FREQUENTIS

FOR A SAFER WORLD

Multisite Cluster with Arbitrator

Roman Lukovics

Agenda

1. Introduction
2. Motivation
3. Common HA Architectures
4. Proposed Cluster Solution
5. Architecture Overview
6. Implementation & Technology Stack
7. Security & Automation
8. Operational Insights
9. Q&A

Introduction

Frequentis AG

Global Provider of Safety-Critical Communication Solutions

- **Headquartered in Vienna, Austria**
- Serving:
  - **Civil & Military Air Traffic Control**
  - **Air Defence**
  - **Police, Fire Brigades, Ambulance Services**
  - **Coastguards & Port Authorities**
  - **Railways & Public Transport**
- Active in **~150 countries**, with **49,000+ operator positions**
- Founded in **1947**, with **~30% global market share** in ATC voice communication

Innovation & Safety-Critical DNA

- Focus on:
  - **Remote Tower Technology**
  - **Drone Management Systems**
  - **5G/LTE for Mission-Critical Use**
- Culture built on deep understanding of **safety-critical workflows**

Introduction

Roman Lukovics

Subject Matter Expert

- Over 20 years of IT experience
- Specialized in solution development and monitoring for **secure, reliable critical infrastructure systems**
- Working with **Zabbix** for more than **6 years**
- Experienced with various monitoring tools including HP OpenView, SCOM, and others

Motivation

- Ensure **high availability** across geographically distributed sites
- Eliminate **single points of failure**
- Meet **business continuity** and **disaster recovery** requirements
- Centralized control: **All resources managed by Pacemaker**

Common Zabbix Cluster Architecture

Layer	Component	HA Method	Notes
Frontend	Zabbix Frontend	HAProxy + Web	Load balances to multiple web nodes
Application	Zabbix Server	Native HA	Uses database locking for leader election
Database	PostgreSQL	Patroni or repmgr	Replication with failover

Side by side comparison

Feature / Aspect	Pacemaker + Arbitrator	HAProxy + Zabbix HA + repmgr/Patroni
Failover Control	Cluster-level, coordinated	Service-level, independent
Split-brain Prevention	Yes (with Arbitrator)	No (manual or app-level only)
Multisite Readiness	Designed for multisite	Requires extra routing/config
Tool Complexity	One integrated cluster stack	Multiple tools to configure and maintain
Zabbix & DB Integration	Managed together in cluster	Handled separately

Technology stack

- **Operating System:** Red Hat Enterprise Linux 9
- **Cluster Management:** Pacemaker 2.1
- **Monitoring Platform:** Zabbix 7.0 LTS
- **Database:** PostgreSQL 16 with TimescaleDB extension

Using Pacemaker for Full-Stack HA in Zabbix

- **Pacemaker** manages failover for:
  - **Zabbix Server**
  - **Frontend**
  - **PostgreSQL**
- Ensures **only one active instance** per component using:
  - **Virtual IPs**
  - **Shared storage**
- All resources run as **systemd units**
- **Arbitrator (quorum device)** acts as a neutral third party:
  - Resolves **split-brain scenarios**
  - Prevents **data corruption**

