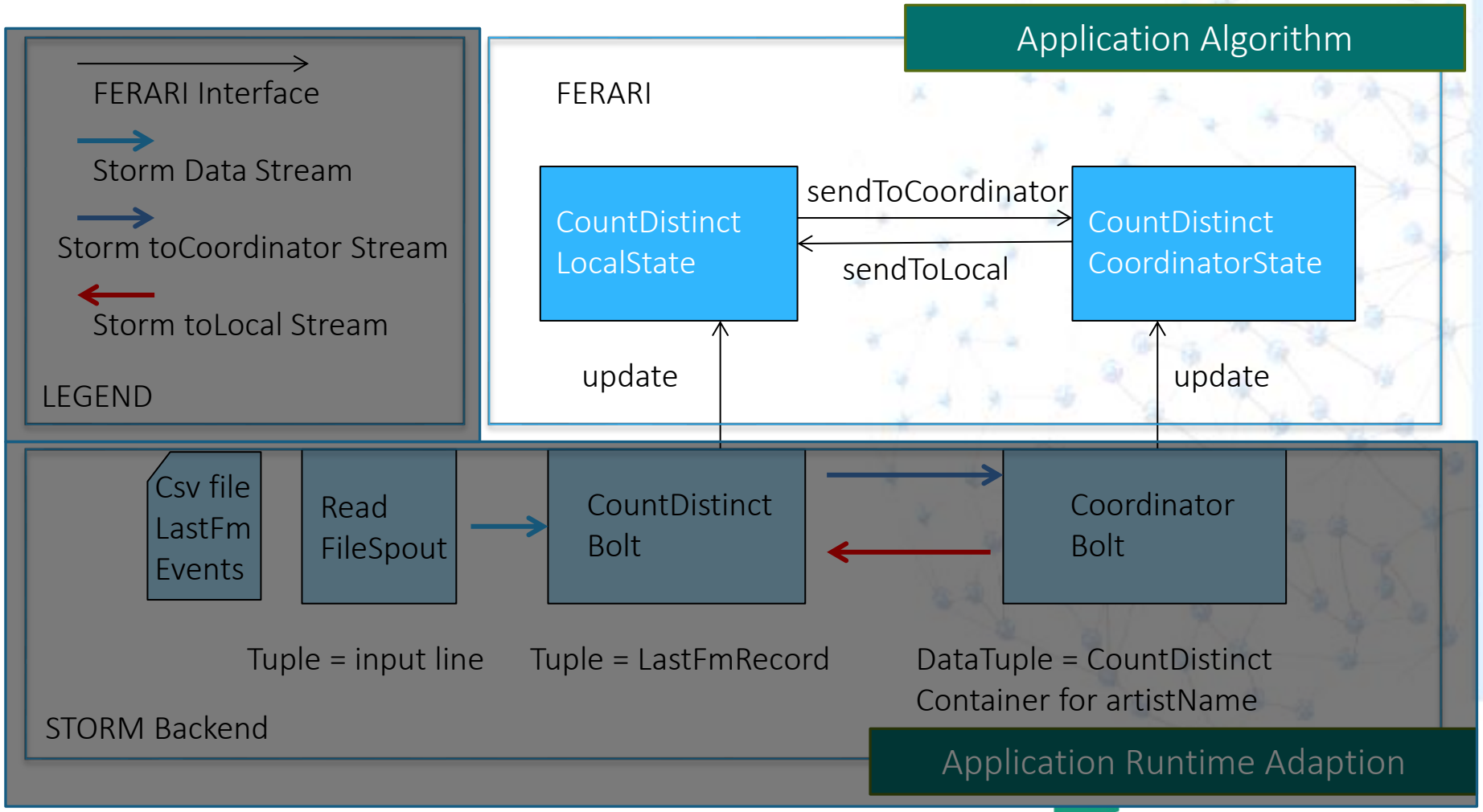
Overview | Count Distinct



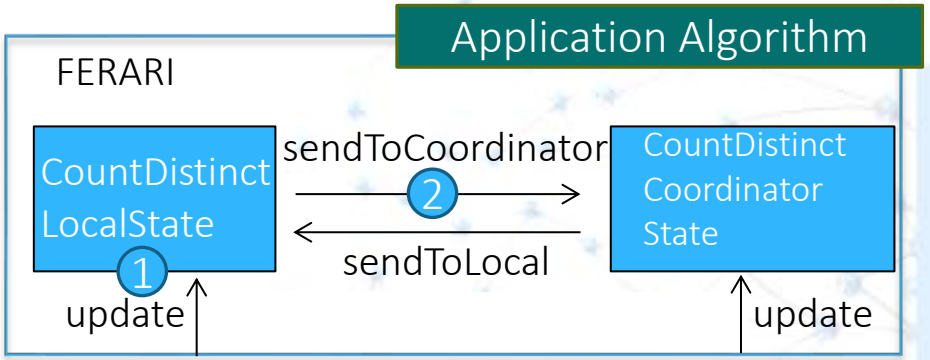
Overview | Count Distinct



Overview | Count Distinct



CountDistinct Local State




```
1 @Override
  public void update(DataTuple input) {

      LastFmDataRecord record = (LastFmDataRecord) input.getValue(0);
      String keyForCounter = record.getArtname();

      ---------- snip ----------

      CountDistinctContainer counter = counts.get(keyForCounter);

      if (counter == null) {
          counter = new CountDistinctContainer(keyForCounter, this.sinceWhenDate);
          counts.put(keyForCounter, counter);
      }

      counter.add(recordTime, userId);

