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

Big data

- We have a problem with data:
  - There is a lot of data to process.
  - There are different sets of data.
- MapReduce.
- Hadoop and batch processing.
- **Storm** and stream processing.

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

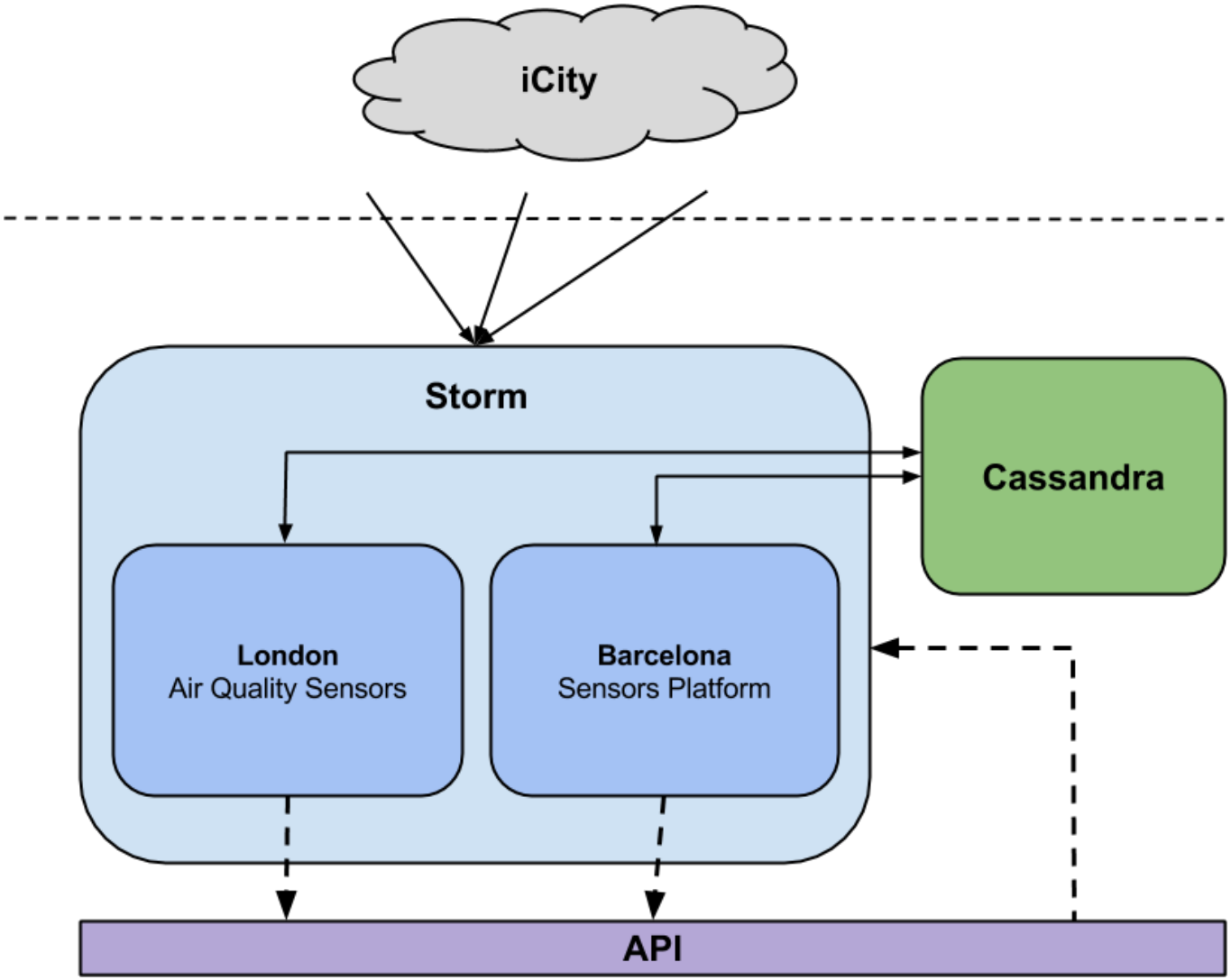
The **goal** *of this project is to build a base platform that is able to generate rich information about a set of cities in real time.*

- Design a base **platform**.
- Design a couple of useful **services**.
- Figure out the **hardware** needed to run all of this.