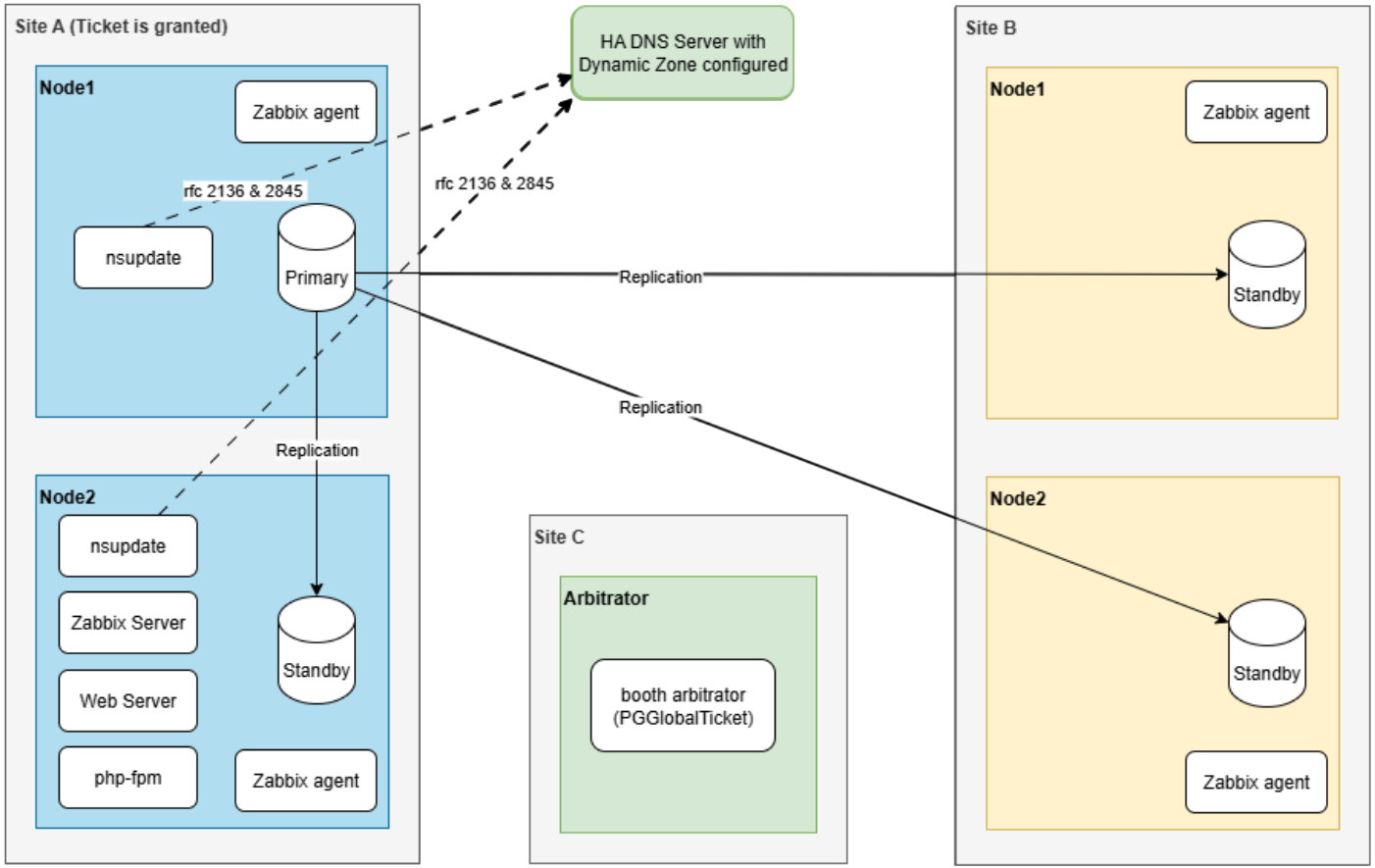
Proposed Cluster Solution

- **Cluster Management:**
  - Pacemaker 2.1 orchestrates all service and node failover operations.
- **Zabbix High Availability:**
  - Native Zabbix HA is **not used**.
  - Zabbix server runs as a **cluster-managed systemd service**.
- **Database Replication:**
  - PostgreSQL 16 with TimescaleDB handles **native streaming replication**.
  - Pacemaker monitors and restarts PostgreSQL as a resource if needed.
- **Virtual IP / FQDN Handling:**
  - Pacemaker controls a **Virtual IP (VIP)** to provide a consistent endpoint.
  - FQDN (DNS name) is tied to the Virtual IP.
- **Failover Behavior:**
  - Automatic service migration to Site B if Site A fails.
  - VIP is reassigned to the active site seamlessly.

