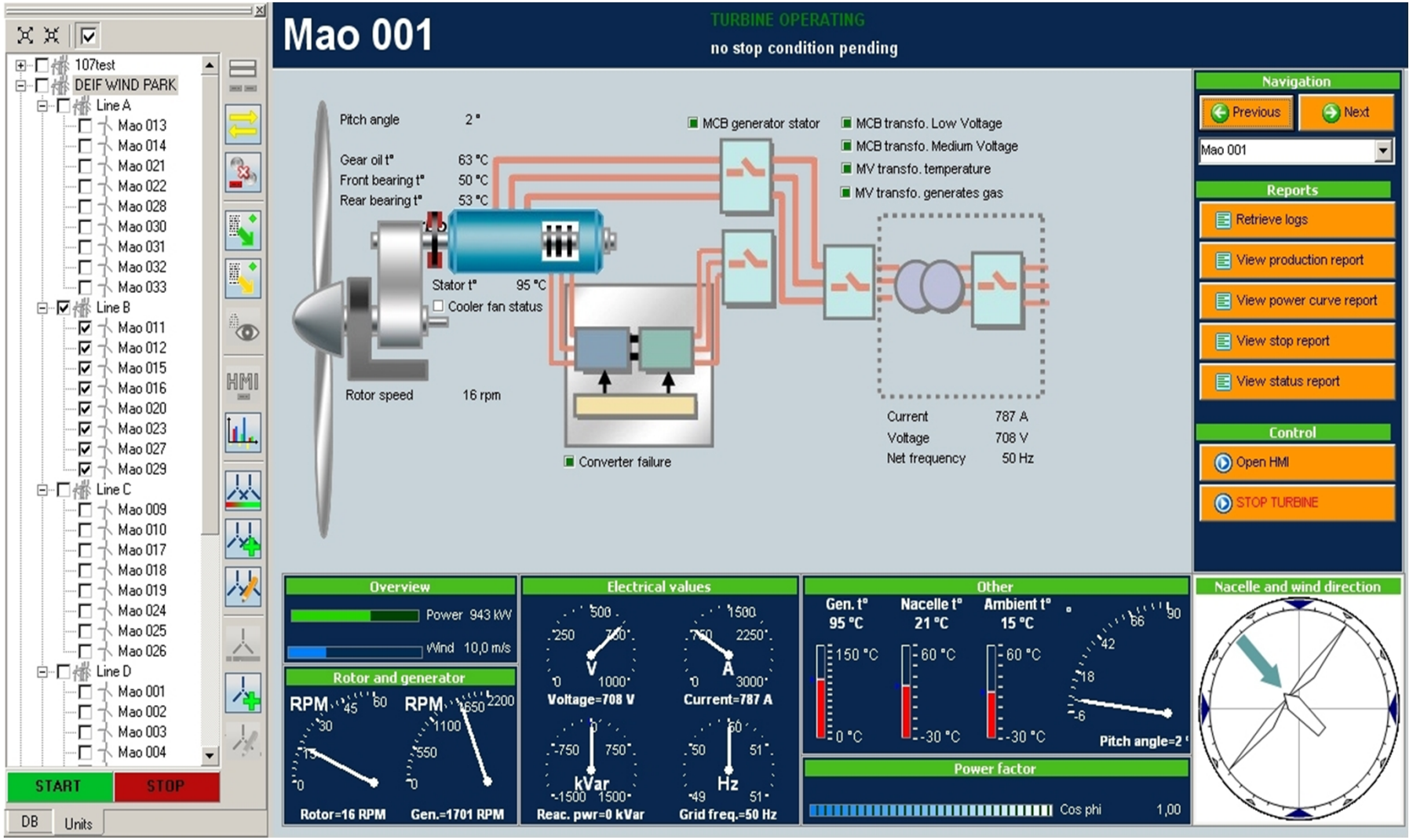
Improving the resilience of SCADA in Critical Infrastructures

