

ARID - Fogxy Pattern Evaluation

July 24, 2017

Andreas Seitz
seitz@in.tum.de

Technical University of Munich
Chair for Applied Software Engineering
Department of Informatics

Felix Thiele
felix.thiele@tum.de

Technical University of Munich
Chair for Applied Software Engineering
Department of Informatics

Active Reviews for Intermediate Designs

- Lightweight method introduced by Clements et al. in 2000
- Evaluate small intermediate designs
- Research question: is ARID feasible to evaluate architectural patterns as well?
- Nine steps divided into two phases:
 - Rehearsal
 - Review

Rehearsal

- Identify the reviewers
- Prepare the design briefing
- Prepare seed scenarios
- Prepare materials



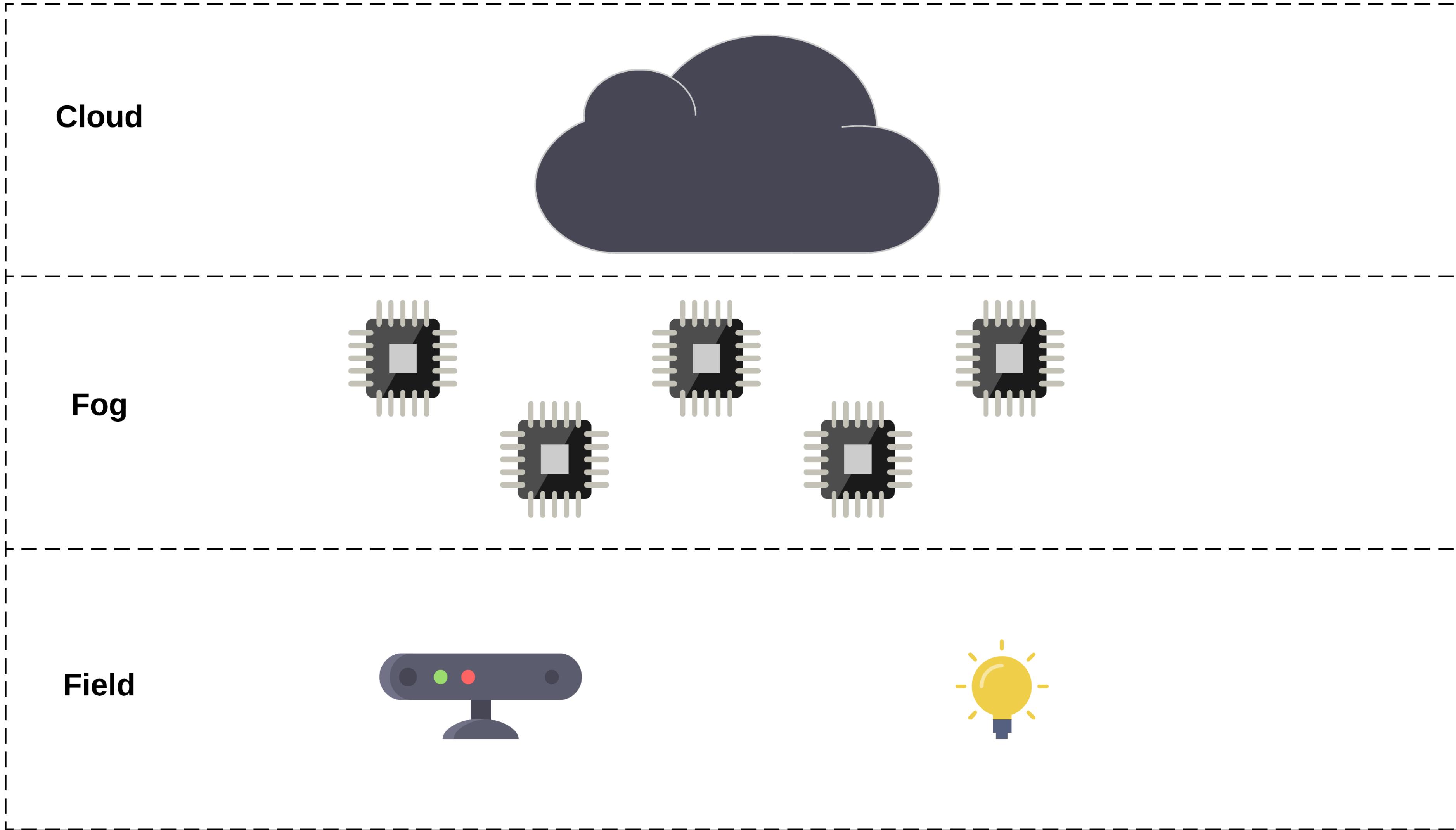
Review

- Present ARID
- Present Fogxy
- Brainstorm and prioritize scenarios
- Apply scenarios
- Summary

