
# MySQL Replication Options

Peter Zaitsev,
CEO, Percona
Moscow MySQL User Meetup
Moscow,Russia

# Few Words About Percona



Your Partner in MySQL and MongoDB Success

100% Open Source Software

We work with MySQL, MariaDB, MongoDB, Amazon RDS and Aurora

“No Lock in Required” Solutions and Services

# In This Presentation

Why Replication ?

How to think about Replication

Overview of what MySQL Has to Offer

# Replication

Having Multiple Copies of the data, updated with changes

# Why Replication

Availability

Scalability

Performance

# Availability

Service Stays up when component fails

# Availability via Redundancy

Have more than one system

Works well for stateless systems

Is not enough for databases

# Availability via Replication

Redundant Computing Resource

Paired with Replicated Data

# Component Failure

## Node Failures

- Total Crash
- Process Crash
- Stall/Unresponsive
- Consistency Issues

## Network Failures

- Single Port Failures
- Partitions
- Complicated Failures

# Scalability

Scales Reads

Does not Scale Writes very well

Data Distribution is needed for scaling writes

# Performance

Reduce response time by maintaining replica closer to user

# Talking about

Full Database Replication
- System of Record

Partial Database Replication Can be used as
- Cache
- Synchronization of Client Data

# Where Replication Happens

Storage Level

Database Level

Application Level

# Storage Level Replication

Replication RAID

Typically provides cold standby

Simple choice which works with many systems

Amazon Aurora – Smart Storage

# Database level

Most Flexible

Most Common

Hot/Warm Spare

Some can do Active-Active

# Application Level

Hard to get right

Rarely used, even more so used right

Partial Replication/Syncronization

Smart conflict resolution

Cross Vendor Redundancy

# Replication Properties



www.percona.com

# Number of Writable Nodes

Single Writer (Master)

Write Anywhere (Multi Master)

# Single Master

All writes go to the single node

One way replication stream

Simple

No Conflicts

Replication aware Application or Connector

# Multiple Active Masters

Can write to multiple masters

Replication is multi-directional

More Complicated

Possibility of Conflicts

# Sync or Async

Synchronous Replication

Asynchronous Replication

# Synchronous Replication

Data “Persisted” on Target Server

Guarantees No Data Loss if Source server fails

Conflicts can be easily prevented

Expensive

# What “Persisted” means ?

In Memory
- Assumes power loss of both servers does not happen

On Disk
- Handles Total Loss of Master and Power loss on the Slave

# Asynchronous Replication

Anything not Synchronous

Many different variants exist!

# Asynchronous Properties

## Commit on Master

- What happens with Persistence ?
- What happens with Visibility ?

# Persistence

Uncommitted Data on Master

Committed Data on Master

Date in Slave's memory

Data on Slave's Disk (log)

# Visibility

Results be visible by concurrent sessions before acknowledgement

- Phantom Reads

Results can’t be visible by concurrent sessions

- No Phantom Reads

# Conflits

No Conflict Handling

Conflict Detection

Conflict Resolution

Conflict Prevention

# Acknowledgements

Applies both to Sync and Some Async

How many node have successfully replicated data ?

All ?

One ?

Majority ?

# Level of Control

Global Control vs Database Control

Control by Writer vs Reader

# Number of Replicas

On  (2 nodes)

Two or more (3+ nodes)

# 2 nodes

Basic Redundancy

Need Help with Failure detection and Split Brain

No Redundancy in case of node maintainance

# 3+ Nodes

Much better Redundancy

Quorum based failure handling works out of the box

Can do node maintenance without redundancy loss

# Failure Detection and Promotion

Built-In

Handled by External Tools

# Replica Provisioning

Manual

Automatic

# Gotchas

Distributed systems are complicated. All products have their own gotchas.

# MySQL Replication



www.percona.com

# “Classic” MySQL Replication

Fully Asynchronous

No Failure Detection and Promotion

Manual Provisioning

Can run Multi-Master

No Conflict Handling

# Semi-Sync Replication 5.6

Wait for one of the slaves to ack update before response to client

Commits too soon on the Master

“Phantom Reads” / Possible loss of visible changes

Can switch to asynchronous mode

# Semi-Sync Replication 5.7

Commits on the Master only after slave acknowledges

Update invisible to other clients until slave acknowledges

Can be Configured to have MySQL 5.6 behavior

# Making MySQL Replication Better

MHA

MySQL Failover

PRM

# Percona XtraDB Cluster and “Galera”



www.percona.com

# Percona XtraDB Cluster

Virtually Synchronous Replication

Well known Innodb Storage Engine

All Reads are Local

Parallel Replication

Write to any node behavior

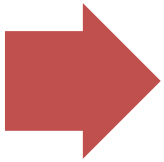
Built in Node Provisioning and HA

Works great in the Cloud !

# PXC vs Galera

## Galera

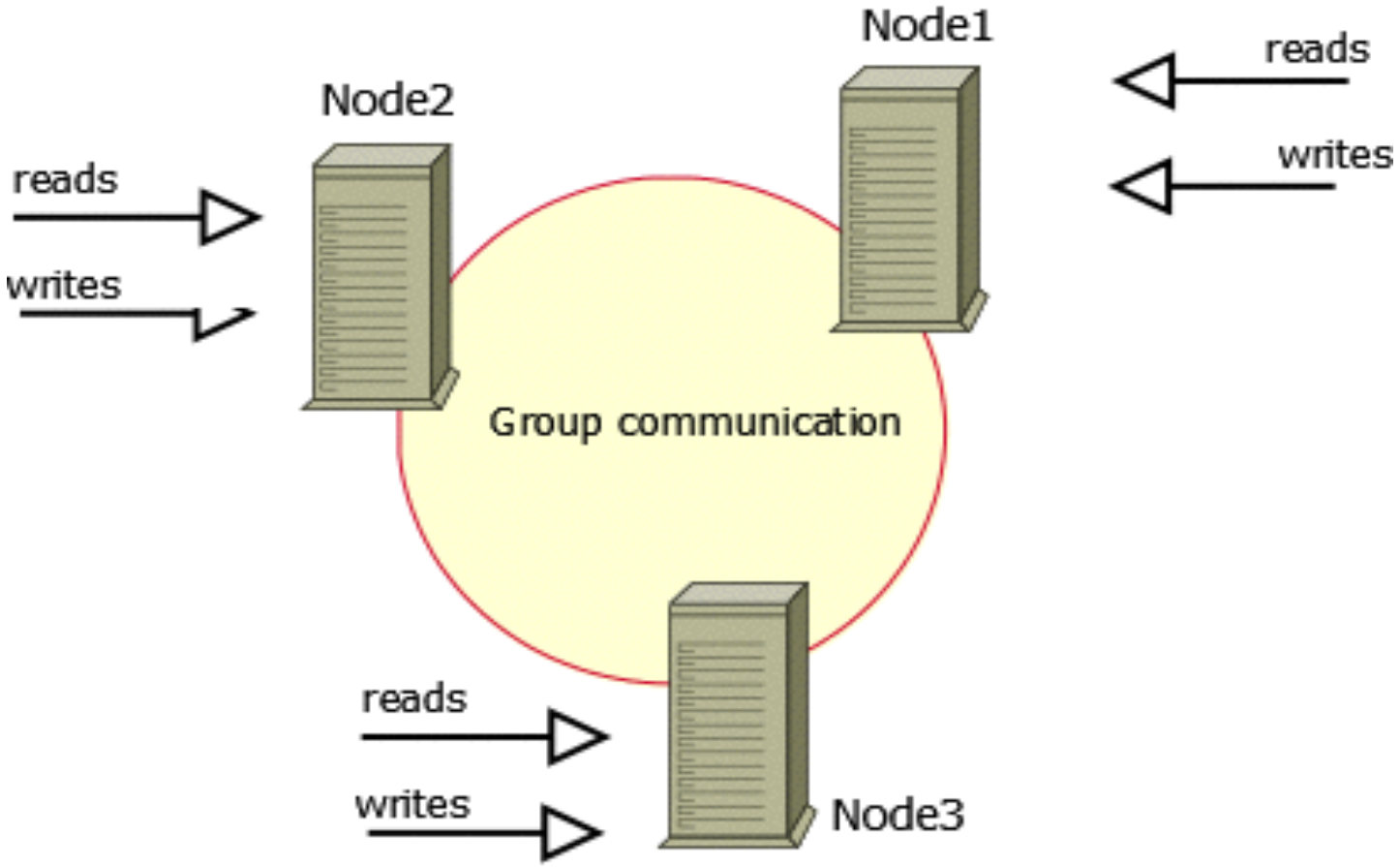
- Is replication technology/library
- Compare to Linux Kernel