      // send update to global model if count distinct was increased
      if (counter.isIncreased()) {
          DataTuple data = new DataTuple(counter);
          if (sendToCoordinator != null) {
2             sendToCoordinator.signal(data);
          }
      }
```

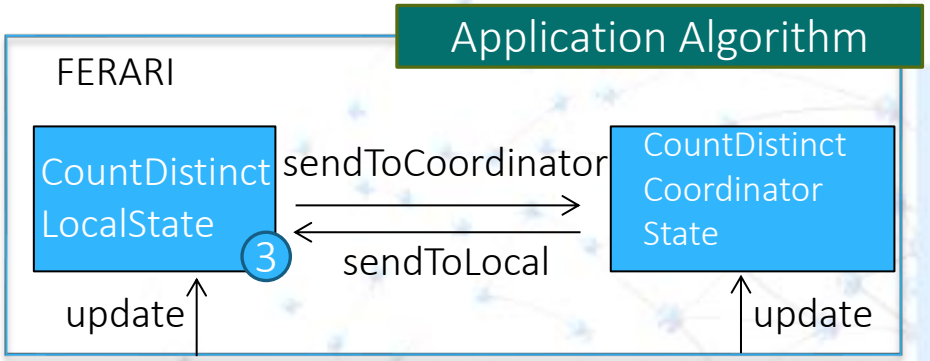
Linear sketch

Local Condition

Send Violation

Fraunhofer IAIS

CountDistinct Local State

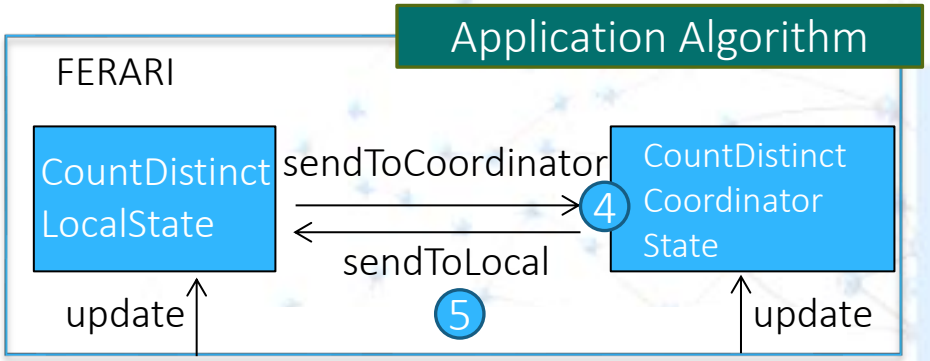



```
3  @Override
   public void handeFromCoordinator(DataTuple data) {
       // reset state and set new reference time
       Date sinceWhenDate = (Date) data.getValue(0);
       this.sinceWhenDate = sinceWhenDate;
       counts.clear();
```

