WaCC: Architecture

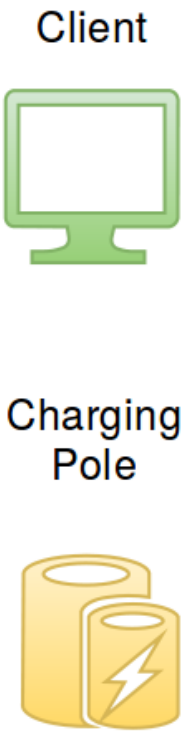
Distributed (smart) electric charging poles

Rick van der Mark & Klaas Kliffen

September 16, 2015

Client types

- Web browsers: requesting information about poles. Where to charge cheapest, fastest, free places etc.
- Electric charging poles: sending heartbeat + charge sessions



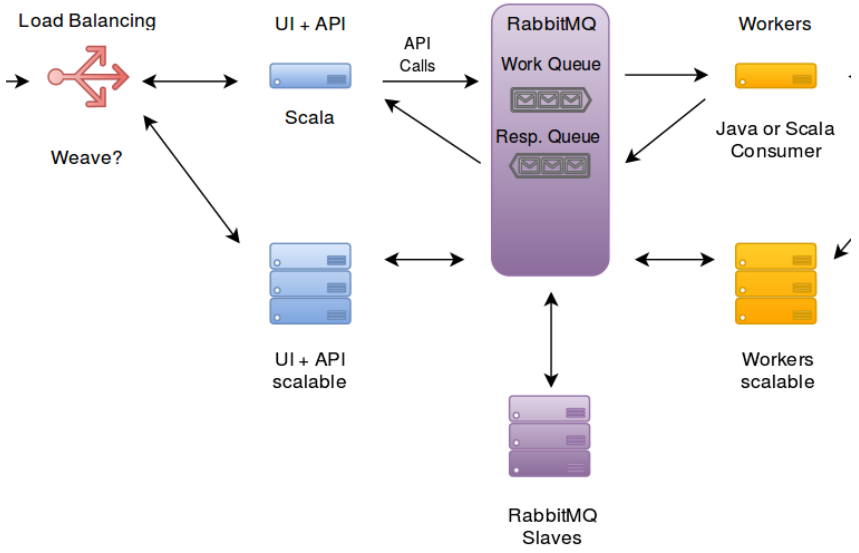
Pole message: data structure

- Is the pole free or charging?
- Last *n* charging sessions
- Per session: id, time, amount, plate number etc.

```
{
    "poleid": "123ABCD",
    "status": "charging",
    "chargingsessions": [
        {
            "sessionid": "ACBD123",
            "starttime": "9-9-2015 15:00",
            "endtime": "9-9-2015 17:30",
            "plate": "KK-RM-15",
            "kwh": 123,
            "price": 12.5
        }
    ]
}
```