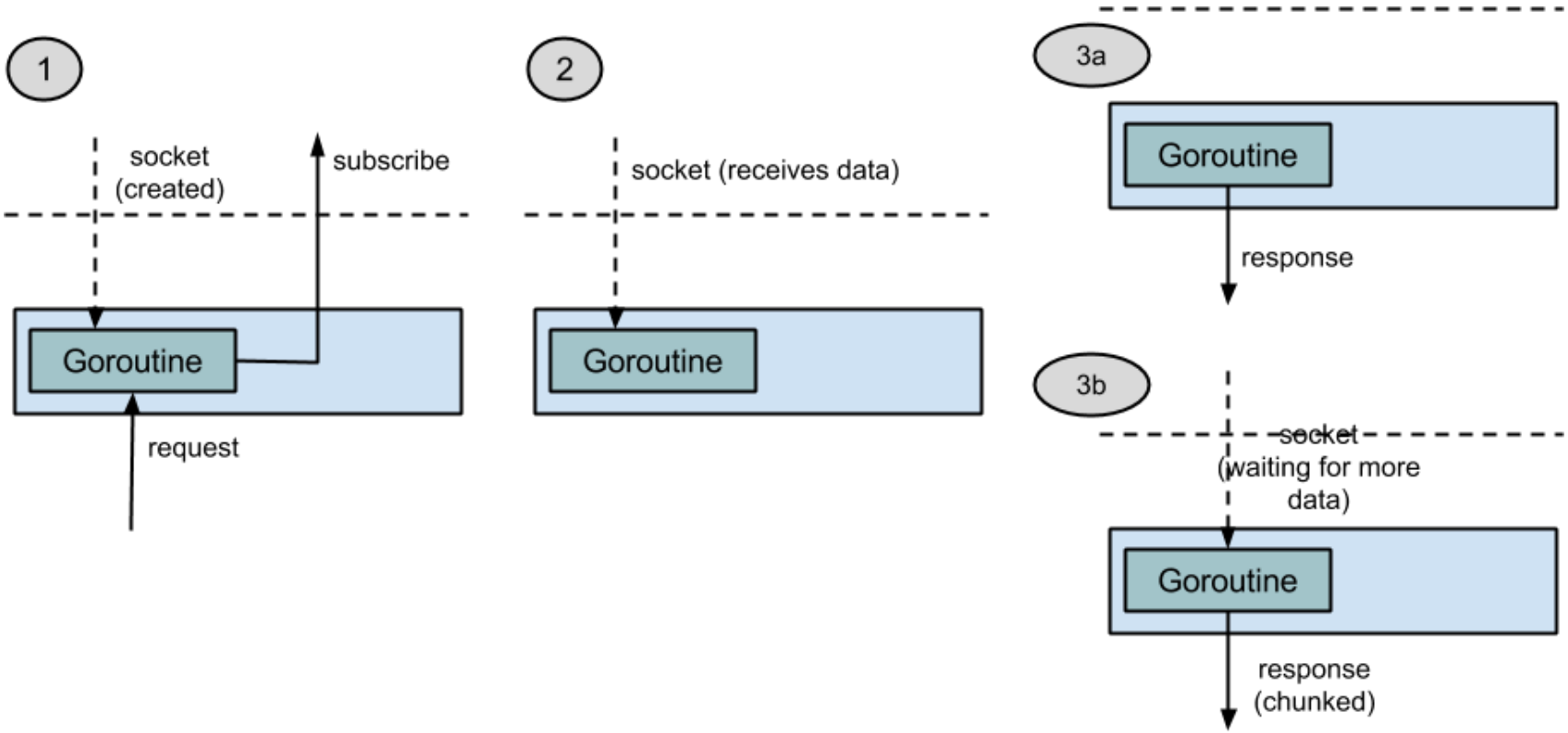
Technologies

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

An overview


iCity
Storm
Cassandra
London
Air Quality Sensors
Barcelona
Sensors Platform
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

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.

API

	February		March				April				May				June			
	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Planning																		
Scope																		
Planning																		
Budget																		
Prelim. presentation																		
Bibliography																		
List of conditions																		
Final presentation and document																		
Analysis and design																		
Requirements and features																		
Design																		
Development of the core																		
Initial implementation																		
Full implementation																		
Tests & documentation																		
Providing services																		
Thinking on services																		
Implementation of services																		
Tests & documentation																		
Designing the cluster																		
Requirements																		
Design																		
Building a base cluster																		
Concluding																		
Merging software and hardware																		
Testing & documentation																		
Final Stage																		
Documentation																		
Final report																		
Final presentation																		

Conclusions

- Meeting the expectations.
- The future.

Questions