André Nogueira, Alysson Bessani, ***Nuno Neves***

LASIGE,
Faculdade de Ciências da Universidade de Lisboa
June 2016

SCADA: Example Industrial Facilities

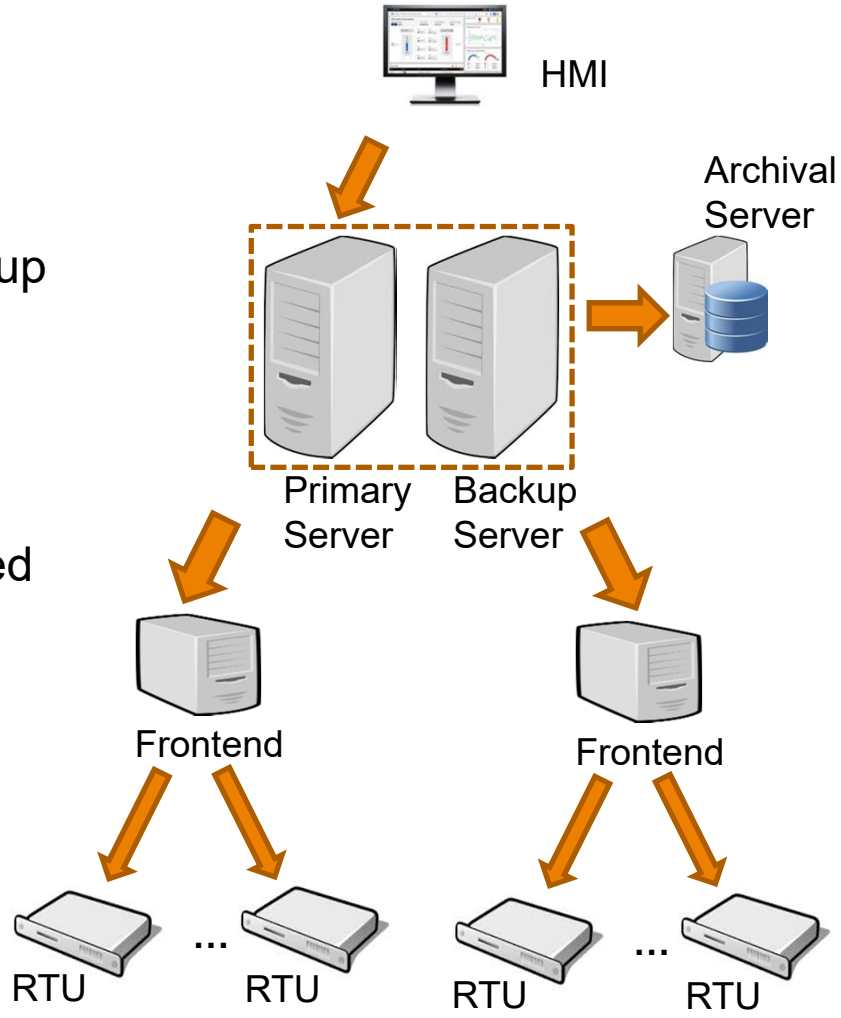
SCADA: Example of an HMI



SCADA: Controller Tolerant to Crashes

- Redundancy method with two identical replicas to tolerate crashes
- Upon failure of the primary server, the backup replica takes over, replacing it
- The backup replica can operate as a hot/warm/cold standby
- The backup replica needs to be kept updated