## Percona XtraDB Cluster

- Percona Server
- Enabling Enhancements
- Galera Library
- Provisioning Tools
- HA and Load Balancing
- Integrated together and Tested
- Compare to Linux Distribution

# PXC Data Architecture



# Architecture Concepts

All Nodes Have Full copy of Data

Every Node is Equal

No Central Management

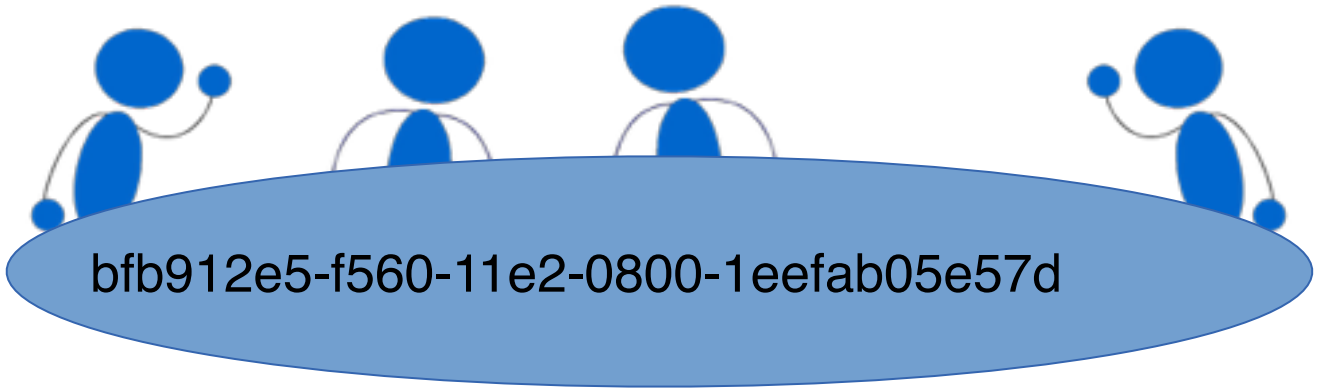
No SPOF

# Understanding Cluster

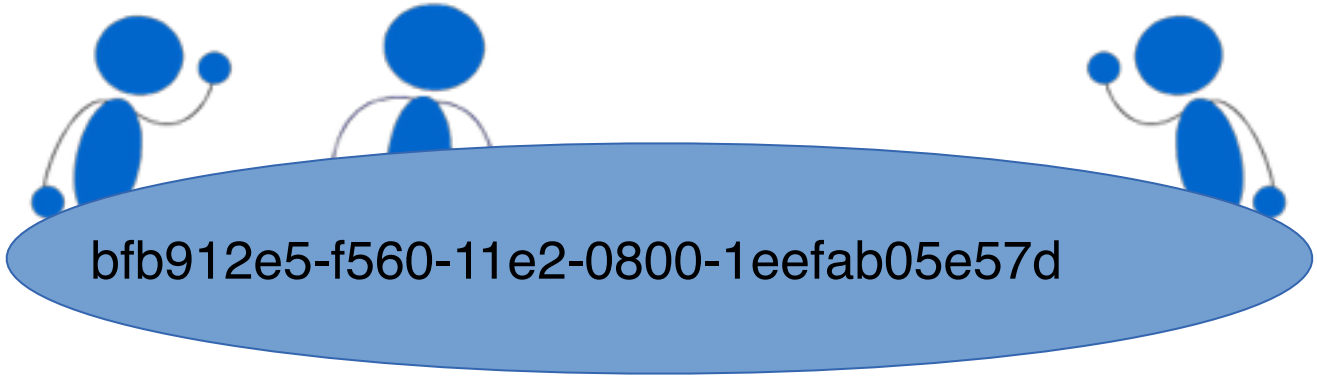
The **cluster** can be seen as a **meeting** !

