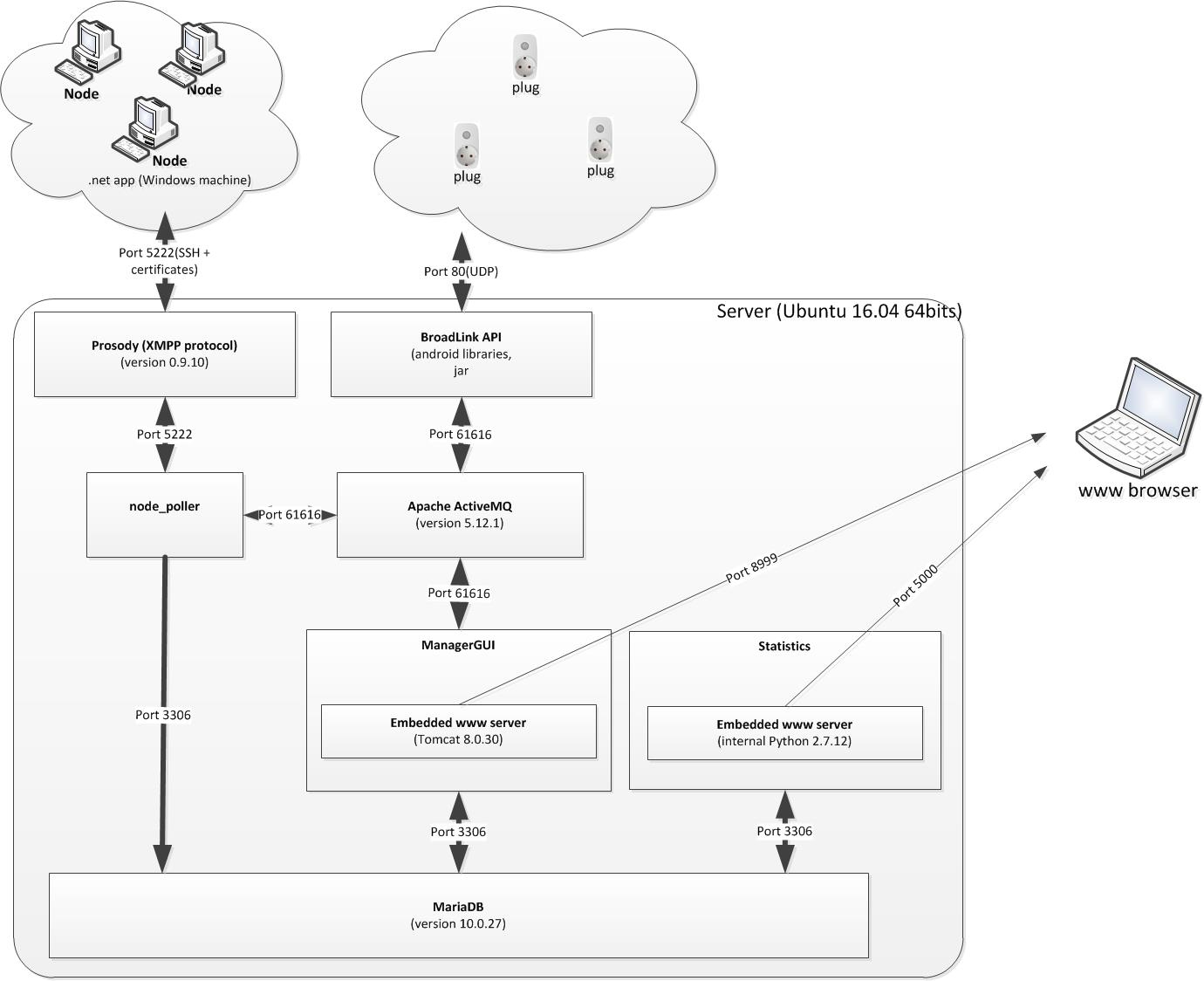
Szczegółowa architektura rozwiązania pokazana została na rysunku poniżej:

****

Wykazane zostały wszystkie porty używane w systemie, które muszą pozostać otwarte, aby poszczególne komponenty mogły się komunikować między sobą. Wskazane zostały także numery wersji oprogramowania użyte w systemie. Do komunikacji pomiędzy poszczególnymi komponentami systemu służą dwa rozwiązania:

- **ActiveMQ** do wewnętrznej komunikacji pomiędzy elementami zainstalowanymi na tej samej maszynie (serwerze).

- **Prosody** do komunikacji zewnętrznej pomiędzy aplikacją kliencką i częścią serwerową. Ta komunikacja **dodatkowo jest zaszyfrowana i oparta o rozwiązanie PKI**. Ważność certyfikatów ustawiona jest na okres **jednego roku**.

**Warto zwrócić uwagę iż wszystkie zewnętrzne kanały komunikacyjne są zaszyfrowane**.

Ważnym komponentem umożliwiającym działanie całego systemu jest baza danych, użyto najnowszego serwera bazodanowego MariaDB zawierającego szereg poprawek bezpieczeństwa.

Część serwerowa została zainstalowana na systemie Ubuntu Linux wersja 16.04 64bity. System Ubuntu jest dobrze znany i dość dokładnie przetestowany. W rozwiązaniu zostanie użyte szereg mechanizmów bezpieczeństwa na poziomie systemu operacyjnego (firewall, uwierzytelnianie, autoryzacja itd.|).

Oprogramowanie najczęściej dostarczane jest w postaci wirtualnej maszyny. Poniżej zostały przedstawione listy bibliotek (oraz ich wersji) używanych przez poszczególne komponenty systemu.

**SmartPlug:**

gson - 2.6.2

org.apache.activemq.activemq-client - 5.12.1

org.apache.commons.commons-lang3 - 3.3.2

**NodePoller:**

com.mchange.c3p0 - 0.9.5-pre10

commons-cli - 1.2

org.apache.commons.commons-lang3 - 3.3.2

**Manager GUI:**

org.springframework.boot.spring-boot-starter-aop

org.springframework.boot.spring-boot-starter-web

org.springframework.boot.spring-boot-starter-security

pl.pragmatists.JUnitParams - 1.0.5

org.assertj.assertj-core - 3.5.1

org.springframework.spring-jms

org.apache.activemq.activemq-broker

org.springframework.boot.spring-boot-starter-test

**WSPÓLNE dla wszystkich komponentów powyżej**:

log4j - 1.2.17

commons-configuration - 1.6

jivesoftware.smack - 3.1.0

jivesoftware.smackx - 3.1.0

junit - 4.4

com.google.guava - 19.0

grinfinity.broadlinkapi.SmartPlugApi - 0.1

gson - 2.5

org.springframework.boot.spring-boot-starter-log4j

log4j.apache-log4j-extras - 1.0

org.mariadb.jdbc.mariadb-java-client - 1.3.7

org.xerial.sqlite-jdbc - 3.6.20

org.springframework.boot.spring-boot-starter-data-rest

org.springframework.boot.spring-boot-starter-data-jpa

org.springframework.boot.spring-boot-starter-jdbc

xom.xom 1.2.5

org.powermock.powermock-api-mockito - 1.6.1

org.powermock.powermock-module-junit4 - 1.6.1

org.assertj.assertj-core - 1.0.0

**Statystyki**

Python 2.7

flask.ext.mysql mysql

json

MySQLdb

flask.Flask

flask.render\_template

bson.dumps

decimal

sys

ast

datetime

time

dateutil.parser

datetime timedelta

psutil

ConfigParser

collections OrderedDict

Embedded Python HTTPserver