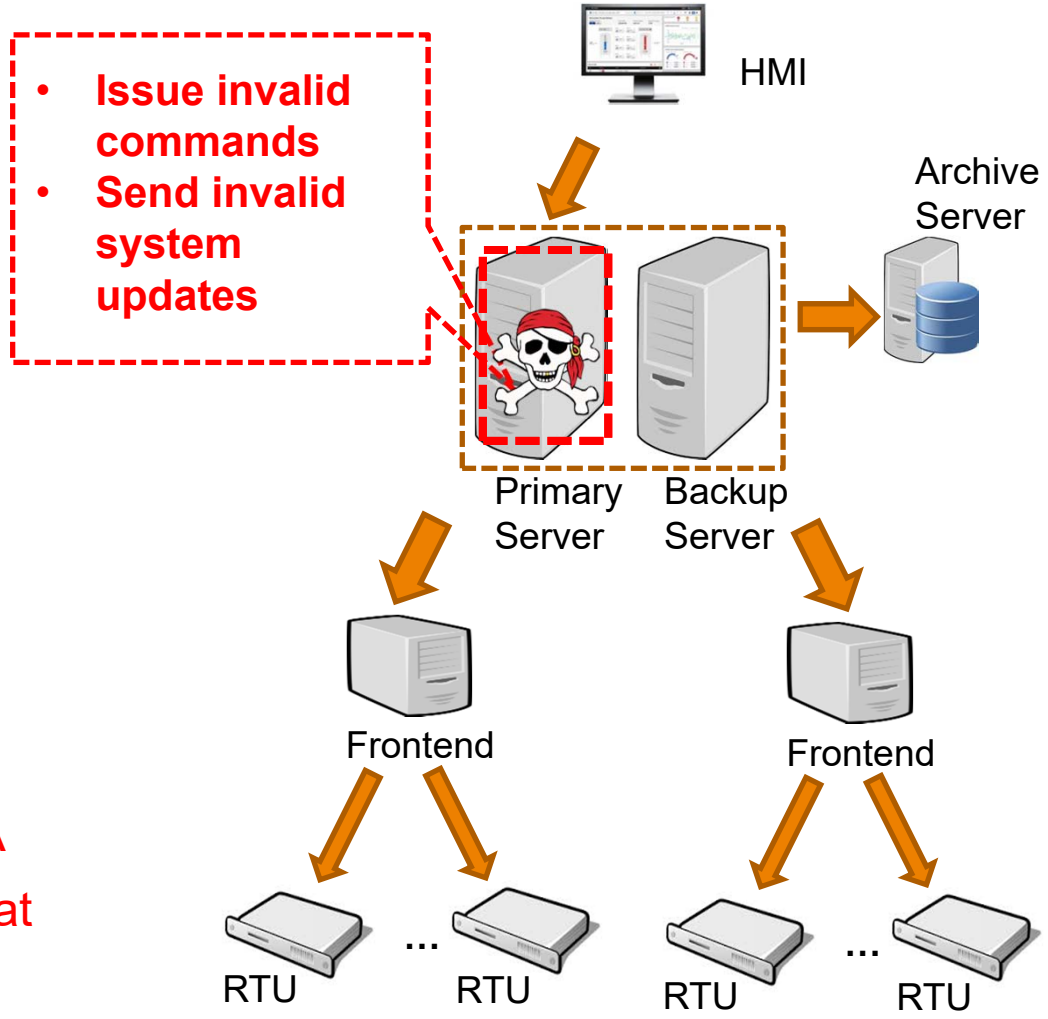
- Tolerates one crash fault
- Relatively short downtime period



Attack Scenario

- Reported advanced cyber attacks continues to rise
  - ➢this is particularly visible in the electrical area with many described incidents
- Add intrusion prevention using standard IT security technologies
  - ➢Firewalls
  - ➢IDS

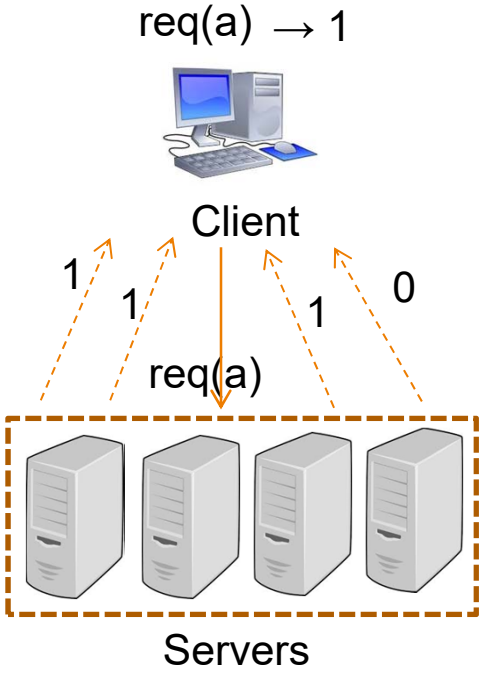
- Given the criticality of the SCADA system it is prudent to assume that prevention is not perfect !



How to make a system intrusion tolerant?