New reference

Fraunhofer IAIS

CountDistinct Coordinator State



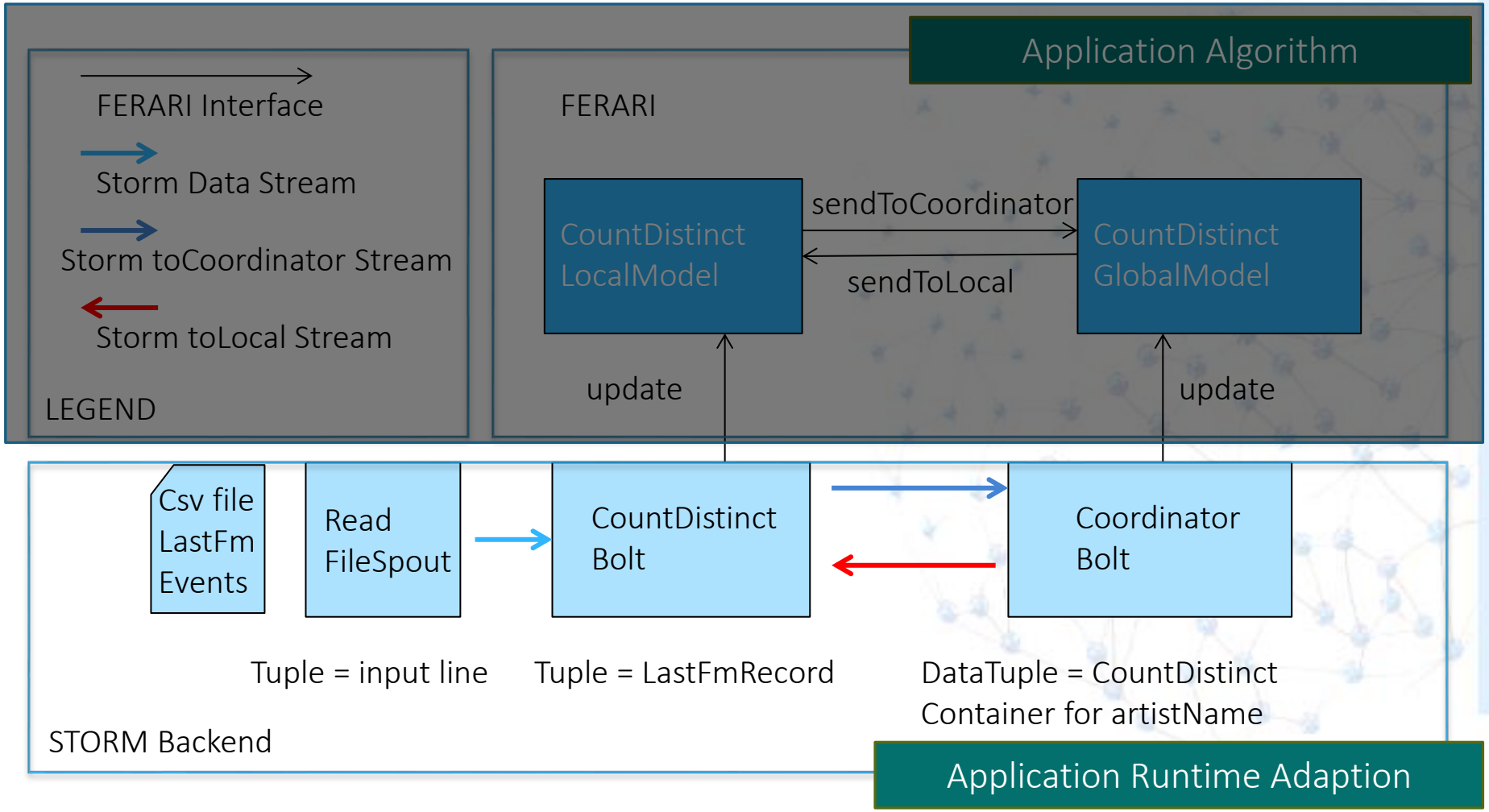

```
4 @Override
public void update(DataTuple input) {
    CountDistinctContainer countDistinct = (CountDistinctContainer) input.getValue(0);
                                snip
 // notifiy if number of users is increased
 if (isnew || globalCounter.isIncreased()) {

     CountDistinctContainer allUsers = counts.get(LastFmConstants.USER_COUNTDISTINCT);
     long allUsersCount = 1;
     if (allUsers != null) {
         allUsersCount = allUsers.getCount();
     }
     count = globalCounter.getCount();

     if (globalCounter.getKey().equals(LastFmConstants.USER_COUNTDISTINCT)
             && globalCounter.getCount() >= LastFmConstants.MAX_REACH) {
         sinceWhenDate = globalCounter.getLastUpdate();
         counts.clear();

     5 sendToLocal.signal(new DataTuple(sinceWhenDate));

     }
}
```