Fog Computing

- The cloud is no satisfactory deployment environment for latency sensitive applications
- Fog computing introduces another layer in between the cloud and the field / clients - the **fog**
- Fog nodes are located in close proximity to the clients and thus latency is greatly reduced

Overview





Fogxy

- Architectural pattern for fog computing presented by A. Seitz
- Aims at satisfying following quality attributes:
 - Real-Time
 - Availability / Reliability

Real Time

- a problem, system, or application that is **concurrent** and has **timing constraints** whereby incoming events must be processed within a given timeframe
- pertaining to a system or mode of operation in which computation is performed during the actual time that an external process occurs, in order that the computation results can be used to control, monitor, or respond in a **timely manner** to the external process

(taken from ISO/IEC/IEEE 24765:2010(E))

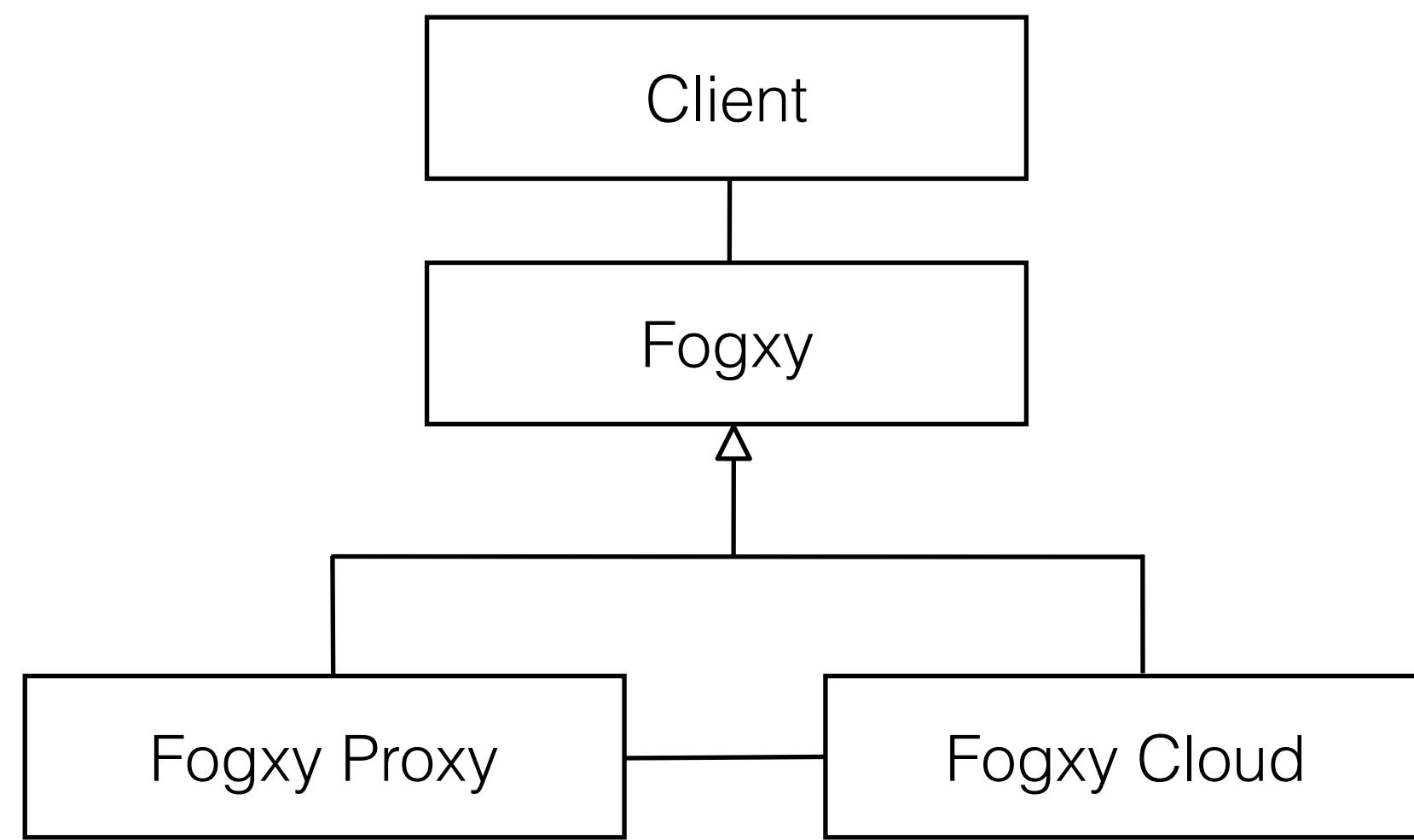
Availability / Reliability

- the degree to which a system or component is **operational and accessible** when required for use
- **ability** of a component or service to **perform its required function** at a stated instant or over a stated period of time

(taken from ISO/IEC/IEEE 24765:2010(E))



