



---

# Tango Database & MySQL Cluster

Claudio Scafuri [claudio.scafuri@elettra.trieste.it](mailto:claudio.scafuri@elettra.trieste.it)



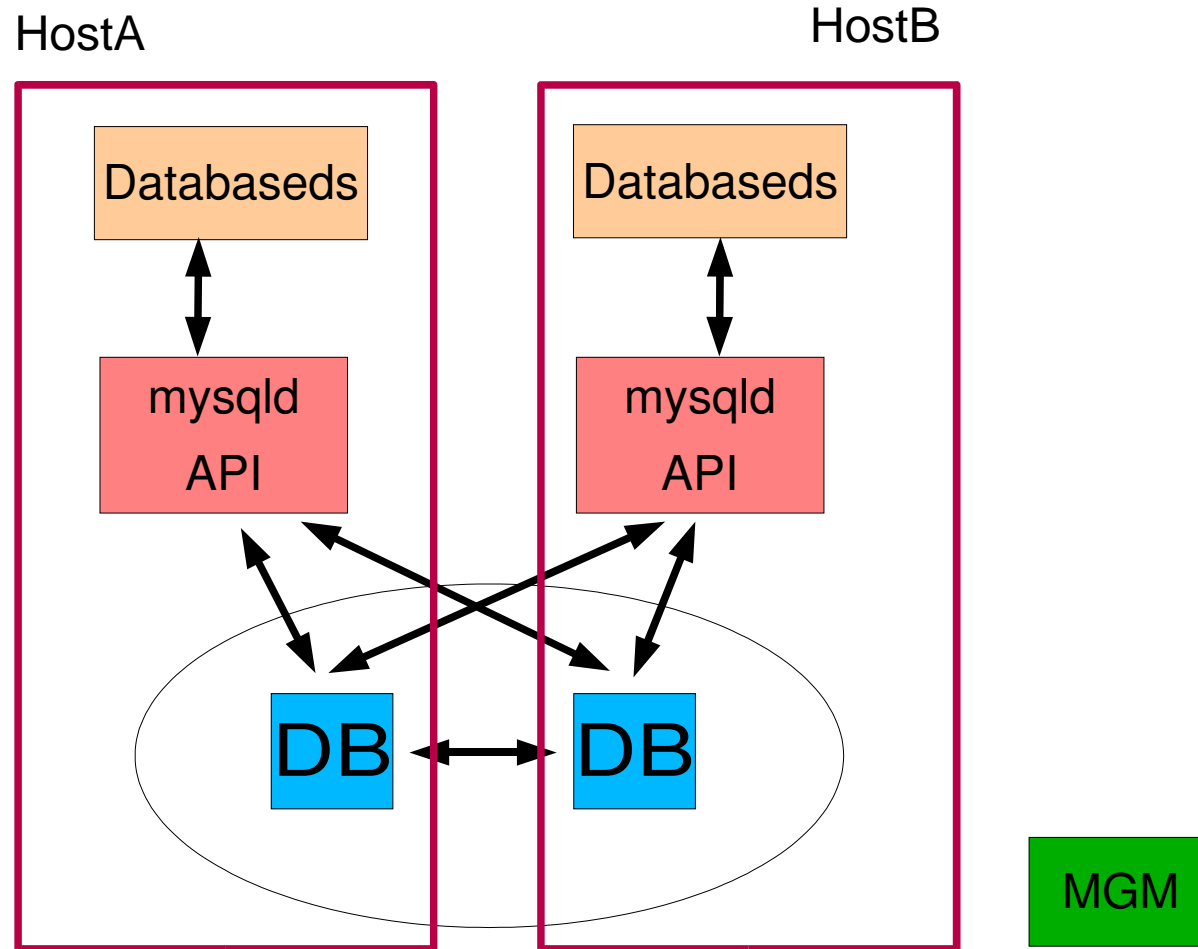
---

# Topics

- news & experiences
- considerations
- presentation of “deliverables”



# Where we left ....





# News

- MySQL max 4.1.12 . Clustering is “stable”
- improved documentation
  - but refers to MySQL 5 ! They are pushing in that direction
  - still some gray area ...
- forums & mailing list by MySQL with precious informations



---

## “Crashes” explained

Conservative design:

save DB from corruption and inconsistencies

Not crashes but *shut-down* of storage nodes!