Monitor global condition

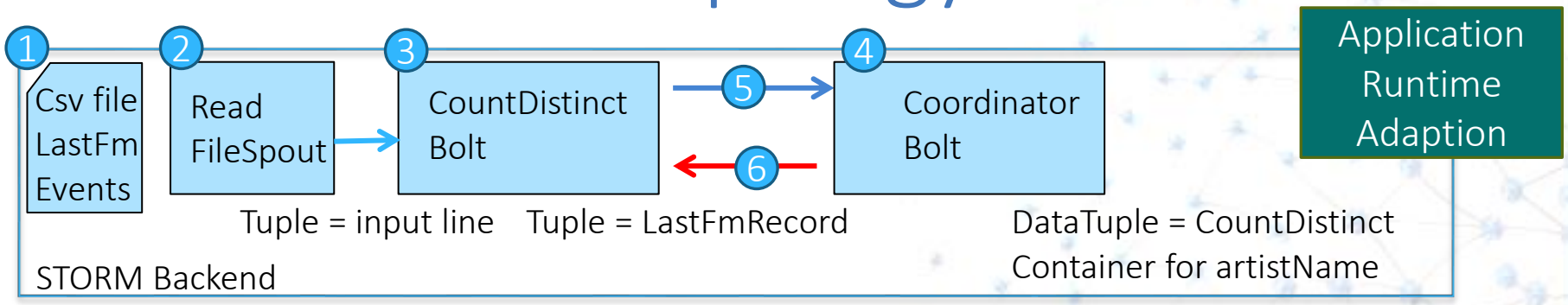
Send new reference

Fraunhofer IAIS

Overview | Count Distinct



CountDistinct Topology




```
Config conf = new Config();
conf.setDebug(false);
conf.put(LastFmConstants.INFILE, LastFmConstants.INFILENAME);

// Build the topology
TopologyBuilder builder = new TopologyBuilder();

builder.setSpout(SPOUT_ONE_ID, new FileRecordReaderSpout(), 1);

builder.setBolt("preprocBolt", new PreprocBolt(), 1).shuffleGrouping(SPOUT_ONE_ID);

BoltDeclarer countDistinctBolt =
    builder.setBolt("countDistinctBolt", new CountDistinctBolt(), LastFmConstants.NUM_PARRALELL_OPS);

countDistinctBolt.shuffleGrouping("preprocBolt"); //better load distribution,

BoltDeclarer coordinatorBolt = builder.setBolt("coordinatorBolt", new CoordinatorBolt());

coordinatorBolt.globalGrouping("countDistinctBolt", StormSendToCoordinator.TO_COORDINATOR_STREAM_NAME);

countDistinctBolt.allGrouping("coordinatorBolt", StormSendToLocal.TO_LOCAL_STREAM_NAME);
```