Class View

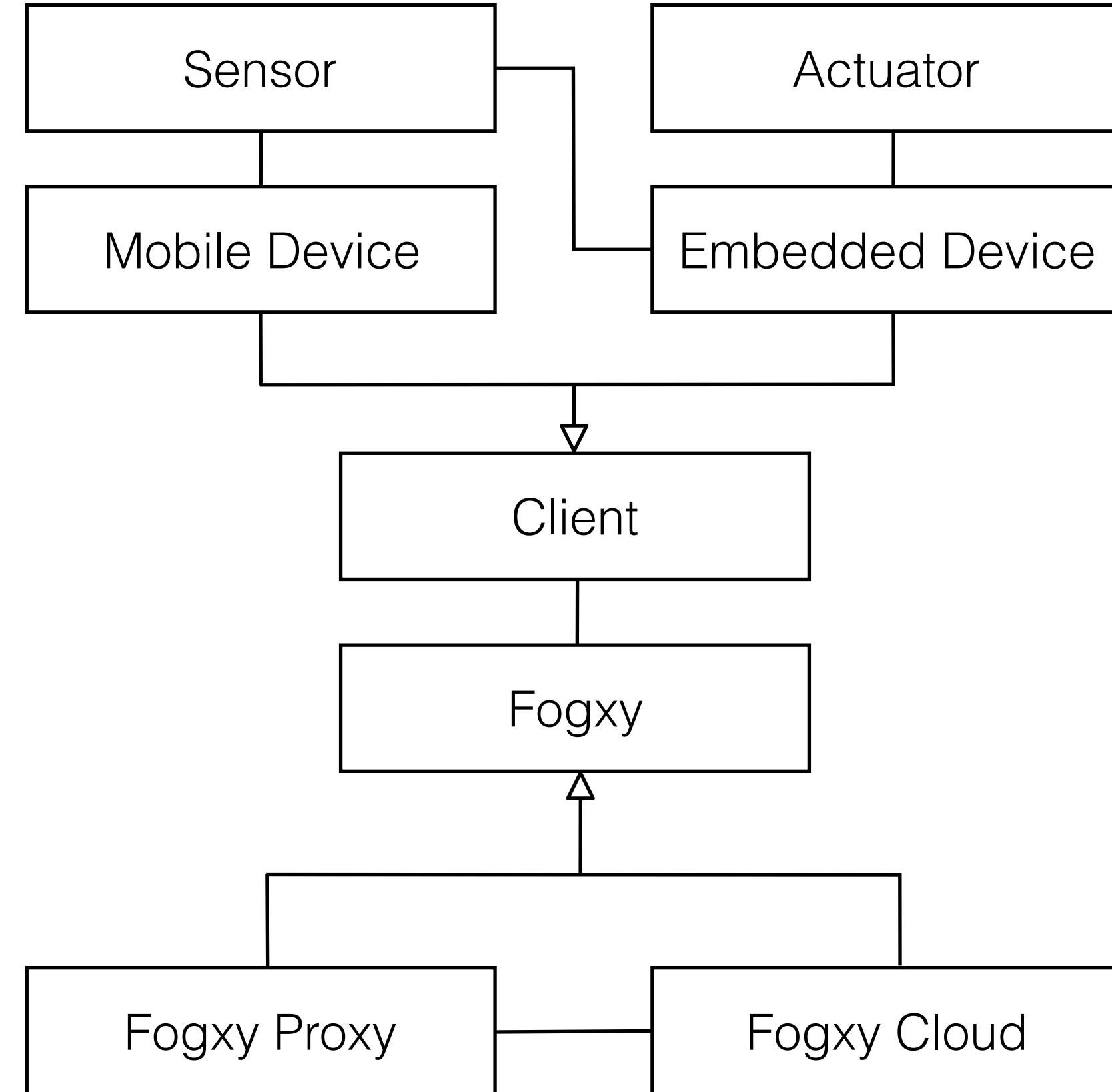




Class View

