Distributed Systems and Middleware Technologies
879II – 22/23

Monitoring system for IoT applications using an Erlang distributed server

Bruno Augusto Casu Pereira De Sousa
Arsalen Bidani
Huang Chuer

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

UNIVERSITÀ DI PISA

Contents

- Functional Requirements
- System Architecture
- Technology Stack

Functional Requirements

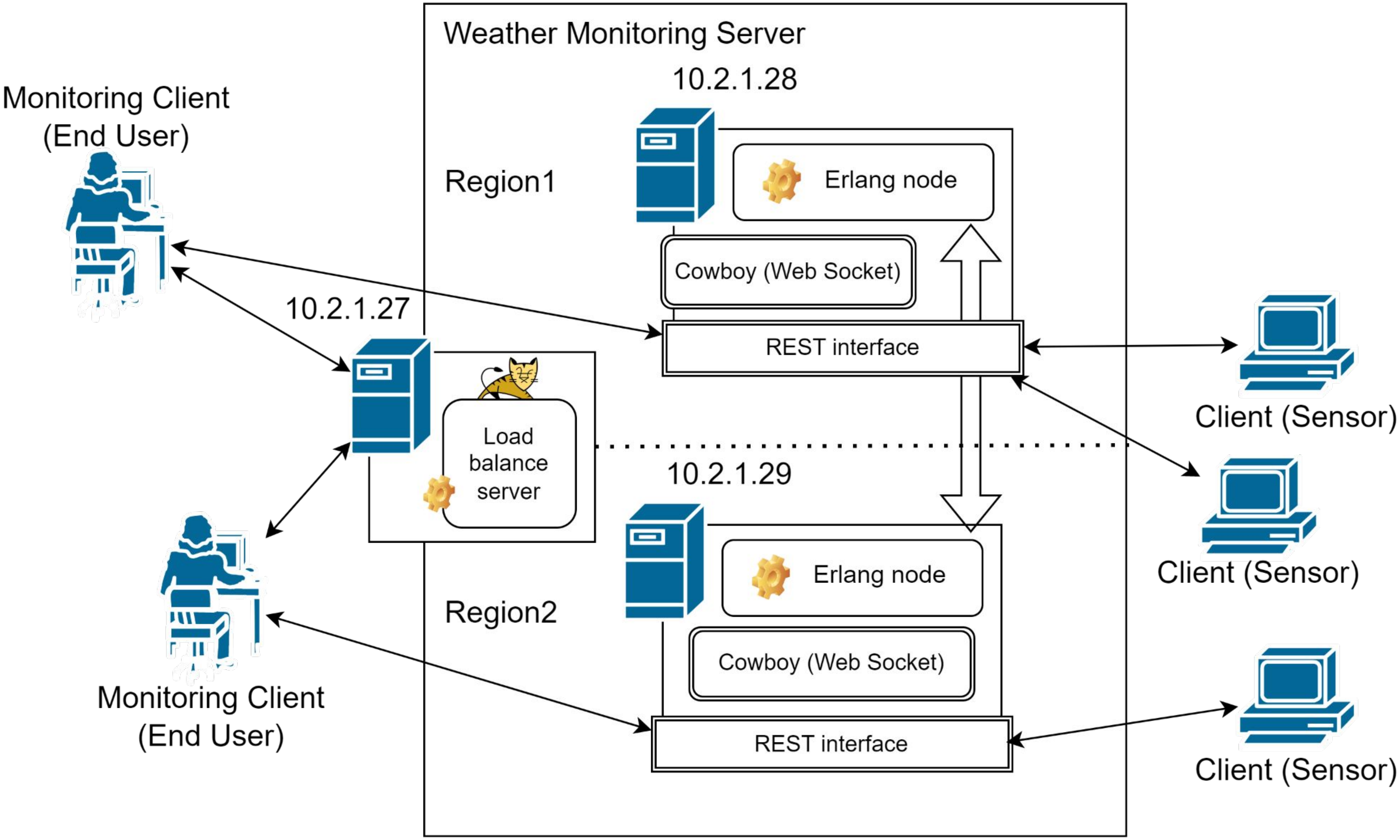
Deployment of a weather monitoring Erlang based server:

1. The servers must expose a REST interface (Clients will be connected using a web sockets).
2. The Monitoring server must provide a Restful API resource for the Sensor nodes to POST the collected data.
3. The Load Balance Server must provide a Restful API resource for the Client nodes to GET the address of an available Monitoring server to connect.
4. The Monitoring server must provide a resource where the Client nodes can GET the data from all propagated Sensors' data (consistency between the server nodes must be kept).
5. The data received by a node will be broadcasted to all other nodes in the monitoring server using the Message passing mechanism available in Erlang (synchronism must be kept in the transactions - a hierarchical server architecture may be used).

Monitoring Client application:

1. A web application will gather the data collected from the server and display it in a number of real-time charts (graphical user interface).

System Architecture



Technology Stack

- Presentation Tier:
  - **ReactJS** : It is an open-source, component-based front-end library responsible only for the view layer of the application.

- Logic Tier:
  - 2 Distributed Erlang server nodes.
  - 1 Tomcat server
  - Websockets API.
  - Restful interface protocol.
  - Cowboy which a small, fast and modern HTTP server for Erlang/OTP.

- Data Tier:
  - According to our functional requirement, we decided that we won’t need a database in this system.