Main ingredients:
- High level of mistrust in all components
- Assume a bound on the number of replicas compromises

- Use State Machine Replication (or Active Replication) to tolerate arbitrary/Byzantine faulty behavior
  - ➢ Weakest possible failure assumption
  - ➢ n= 3f+1 (f=1, n=4)



1. Every client request is processed by a group of servers
2. Servers must execute the same sequence of requests
3. The client infers the correct result of a request from the majority of the answers

- Servers coordinate to decide the order of request processing

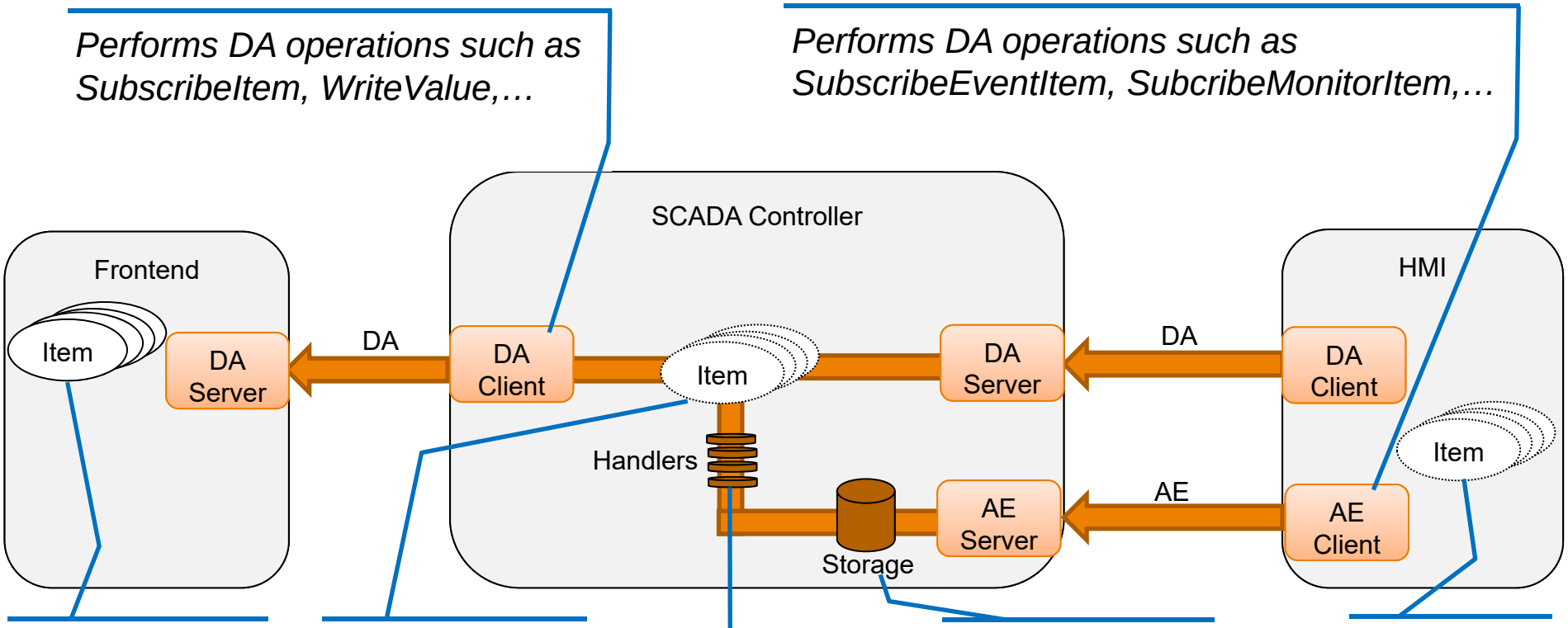
Putting the Idea into Practice!