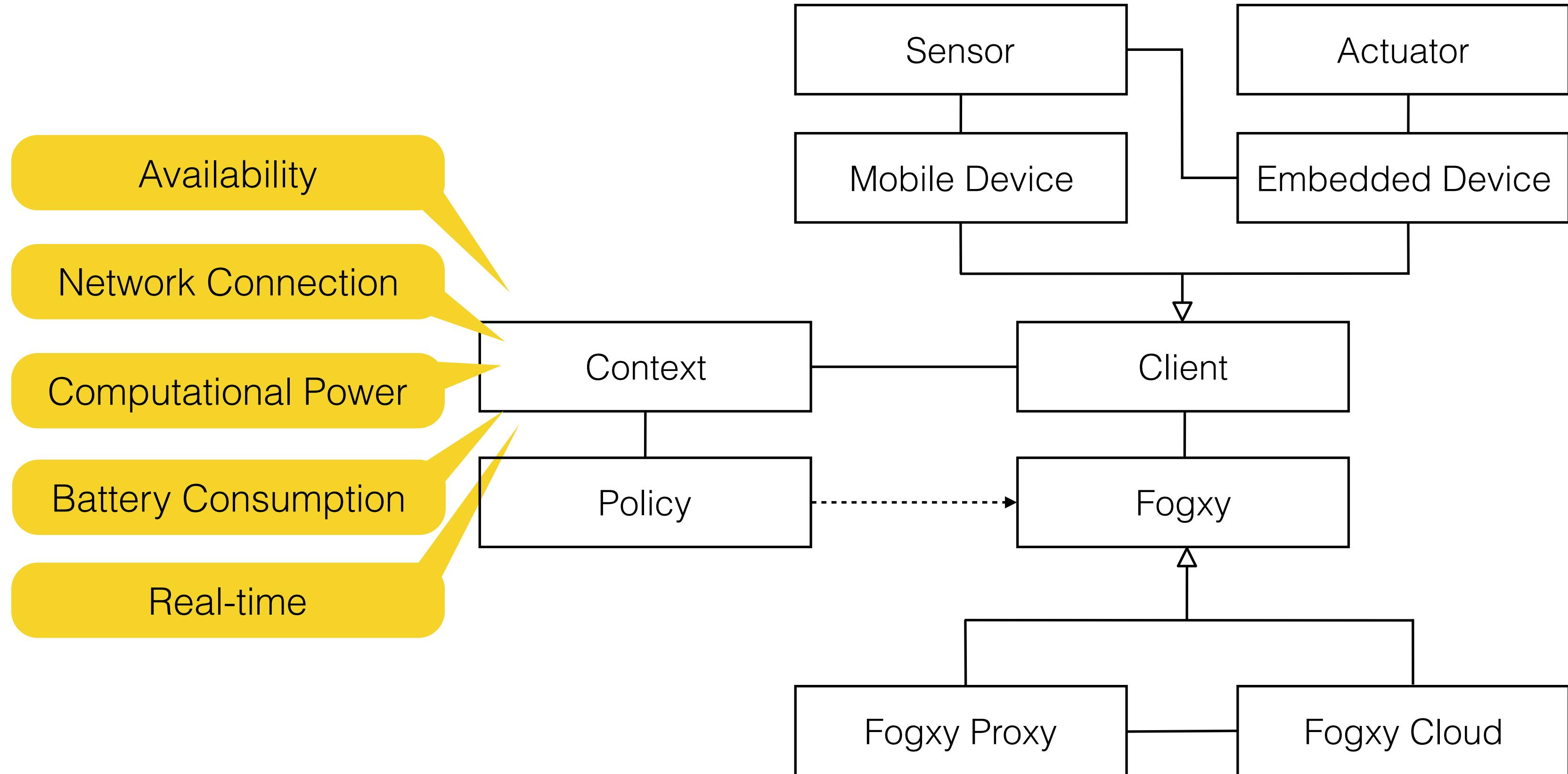
Internet of Things