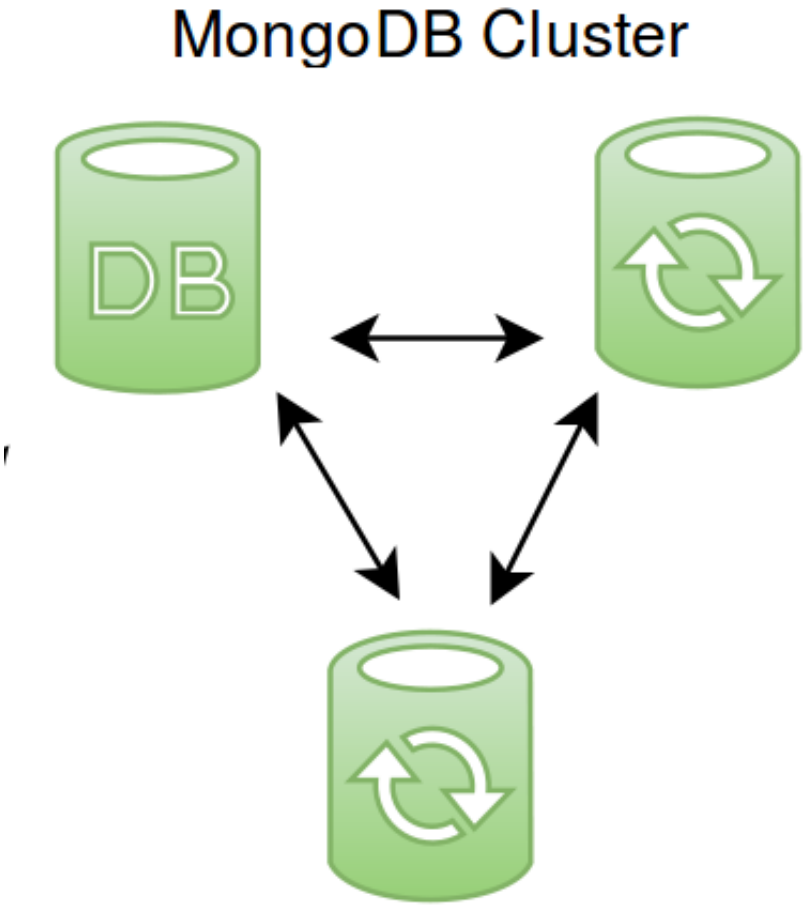
Backend architecture

- Load balancing with weave (or similar)
- UI + API: scala + play + bootstrap
- Workers: java or scala
- Connected by a queue: RabbitMQ
- Can scale with both UI and Worker servers

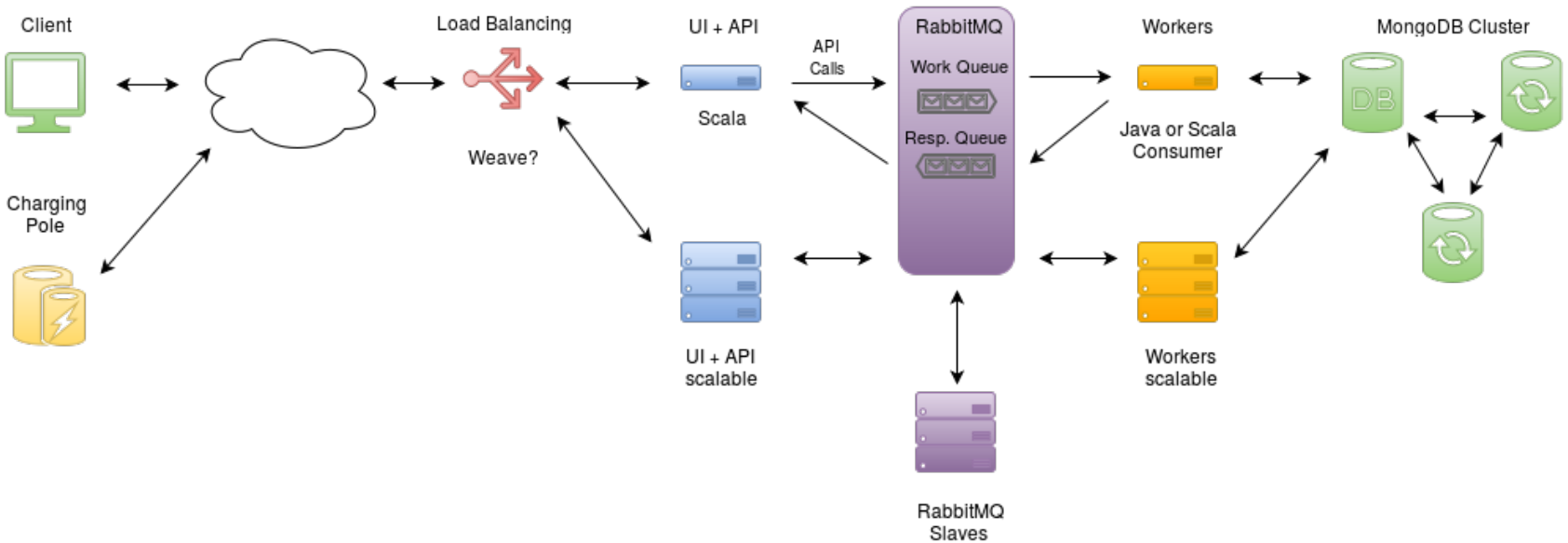


Database

- MongoDB
- Using master slave replication or shard cluster



Overview


Client
Charging Pole
Load Balancing
Weave?
UI + API
Scala
UI + API scalable
API Calls
RabbitMQ
Work Queue
Resp. Queue
RabbitMQ Slaves
Workers
Java or Scala Consumer
Workers scalable
MongoDB Cluster
DB