## “Crashes” explained

- lack of memory
- disk space
- network connectivity

*Tests were carried on “low end” PCs !*



## “Crashes” explained

- lack of memory:
  - 98 devices
  - 895 properties
  - 176 attribute properties
- `ndbd` : 427 MB Virtual / 27 MB Resident



## “Crashes” explained

- disk space:
  - 98 devices
  - 895 properties
  - 176 attribute properties
    - 314 MB on Master node
    - 236 MB on Slave node
  - plus logs...



## “Crashes” explained

- network connectivity:
  - must avoid “split brain” syndrome: 2 different commits on the same rows/tables.*
  - majority of storage node must “see” each other
  - or*
  - storage nodes can see an “arbitrator”:
    - MGM node
    - myqld node



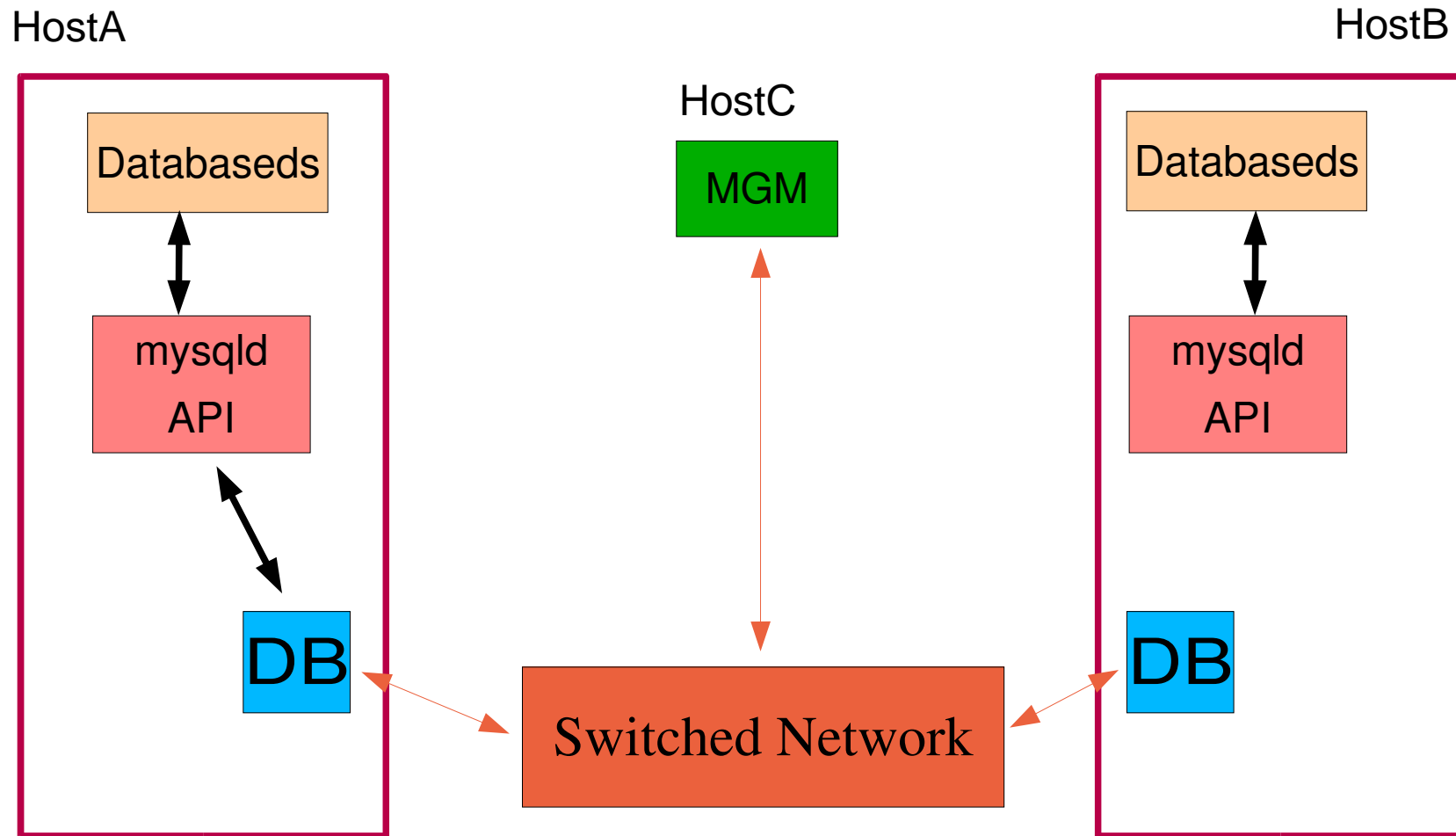
# Cluster Configuration

- MySQL strongly recommends to have a single management node in the cluster. It works as arbiter.

In principle it is possible to have multiple management nodes by assigning an arbitration rank. But : “*the configuration has not been completely tested.*”