Cyber-Physical Systems



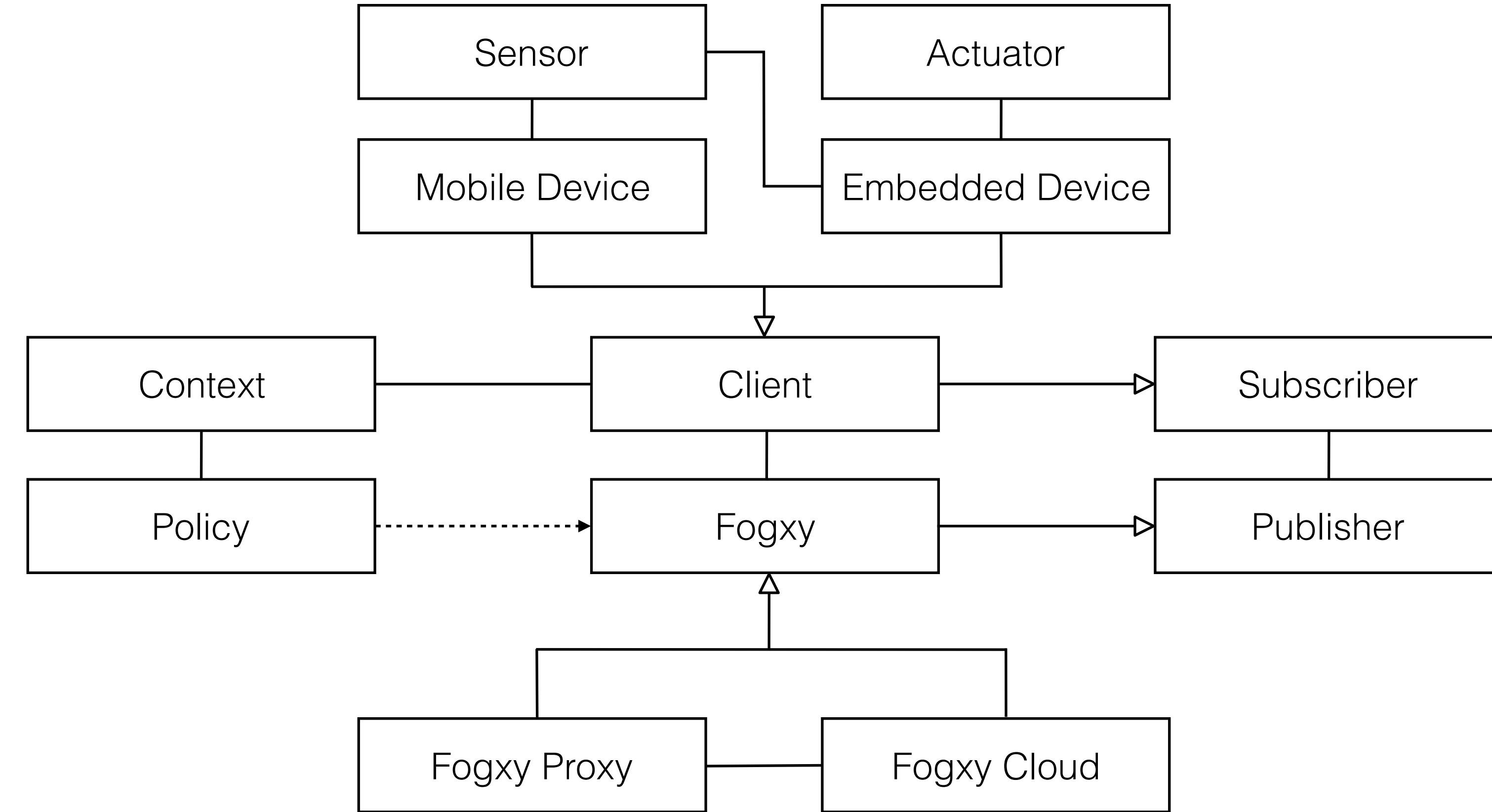