# PXC (Galera cluster) is a meeting

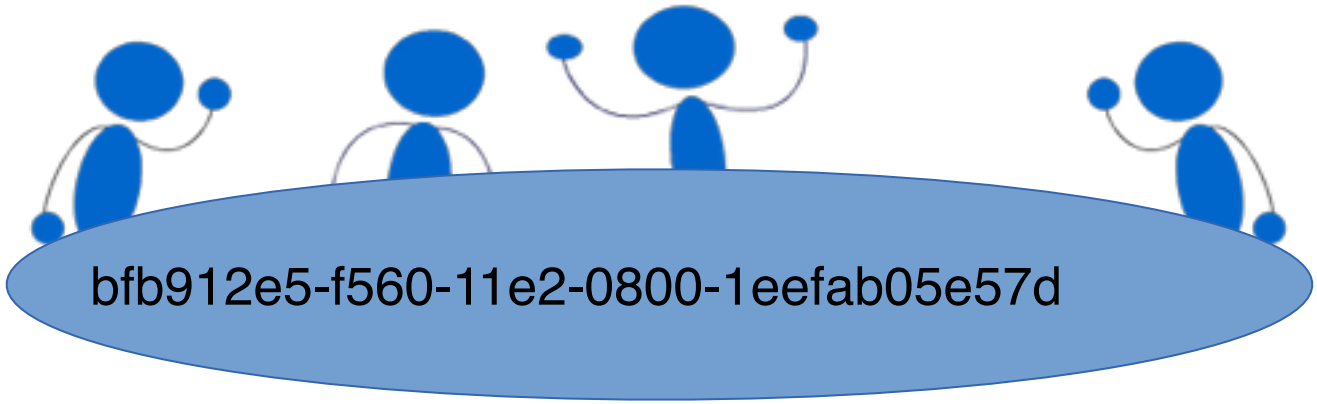




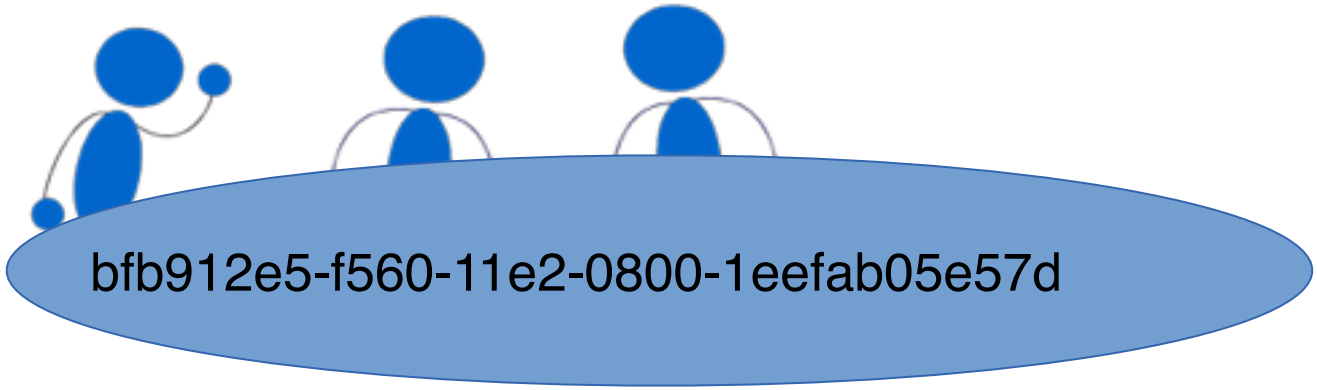
# PXC (Galera cluster) is a meeting



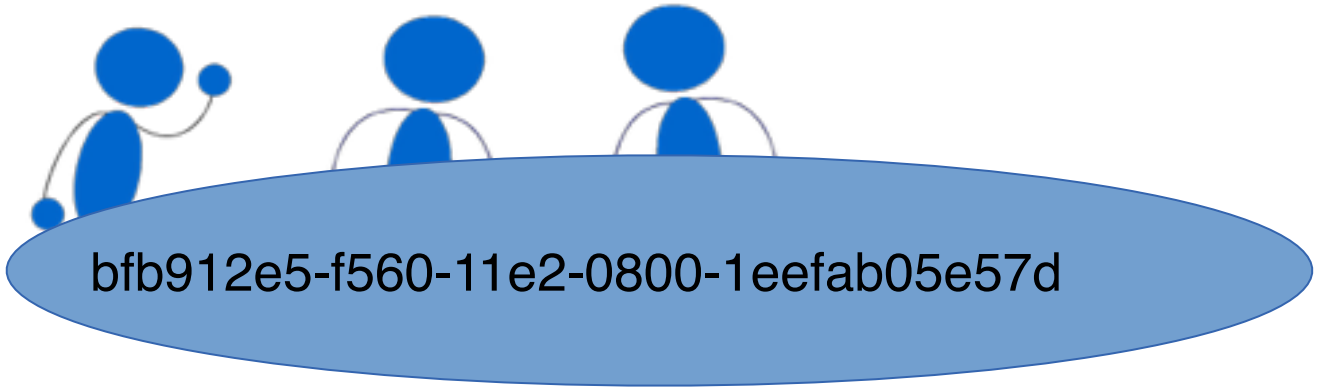

# PXC (Galera cluster) is a meeting



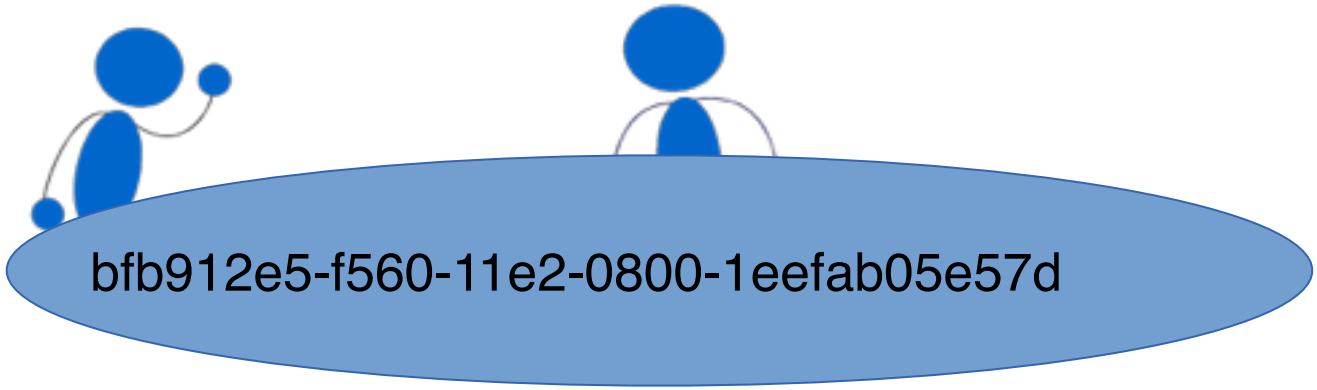

# PXC (Galera cluster) is a meeting



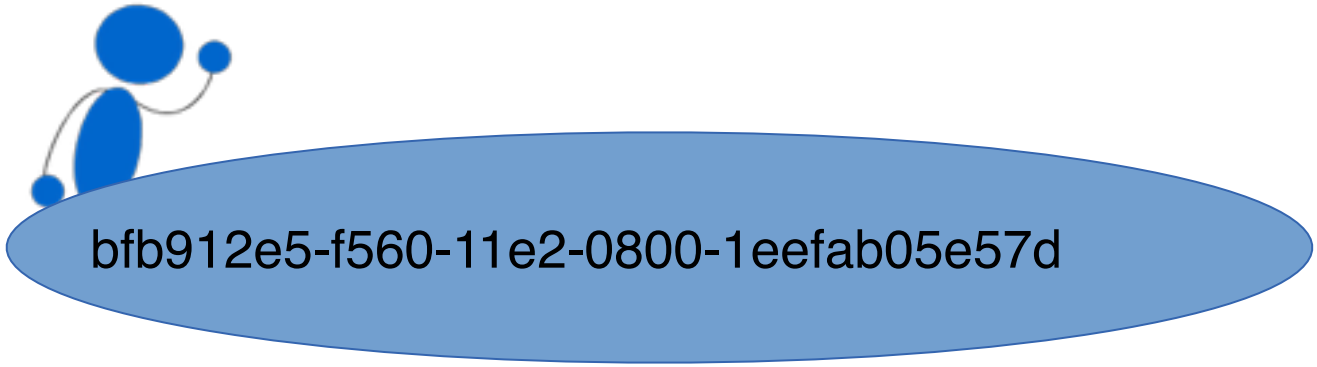