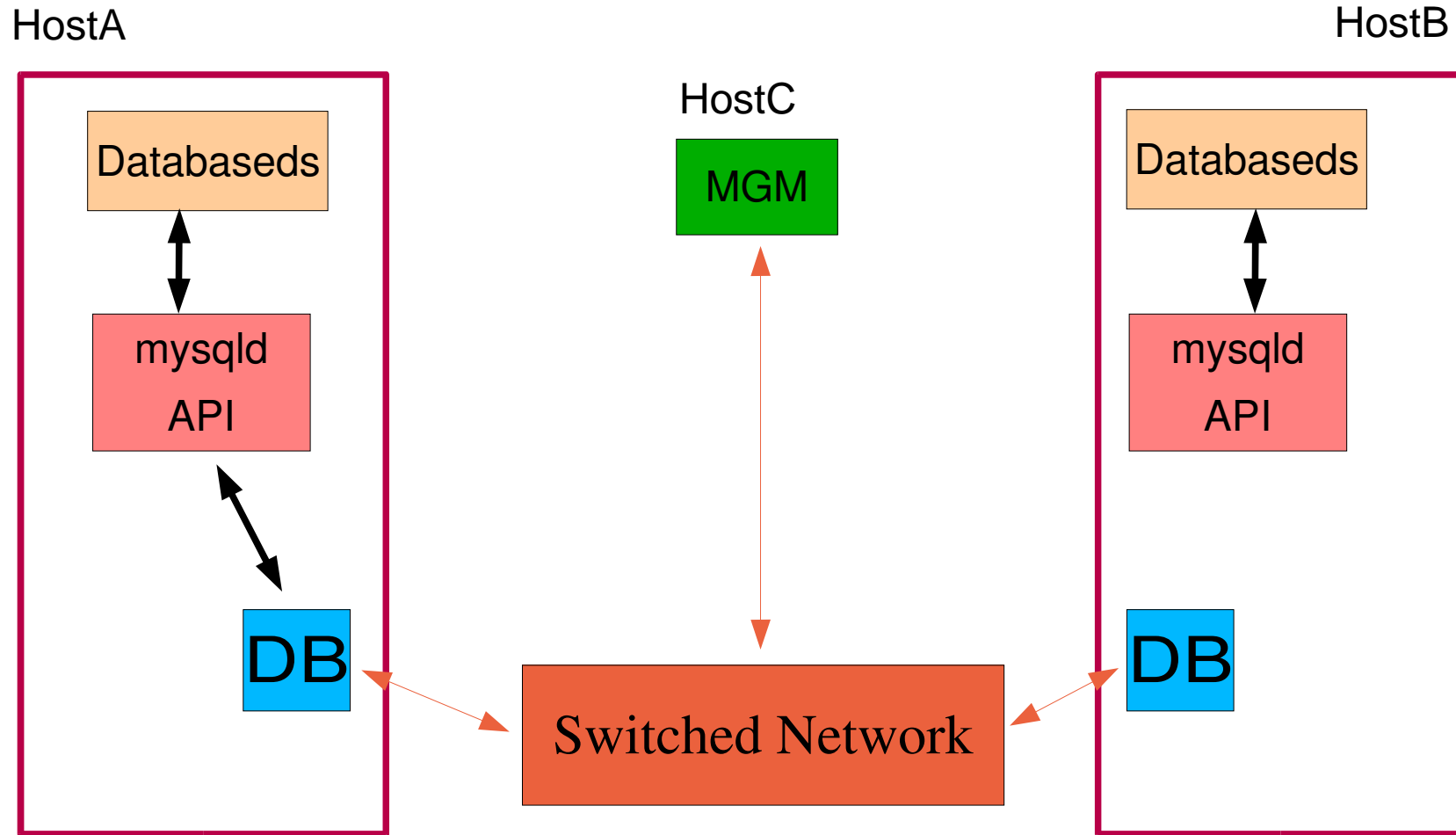


# Cluster Configuration



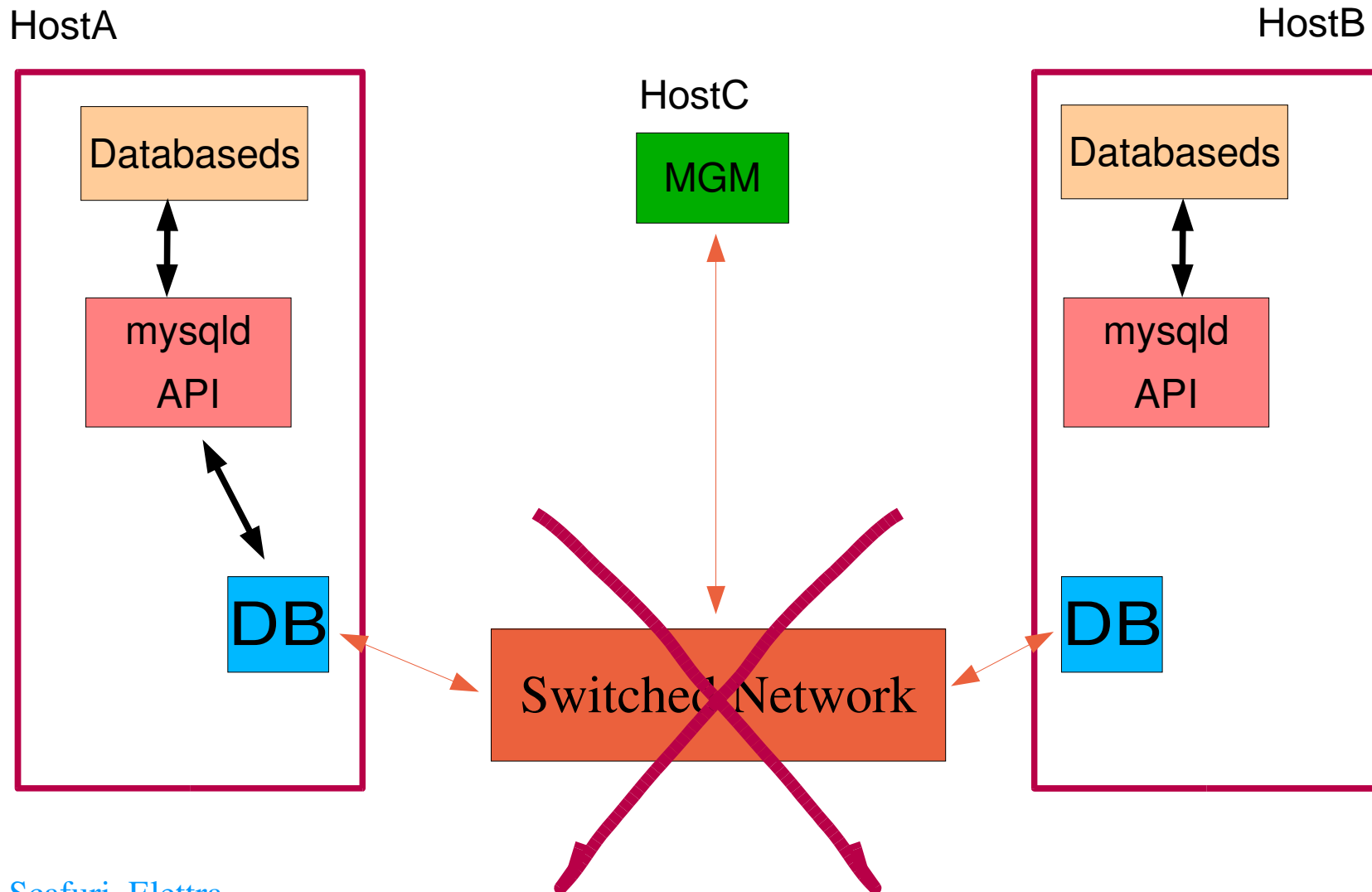


# Failure scenarios



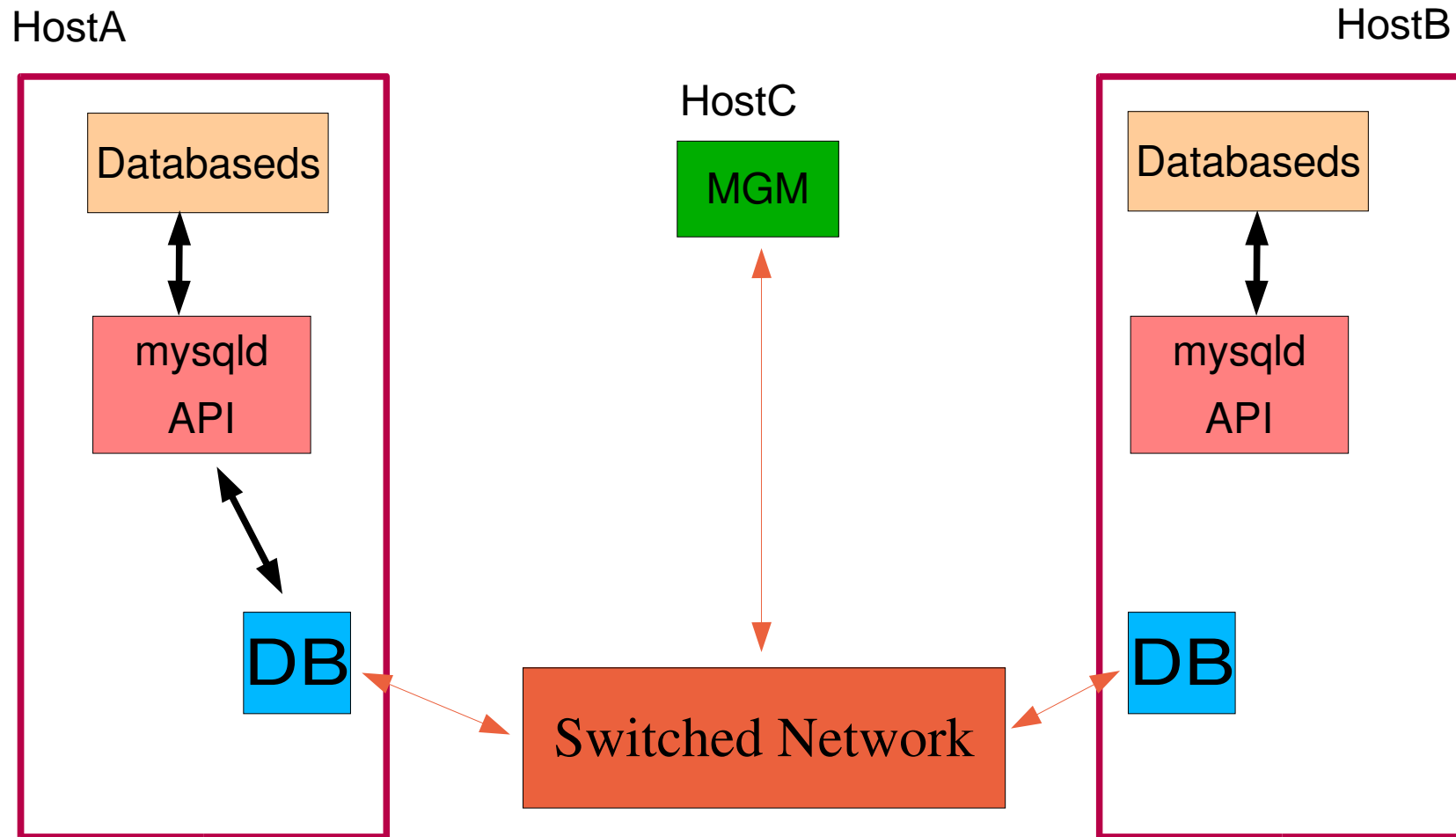


# Failure scenarios



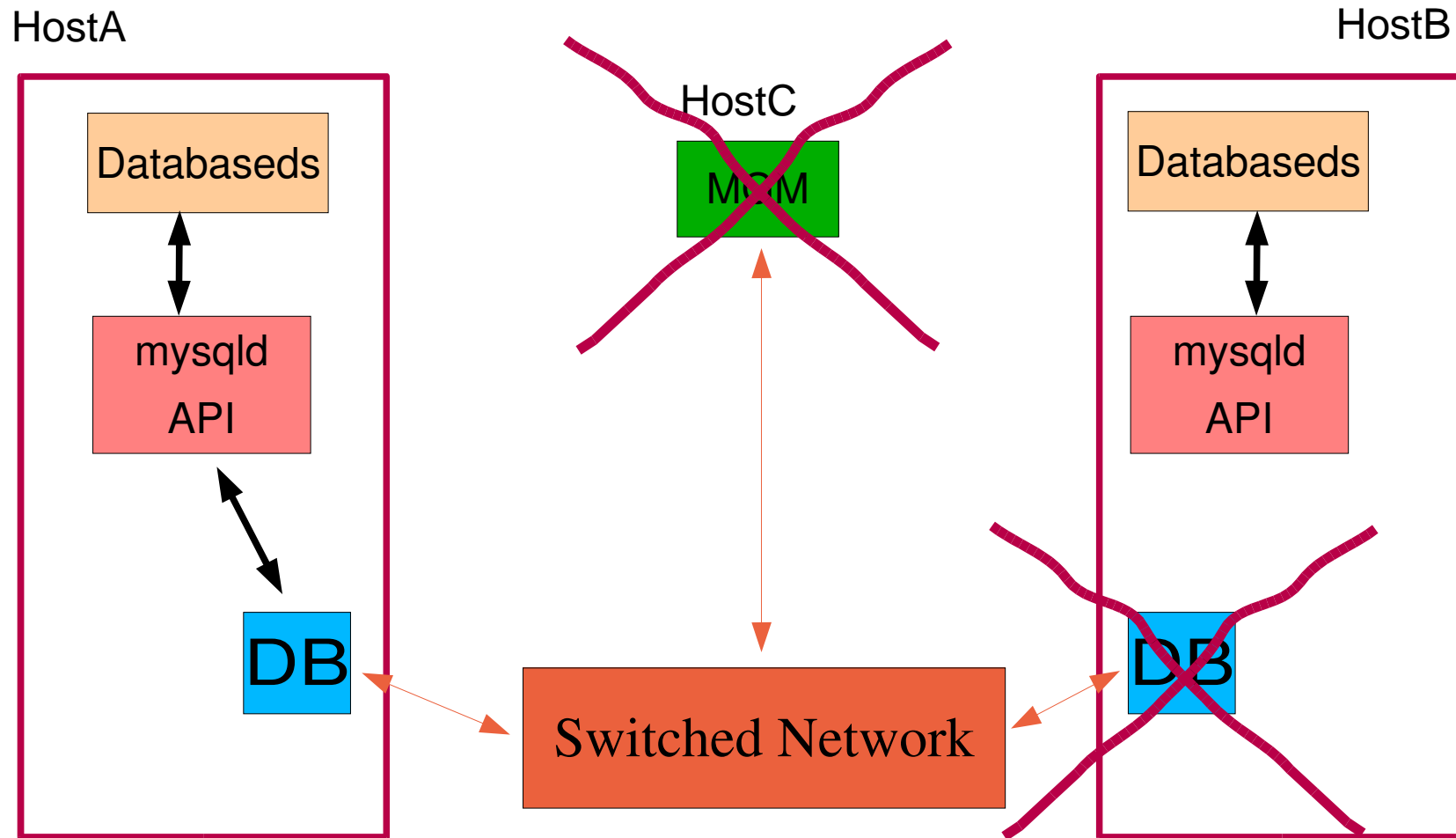


# Failure scenarios

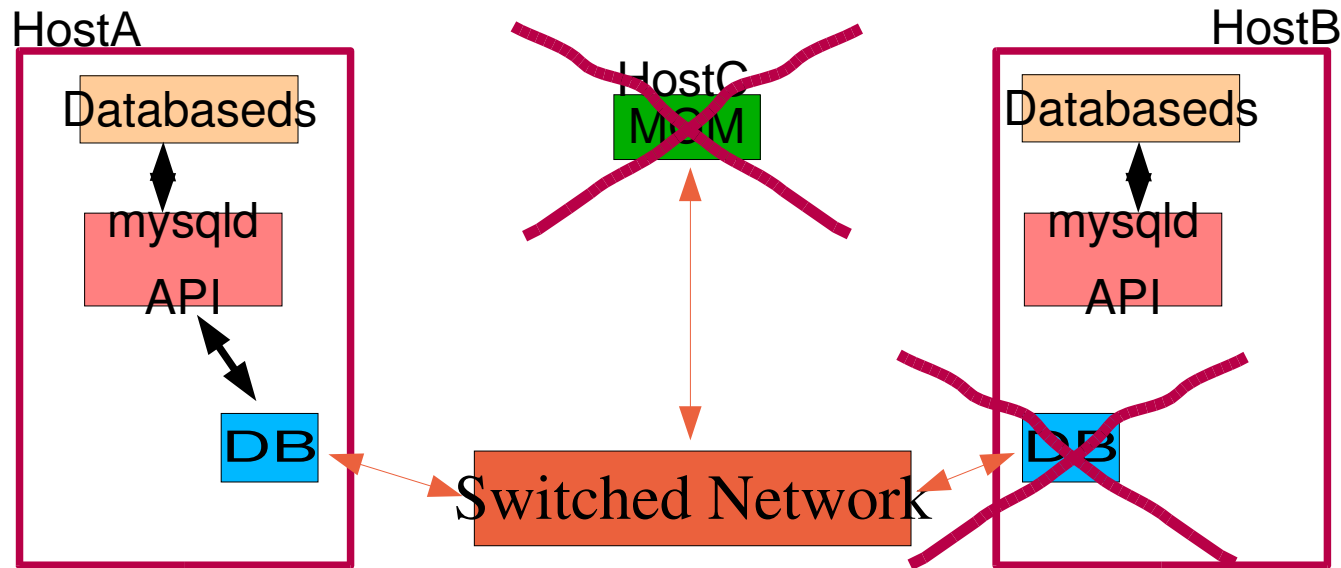




# Failure scenarios



# Failure scenarios



According to MySQL this should