Class View



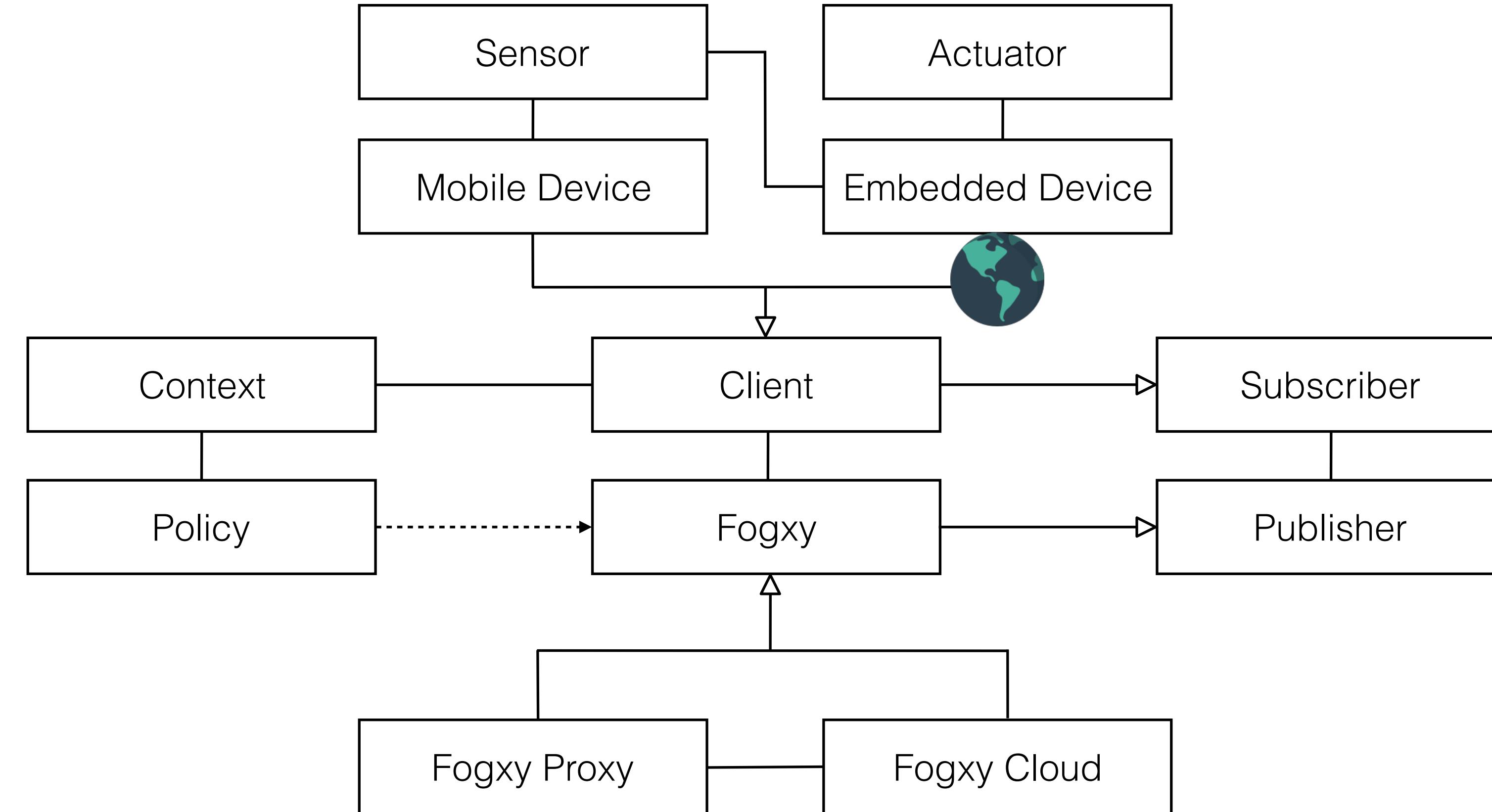


Class View