- Develop an intrusion tolerant SCADA controller
- Integrate an active replication library with an open source SCADA
- Minimize the necessary modifications on the SCADA system
- Achieve a performance level to be used in the field

Some of the challenges:

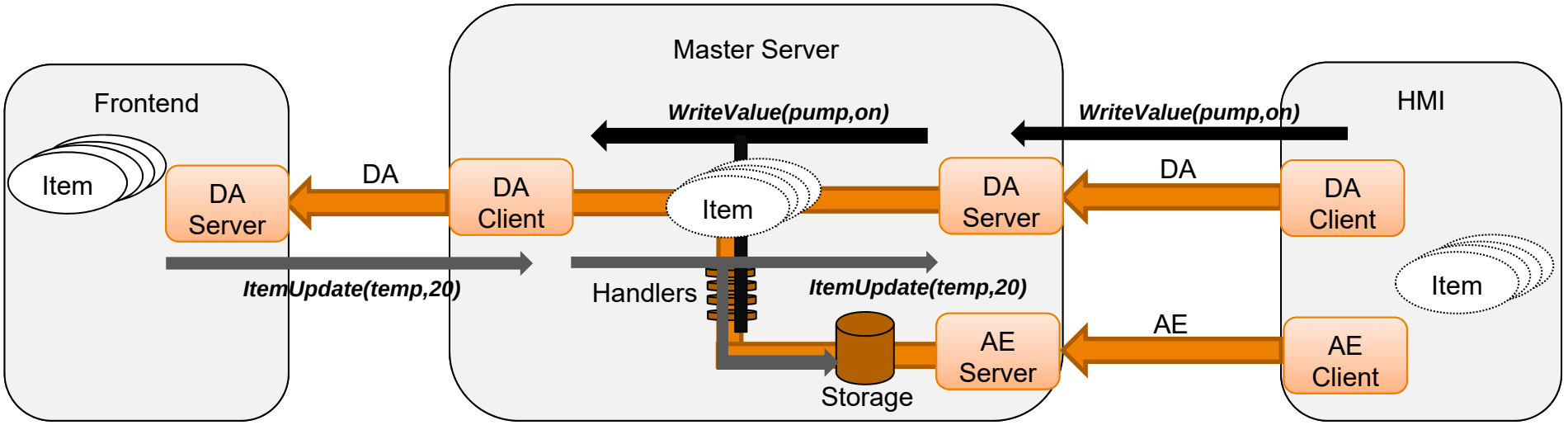
- Find a suitable open source software → **EclipseSCADA**
- Project size (more than 500 sub-projects → **900.000 LOC**)
- Poor documentation
  - source code
  - use case examples
- EclipseSCADA is a framework and not a ready-to-use solution

