



US Embassy Mission Poland – Server Solution

Document Identity

| Type | Value |
|---------------------|-----------------|
| Document Identifier | USEmbassyServer |
| Document Author | Agitive |
| Document Owner | Agitive |

Change Log

| Revision Id | Revision Date | Author | Description |
|-------------|---------------|---------|---------------------------------|
| 1.0 | 23.12.2013 | Agitive | Document creation |
| 1.1 | 21.03.2013 | Agitive | Load balancer description added |

1 Introduction

1.1 Document purpose

This document intends to describe server architecture solution for back-end system which supports CMI of US Embassy Mission Poland mobile application.

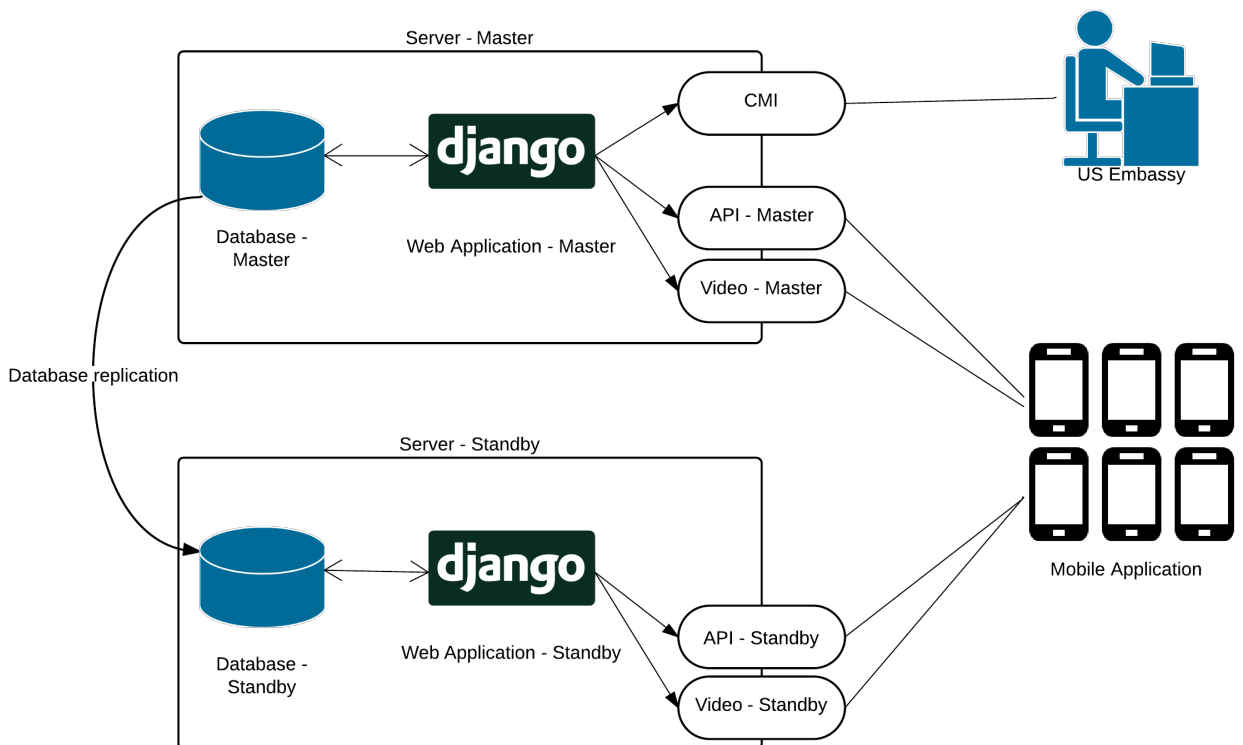
1.2 Problem description

Server Architecture has to provide:

- 99.9% network availability,
- Handling up to 50 000 unique users daily,
- Up to 200 simultaneous video streams.

2 Server Solution

2.1 Schema



2.2 Description

Server solution contains two dedicated servers: master and standby. On each one of them the environment configuration is the same. Both include: MySQL server/database, Django application, additional Python libraries, and interfaces to communicate with application.

2.3 MySQL database

Data provided by US Embassy (via CMI) is stored in Master Database. Replication mechanism enables data from Master Server to be replicated to Standby Server. This means that the data accessible for Django application and Mobile Client will be the same on Standby server, as on Master server.

2.4 Django application

Django is an open source Python framework which allows to create web application. It is responsible for main logic processes in system:

- Content management (via CMI),
- Content presentation (providing data from database and videos to mobile application).

2.5 Interfaces

- API – interface used for communication between the server and mobile application,
- Video – interface used for serving videos from file system,
- CMI (only Master Server) – interface which allows to manage content in system.

2.6 Load balancing

Mobile application will establish connection with randomly selected server. Probability that given server will be selected is 50%, thus load distribution in the long run will be equal.

Server selection will be repeated before request if any of the following conditions is met:

- previous selection expired (expiration time = 1 hour),
- application was restarted.

If selected server is not responding (response time is longer than 3 seconds), mobile application will try to connect with second server. This means that even if one of the servers is offline, user should be able to use mobile application without noticing that fact.