# PXC (Galera cluster) is a meeting



# PXC (Galera cluster) is a meeting

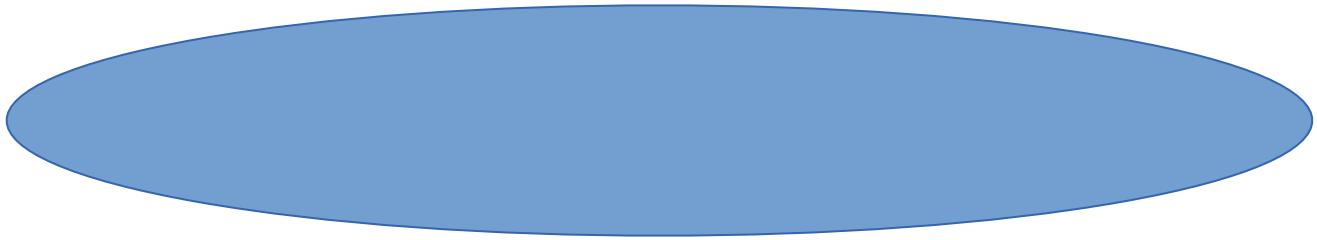


# PXC (Galera cluster) is a meeting

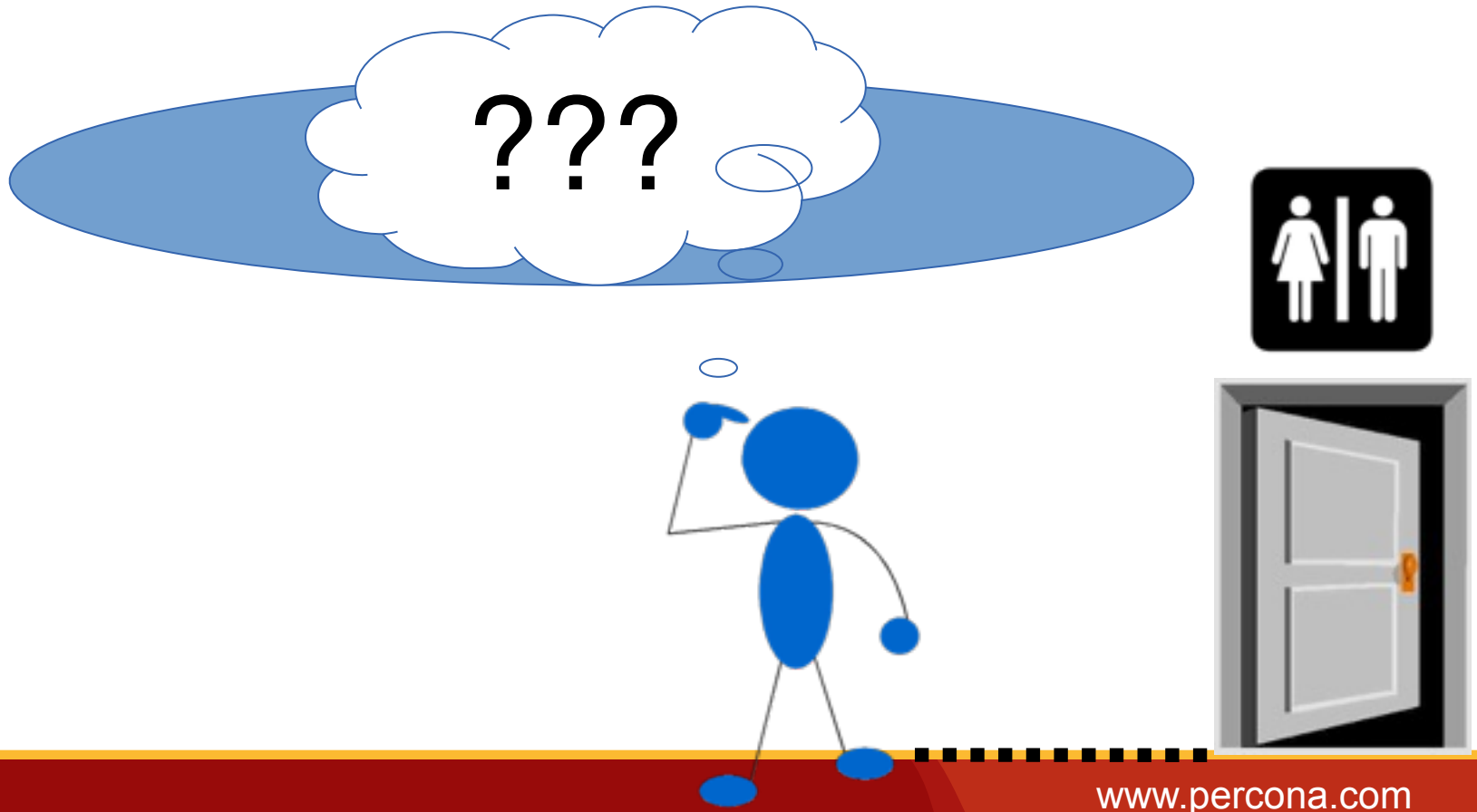


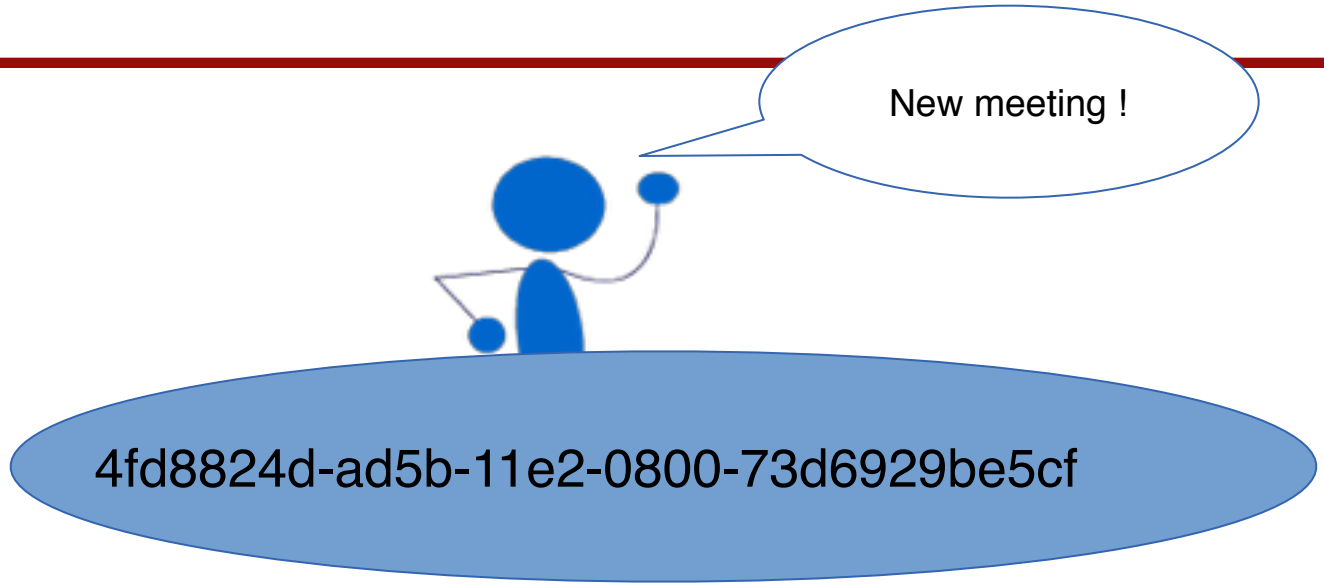
# PXC (Galera cluster) is a meeting

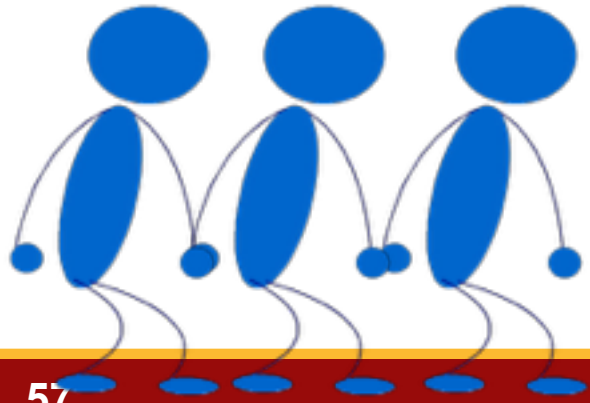




# PXC (Galera cluster) is a meeting




New meeting !
4fd8824d-ad5b-11e2-0800-73d6929be5cf



