# Stream processing with Storm

Miquel Sabaté Solà

June 27, 2014

# The problem

- Cities start to embrace technology.
- There's a lot of realtime data to be processed.
- Different sets of data.
- iCity

# The idea

- Build a platform that:
  - Fetches and processes data in realtime.
  - Provides an easy way to extend it.
  - Wraps the iCity API, instead of replacing it.

# Goals

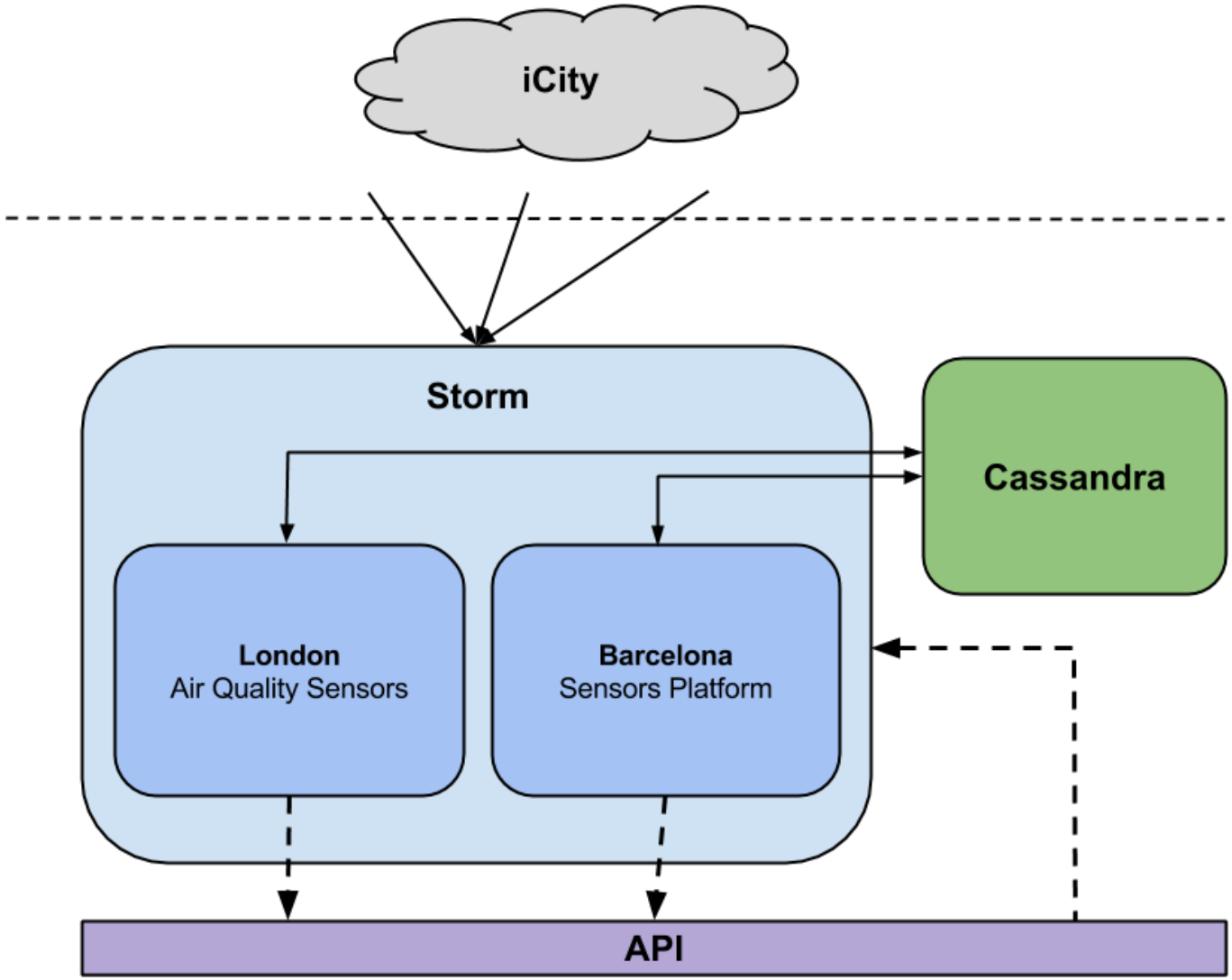
*The* **goal** *of this project is to build a base platform that is able to generate rich information about a set of cities in real time.*

- Design a base platform.
- Design a couple of useful services.
- An ideal cluster.

# Technologies

- Linux.
- Java & Scala.
- Storm.
- Cassandra.
- Go.