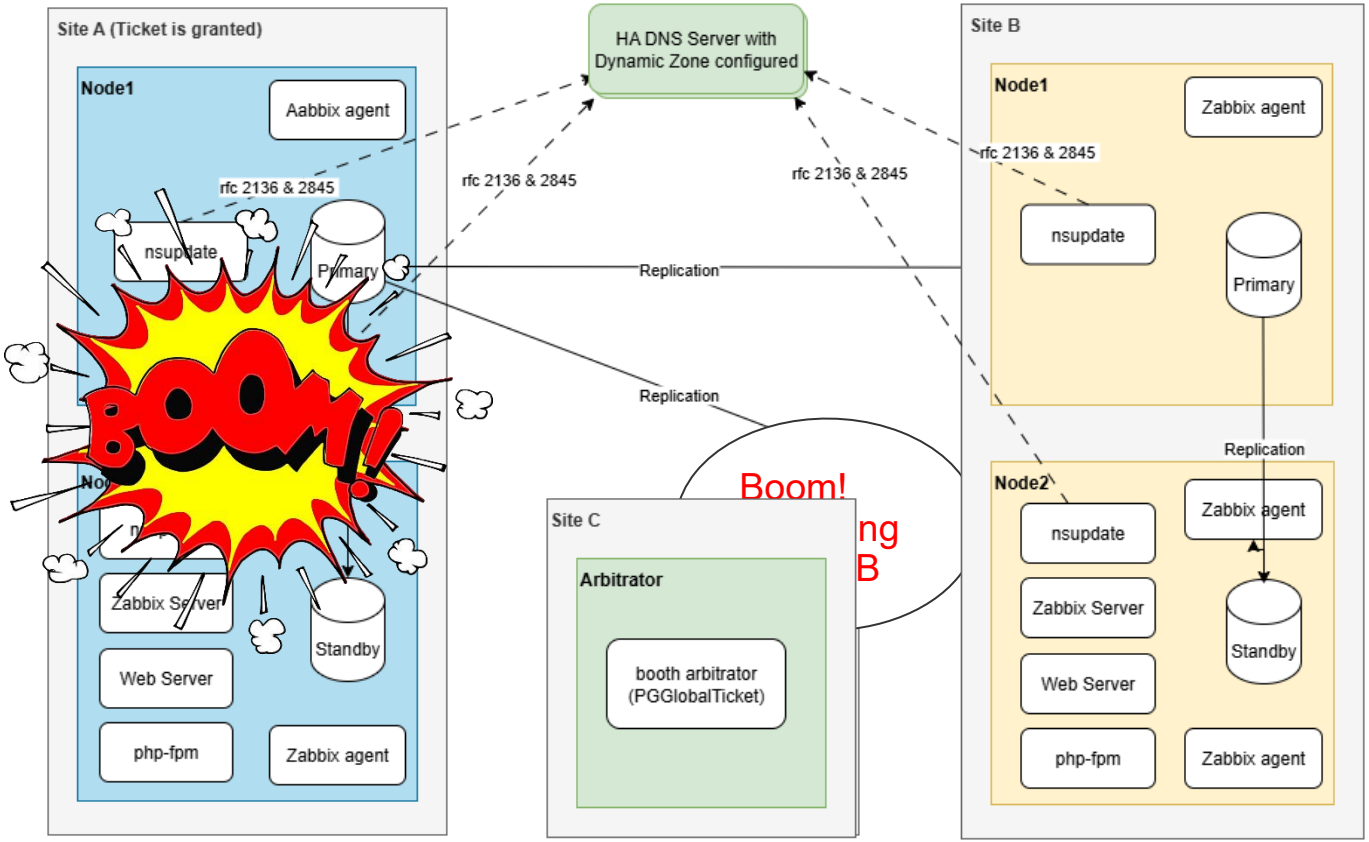
Environment Overview

Site A	Site B
• 2 VM‘s • 1 IP node 1 • 1 IP node 2 • 1 VIP Site A IP	• 2 VM‘s • 1 IP node 1 • 1 IP node 2 • 1 VIP Site B IP
Site C (Arbitrator)	**Global IP‘s**
• 1 VM • 1 IP Arbitrator node	• 1 VIP PostgreSQL • 1 VIP Zabbix/Frontend

Architecture Overview



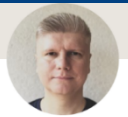
Architecture Overview - Cluster Reaction to Failure



Security Measures

Access Control	System Hardening
• Role-based access • Strong passwords • Secure SSH (key-based, no root login)	• SELinux enforced • Disable unused services • Restrict kernel modules • CIS Benchmarks compliance
Network & Communication	**Monitoring & Maintenance**
• Firewall: only required ports • SSL for PostgreSQL (replication & access) • SSL for Web & API • Time synchronization (NTP/chrony)	• Centralized logging • Regular updates & patching • (Optional) Audit logging

Q & A



Roman Lukovics

Subject Matter Expert (Monitoring)

FREQUENTIS