trigger a shutdown of the cluster. But it may keep working...



---

# Failure scenarios

- “failures” never led to data loss or corruption
- lack of explicit notification

You have to check the logs and the status of the processes



# Considerations

- excellent tool, does its job
  - but keep in mind its design and goals
- redundancy does not come for free
  - 3 or more servers, lots of memory, disk space
  - non-trivial management (compared to base MySQL)
  - network failure to be dealt with by other means



# Deliverables

## - configuration files

**my.cnf** : MySQL general configuration file with cluster support enabled

**config.ini**: cluster configuration file.

Describes the minimal cluster showed in the slides

## - database creation sql script

**create-db.sql**: creates the TANGO schema with cluster enabled tables. Preloads 2 instances of

Database: `sys/database/2` , `sys/database/4`



---

# Deliverables

extra bonus!

2 HOW-TOs retrieved from the web :

**cluster-howto-1.pdf**

by A. Davies. Official MySQL document

**cluster-howto-2.pdf**

courtesy of LOD Communications, Inc.

<http://www.lod.com>



## To be done:

- testing of deliverables by other institutes
- choose how (if) to distribute the deliverables
- decide whether to add support for multiple tango host in Java:

```
TANGO_HOST=myhost:10000,otherhost:1000
```

now it is understood only by C++ thanks to omniORB.



---

## To be done:

- tuning of db parameters (lots of them!)
- testing of backup and recovery
- trying to break and then recover the database.