EclipseSCADA: Components in detail

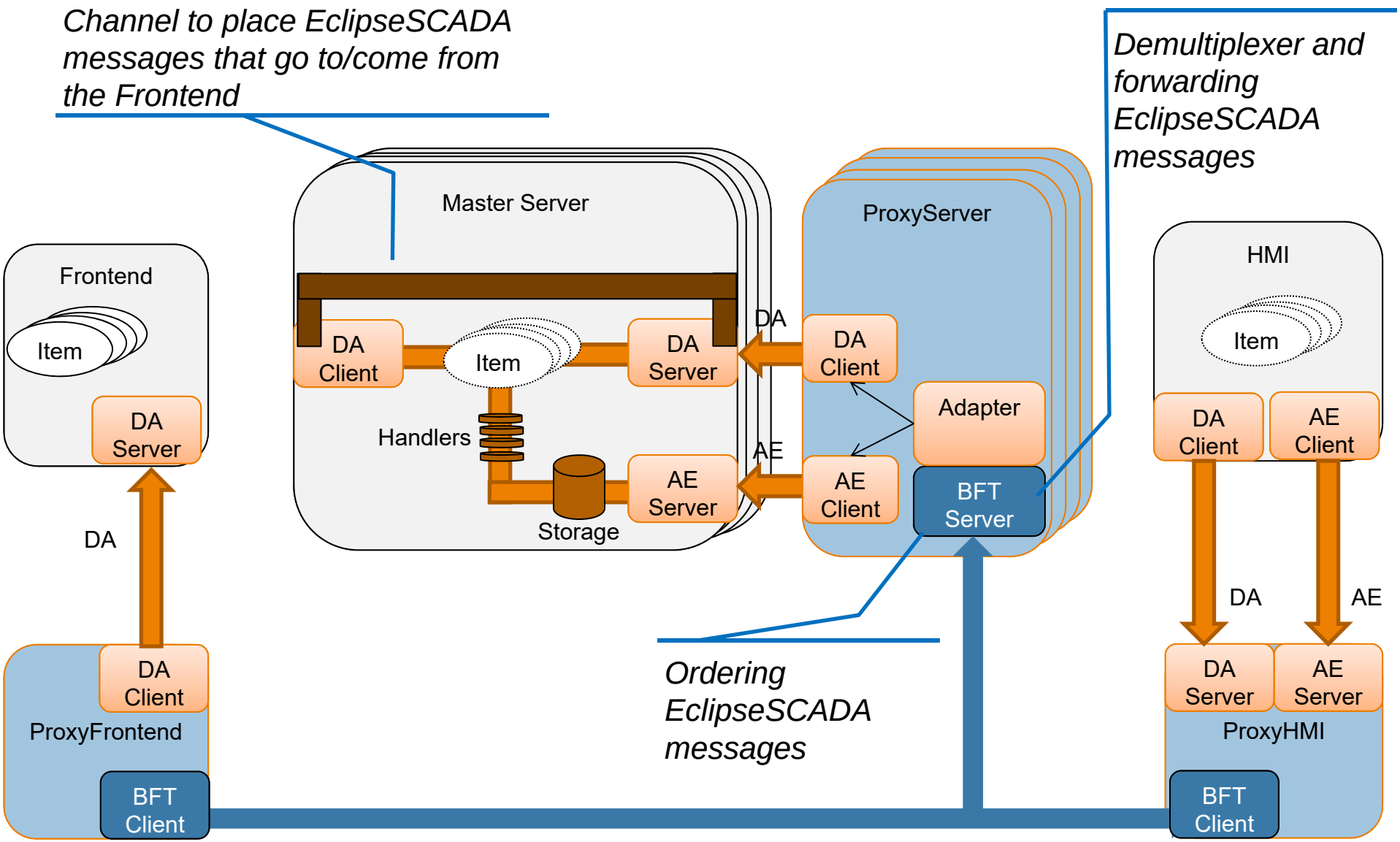


Challenges for replication

- Multiple I/0 channels
- Messages arrive at any moment and do not have bounded delivery times
- Multiple sources of unpredictability that cause a violation of deterministic replica execution
  - ➢ asynchronous execution
  - ➢ timestamps