# Cluster Reads

Reads are always Local

Stale Reads can be allowed or disallowed

# Cluster Writes

Write on one node or Write Anywhere

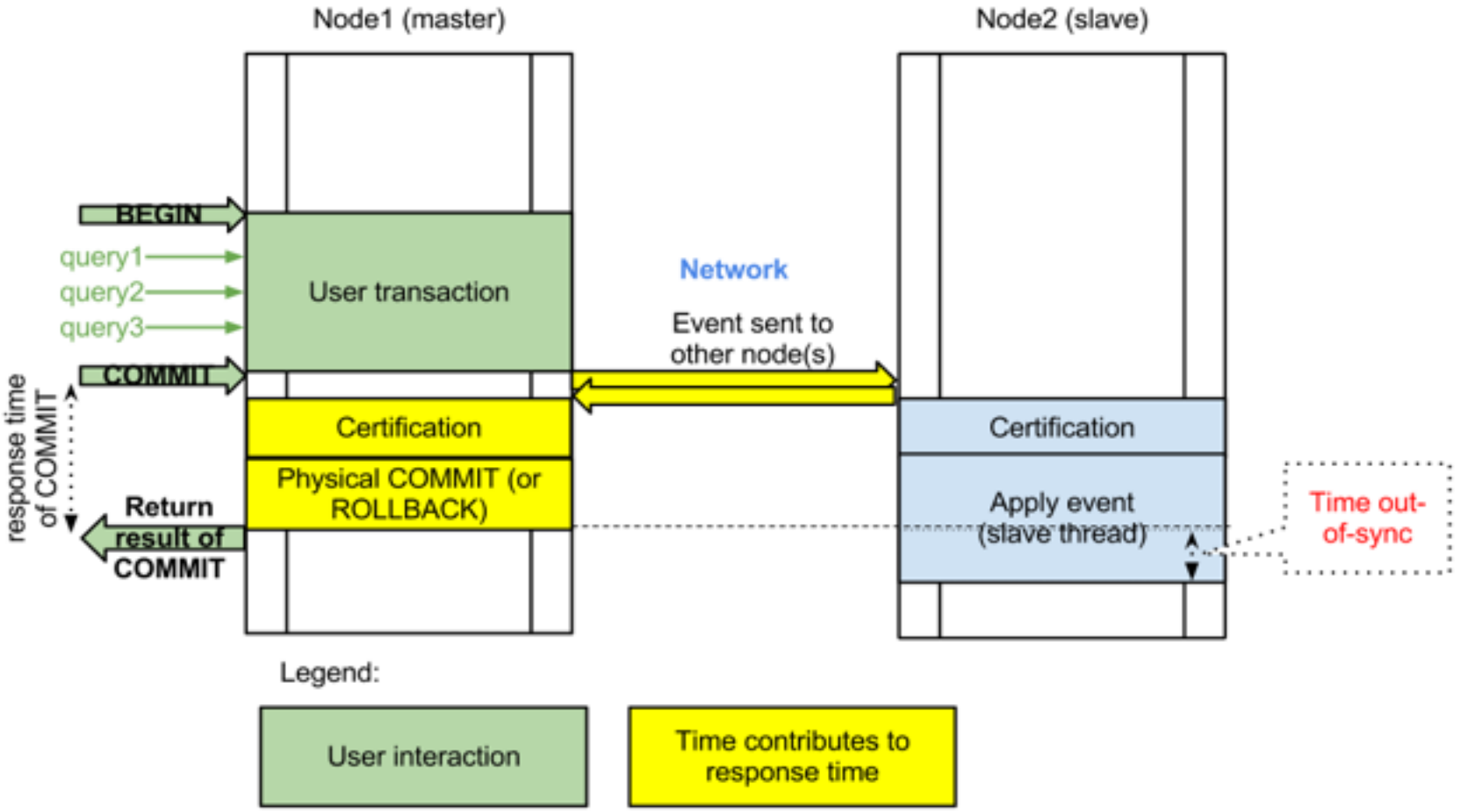
Certification Based Replication

Communication on Commit only

Asynchronous Application

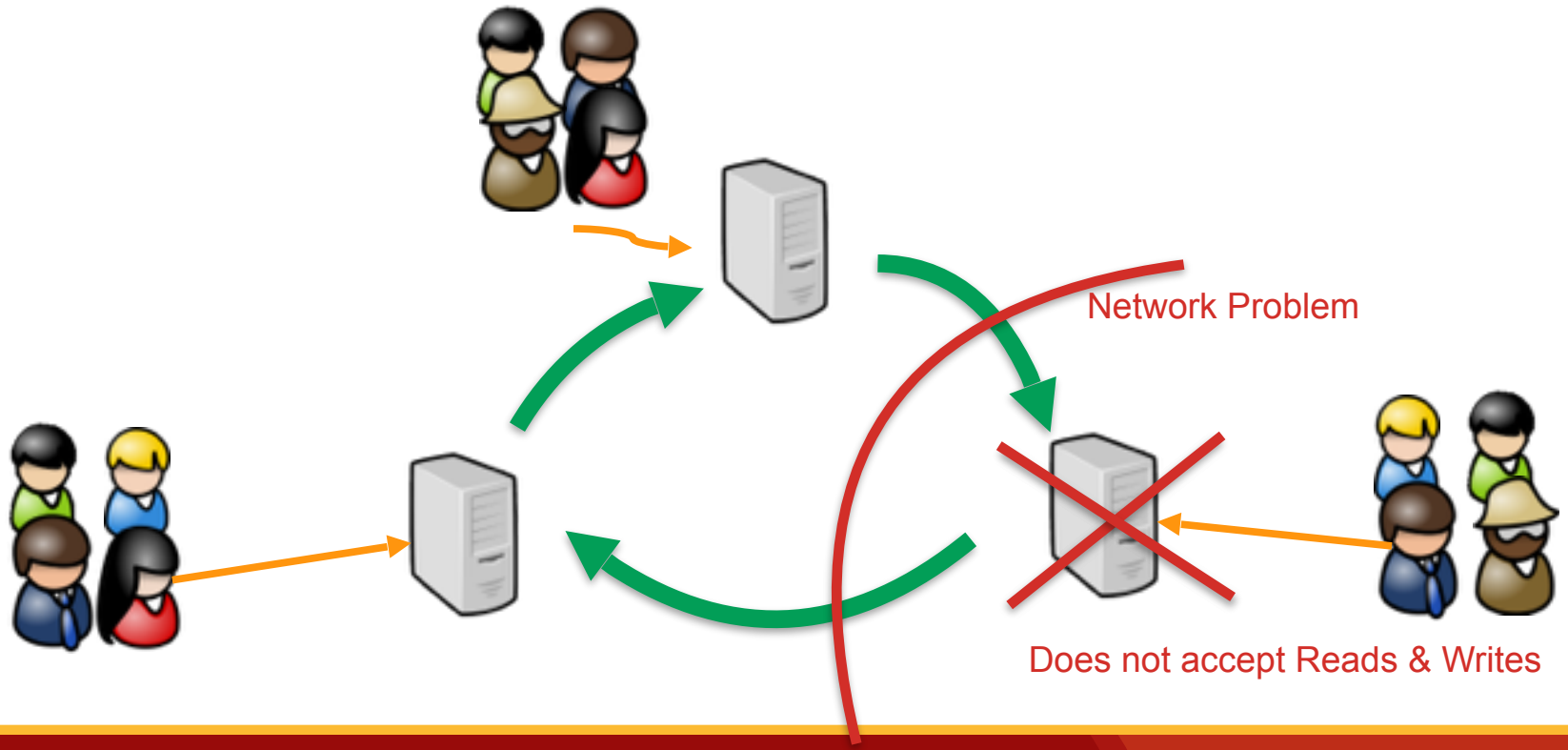
Parallel Replication

# Transaction Commit Flow

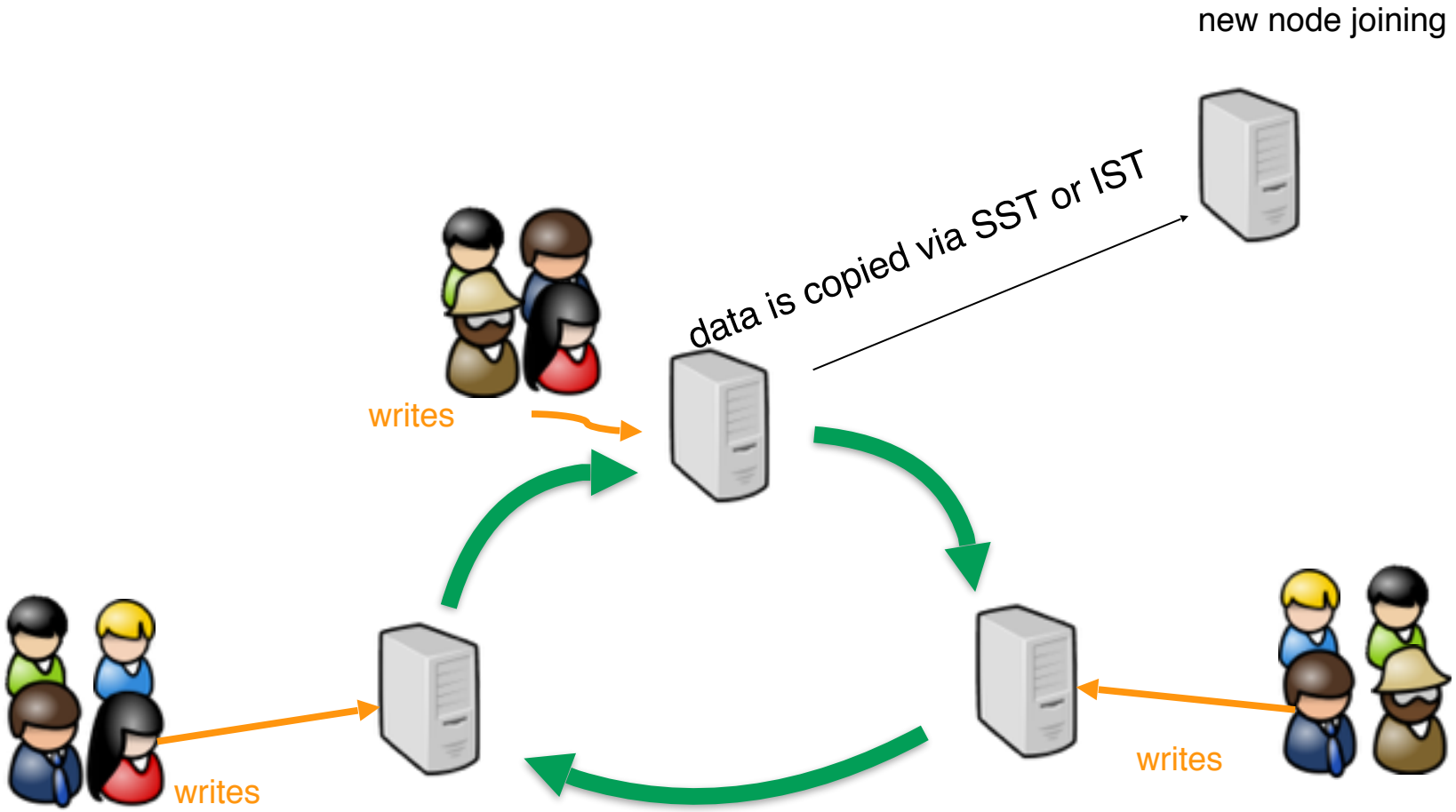