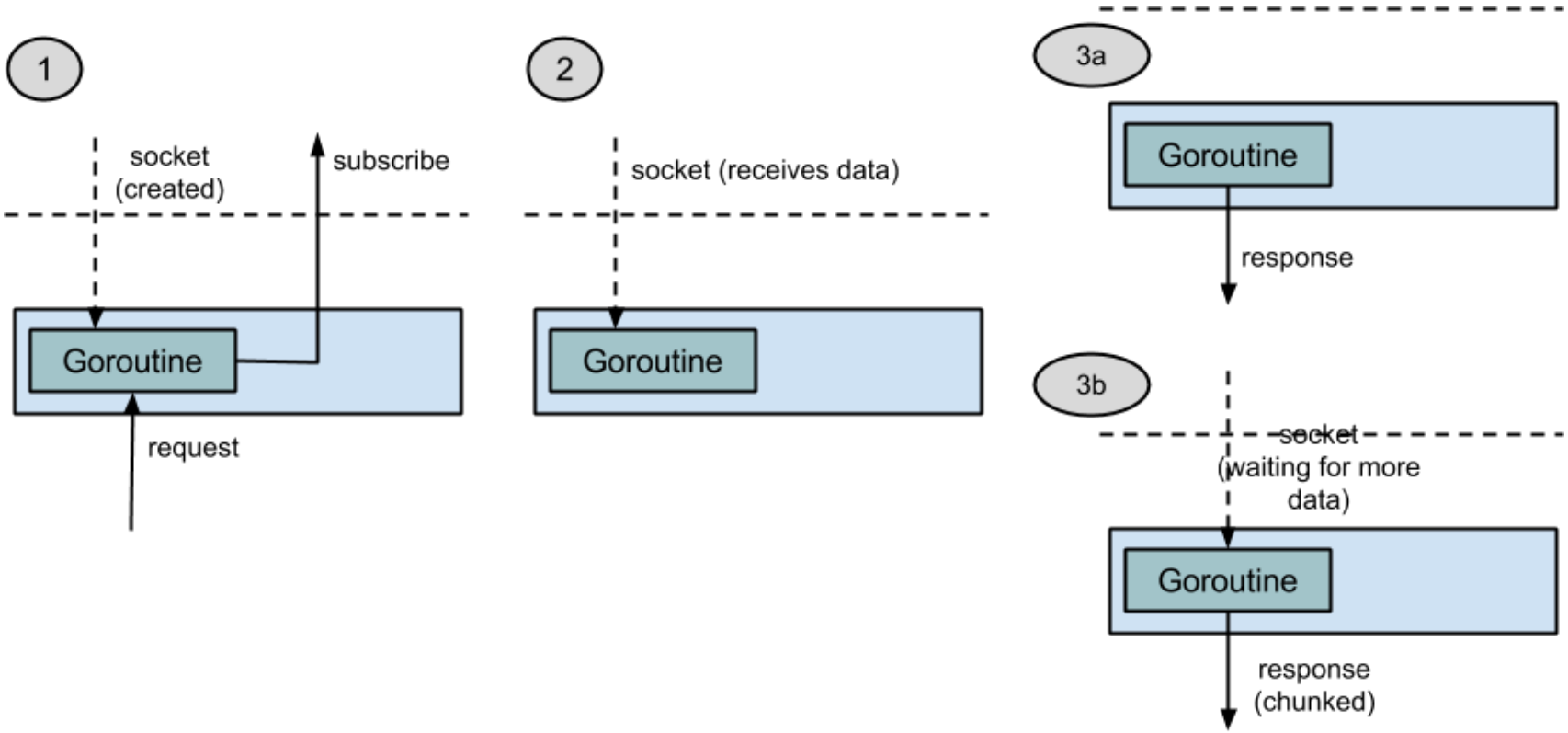
# An overview


iCity
Storm
Cassandra
London
Air Quality Sensors
Barcelona
Sensors Platform
API

# The Storm application

- The com.mssola.snacker.core package.
- The AQS service as a traditional API.
- The BSP as a streaming API.

# API


1
socket
(created)
subscribe
Goroutine
request
2
socket (receives data)
Goroutine
3a
Goroutine
response
3b
socket
(waiting for more
data)
Goroutine
response
(chunked)

# Demo 1

# Demo 2

# Requirements & limits

- Normal execution.
- Benchmark
- Conclusions:

| Component | Minimum | Recommended |
|---|---|---|
| Memory | 900 MB | 2 GB |
| CPU | No minimum | multi-core |
| Disk storage | 2 MB | keep it simple |

# Social & environmental impact

- The burden of maintaining a cluster:
  - Power supply.
  - Maintaining a cooling system.
  - Building the cluster.
- Social impact:
  - Local economy.
  - How citizens interact.

# Conclusions

- Meeting the expectations.
- The future.

# Questions