Resilient EclipseSCADA



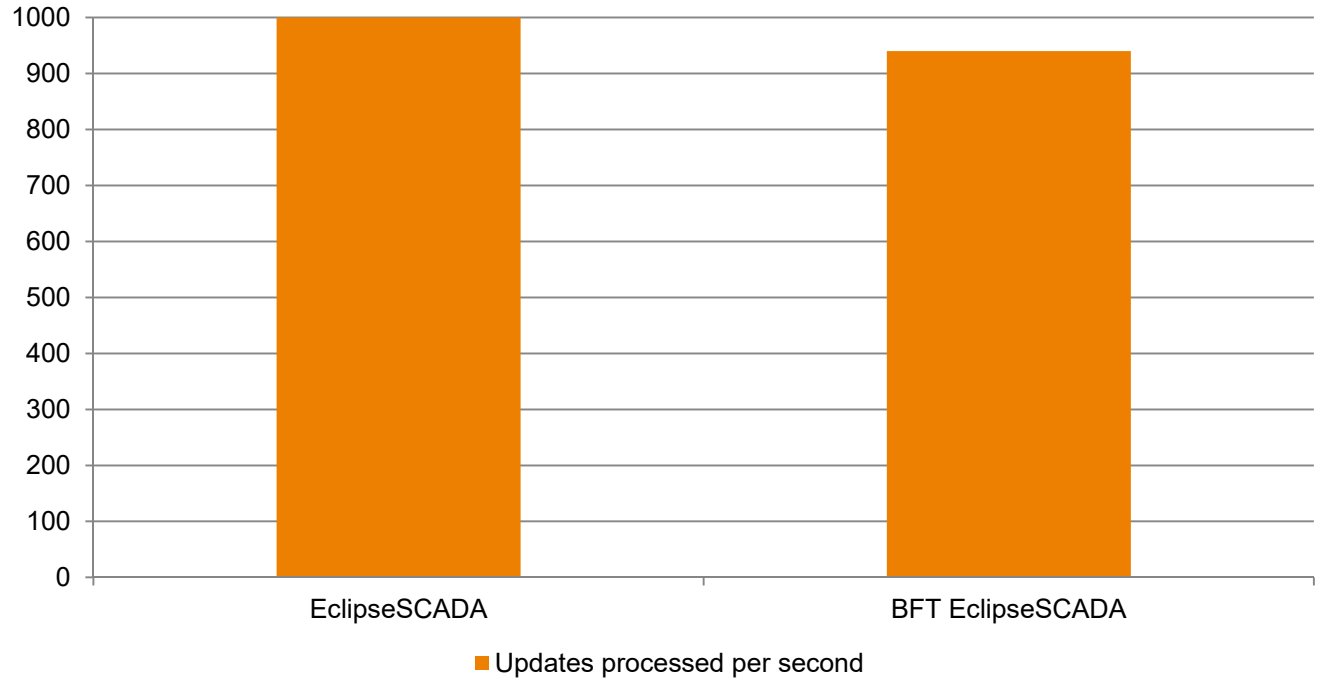
Evaluation

- Measure the overhead introduced by using active replication.
  - ➢compare the performance of the EclipseSCADA and the BFT EclipseSCADA

- Hardware setup infrastructure
  - ➢1 Frontend
  - ➢4 EclipseSCADA controller
  - ➢1 HMI

Item update test

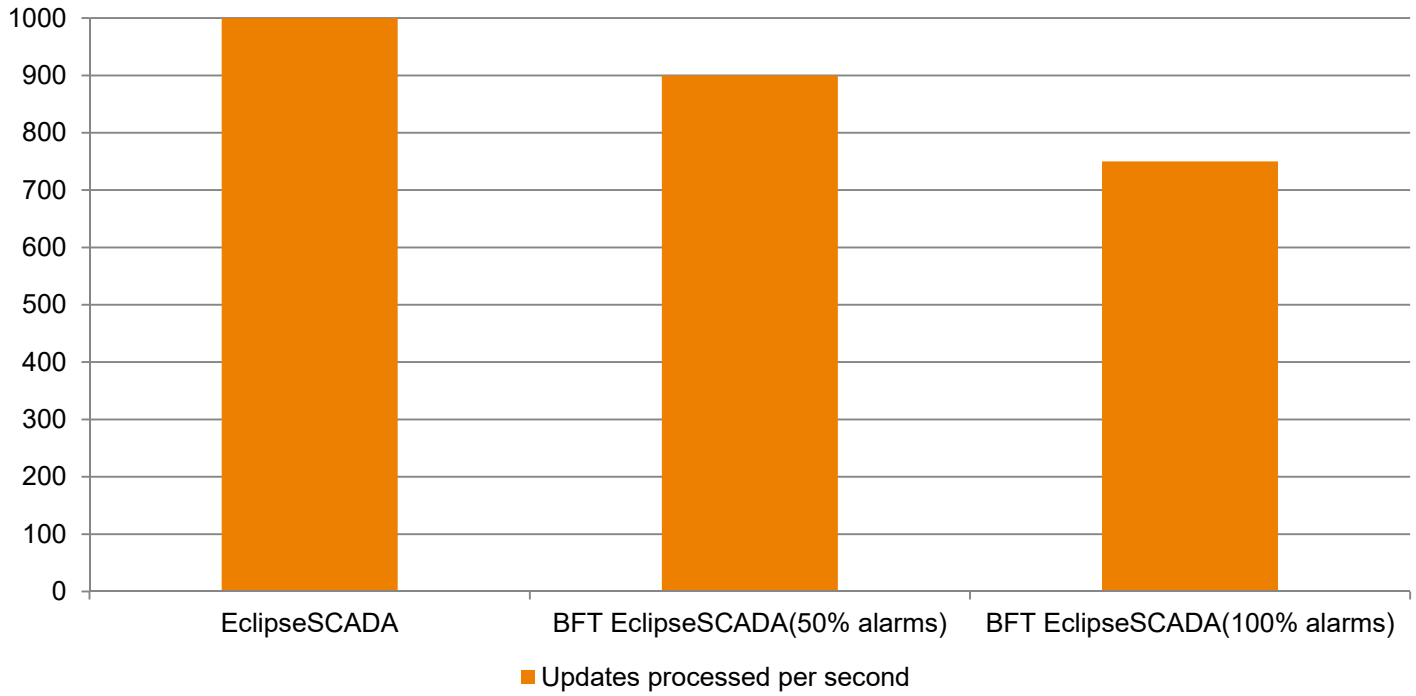
- Benchmark
  - Update10 items in the frontend
  - Each item updates itself 100 times per sec