# Quorum

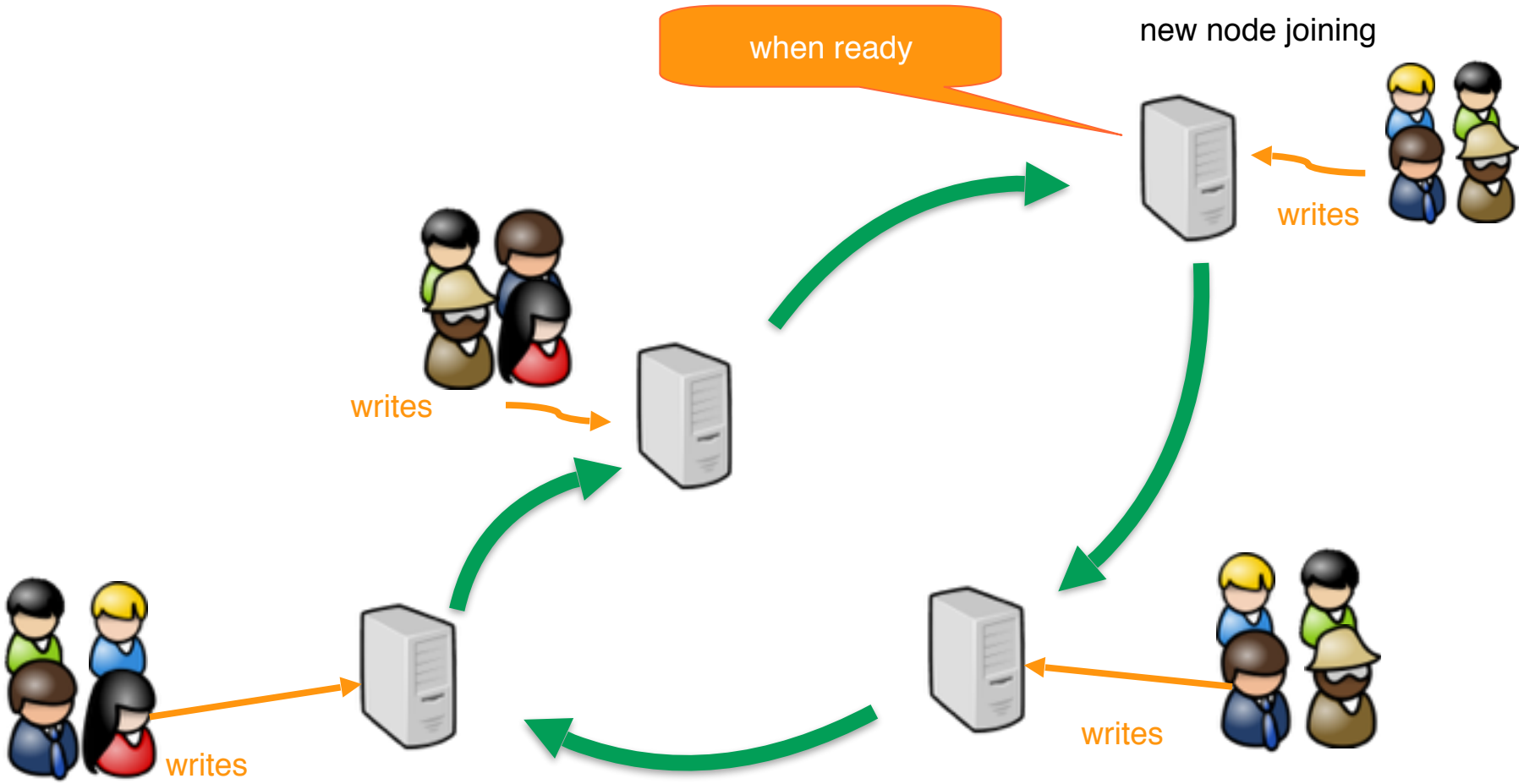
Loss of connectivity



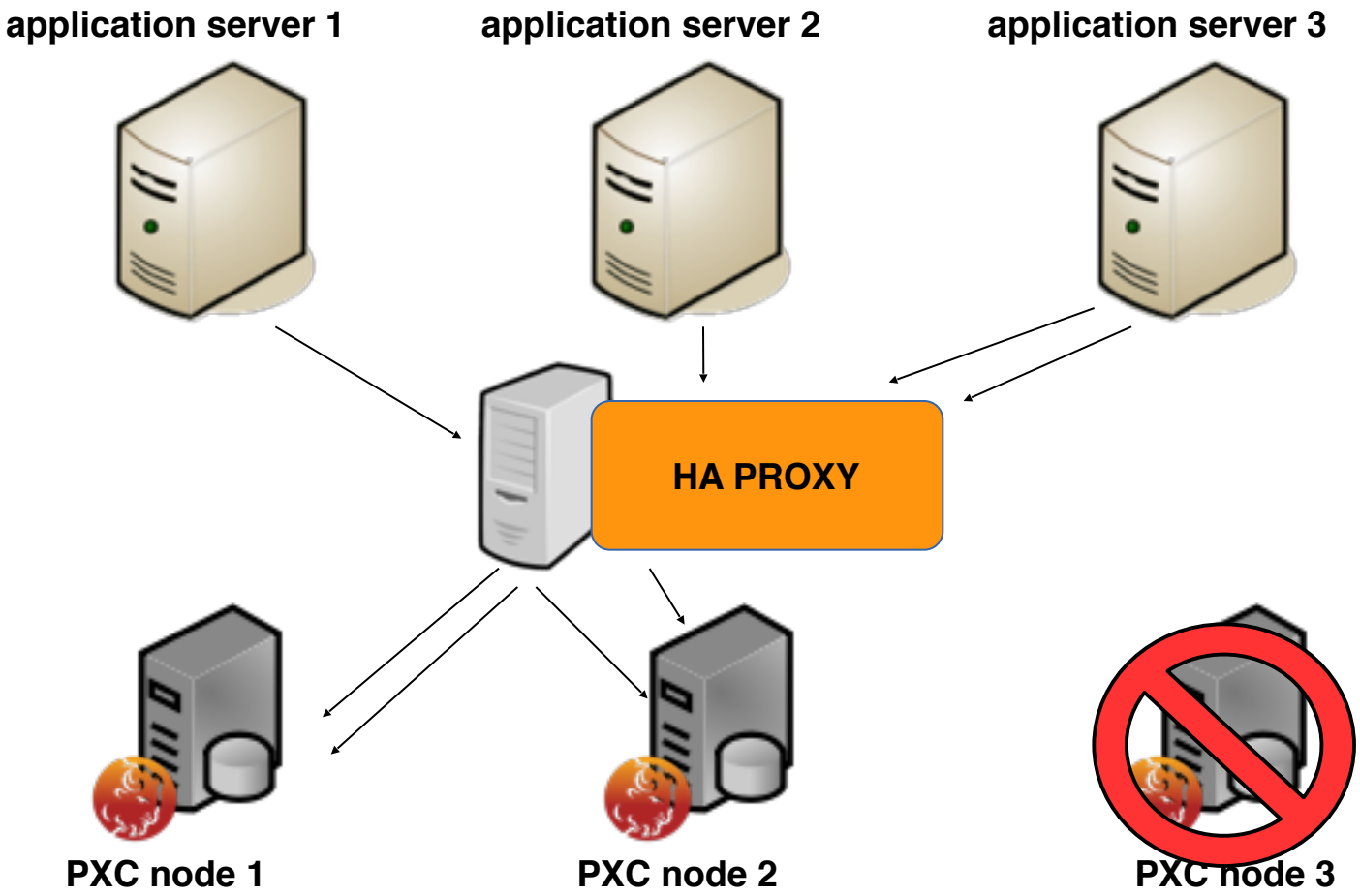
# Automatic Node Provisioning



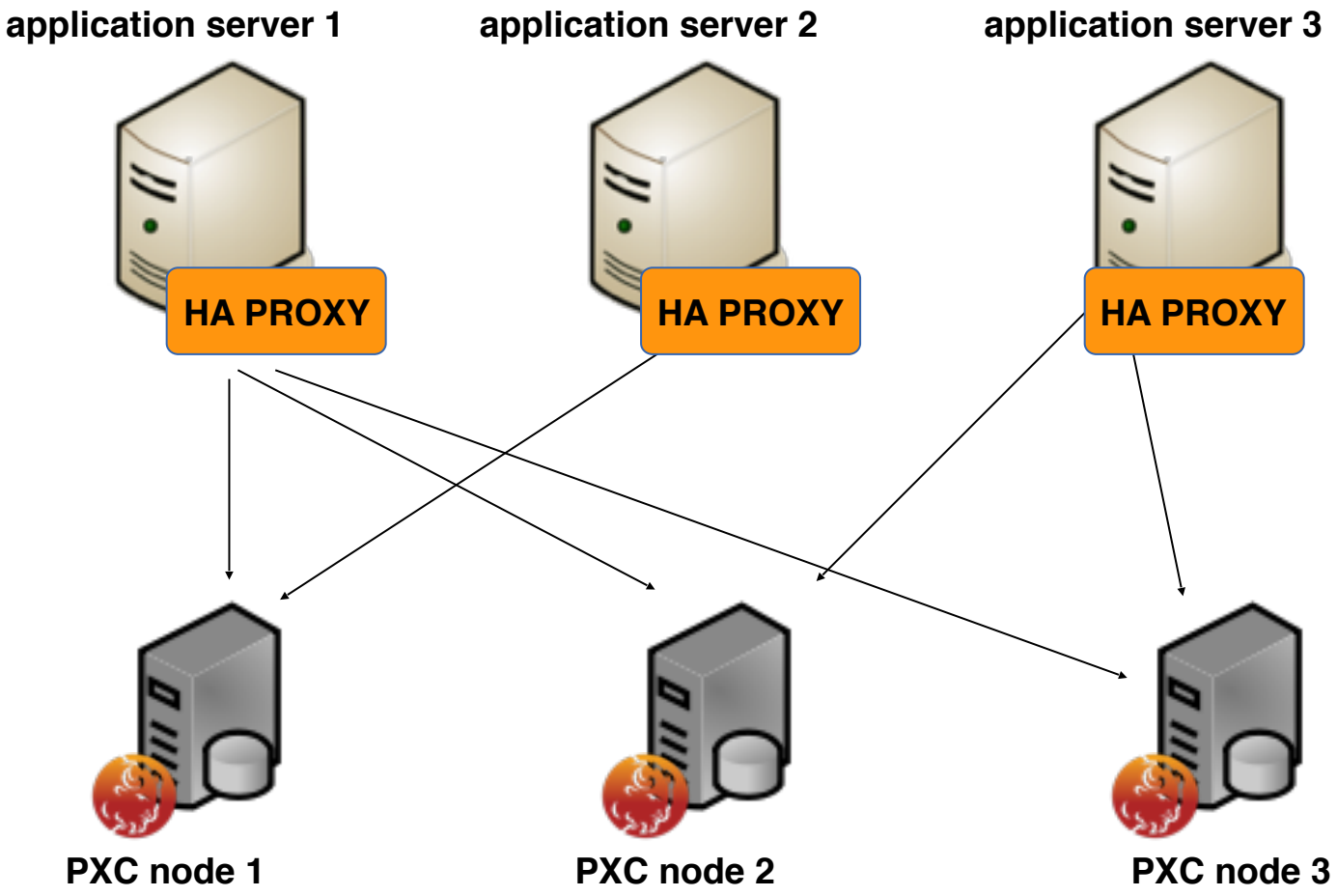
# Automatic Node Provisioning



# Dedicated shared HAProxy



# HAProxy on application side



# Scaling Writes ?

Shard over PXC Cluster rather than MySQL Replicas

# Thinks To Keep in Mind

- Use Innodb storage engine
- Primary Key on all tables
- Avoid Large write transactions (changing many rows)
- Plan Data size for SST time correctly
- Hot Rows
- Optimistic Locking

# MySQL Group Replication

Similar to Galera

Currently in Development

Better integrated with MySQL

No Automated provisioning (yet)

# MySQL Cluster

Mostly In Memory Storage

Synchronous Replication

Pessimistic Locking

Conflict Detection with Async Replication

Niche Use Only

# New Developments

MySQL Fabric

MySQL Router

MariaDB Max Scale

# Percona Live 2016 call for papers is Open



Call for Papers Open until November 29, 2015

MySQL, MongoDB, NoSQL, Data in The Cloud

Anything to make Data Happy!

http://bit.ly/PL16Call

# Thank You!

Peter Zaitsev
pz@percona.com

P.S  We're Hiring
http://bit.ly/PerconaJobs