- Overhead of 6% mainly explained due
  - 2 steps in EclipseSCADA, 6 steps in BFT EclipseSCADA

Item updates with alarms test

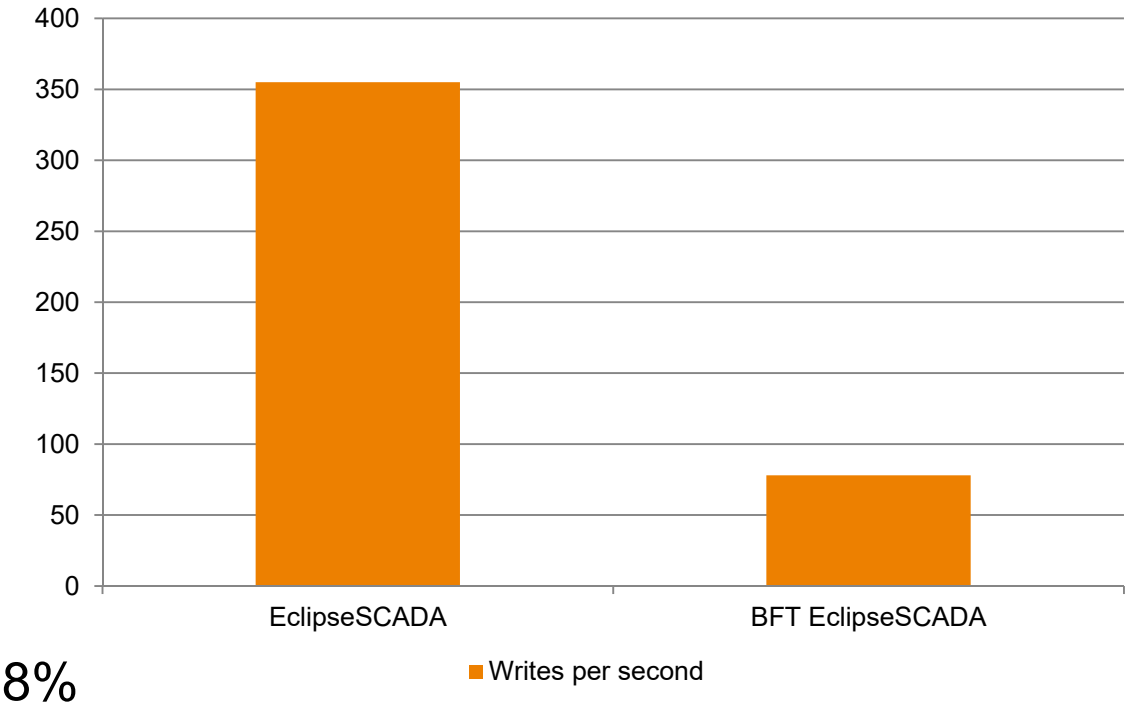
- Benchmark
  - Update 10 items in the frontend
  - Each item updates itself 100 times per sec
  - Add monitor handlers



- Overhead of 25% in worst case scenario (100% alarms)

Item writing test

- Benchmark
  - Write 1 item in the frontend



- Overhead of 78%

➢ 4 steps in EclipseSCADA, 12 steps in BFT EclipseSCADA

➢ The active replication library introduces an overhead of only 12%

Conclusions

- Described the risk of not having an intrusion-tolerant SCADA controller
- Presented a modular solution to make the operation of a SCADA Controller more resilient
- Explained how to integrate an intrusion-tolerant replication library with a open source SCADA system
- Presented some preliminary evaluation results