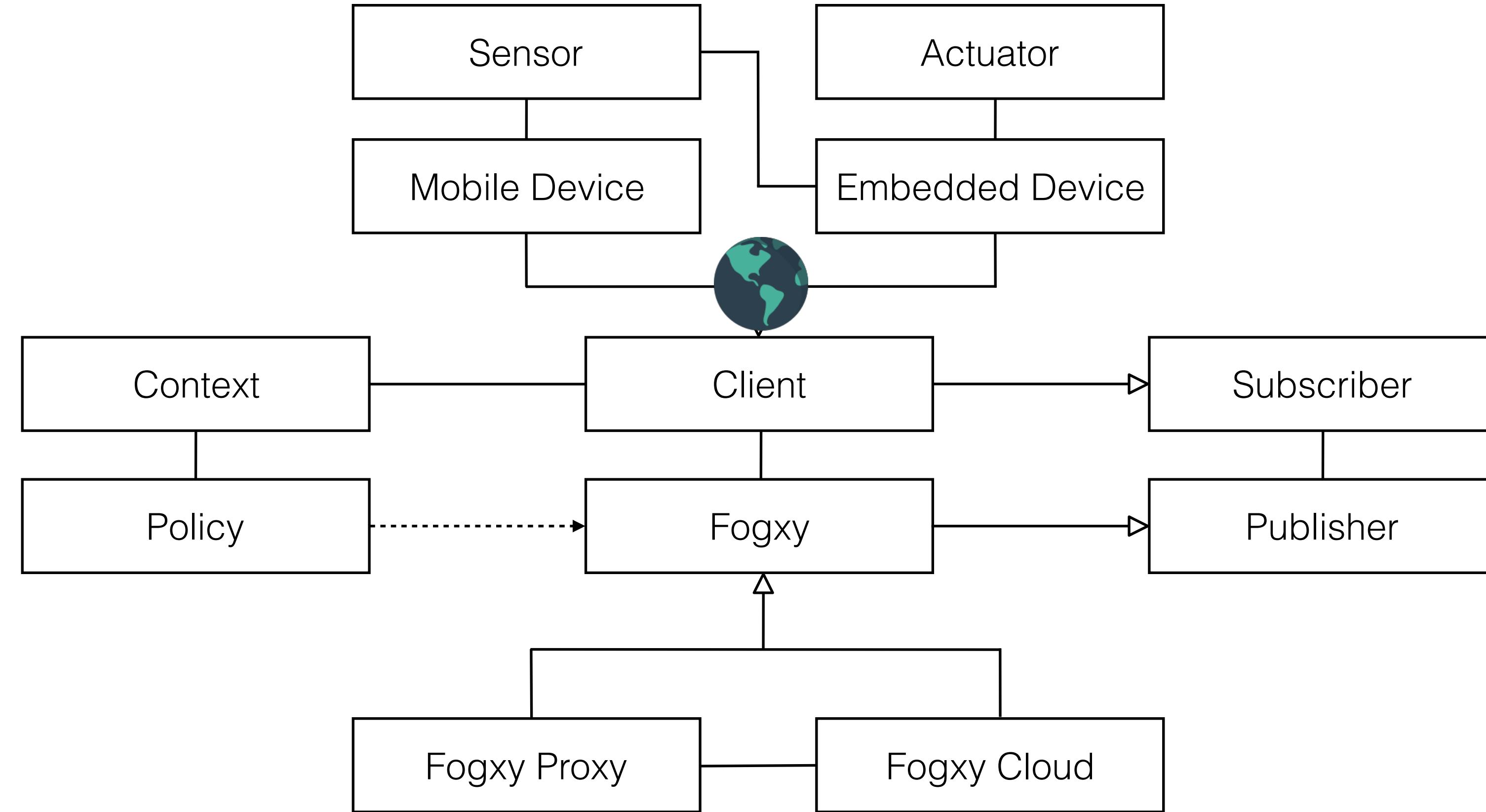


Dynamic View



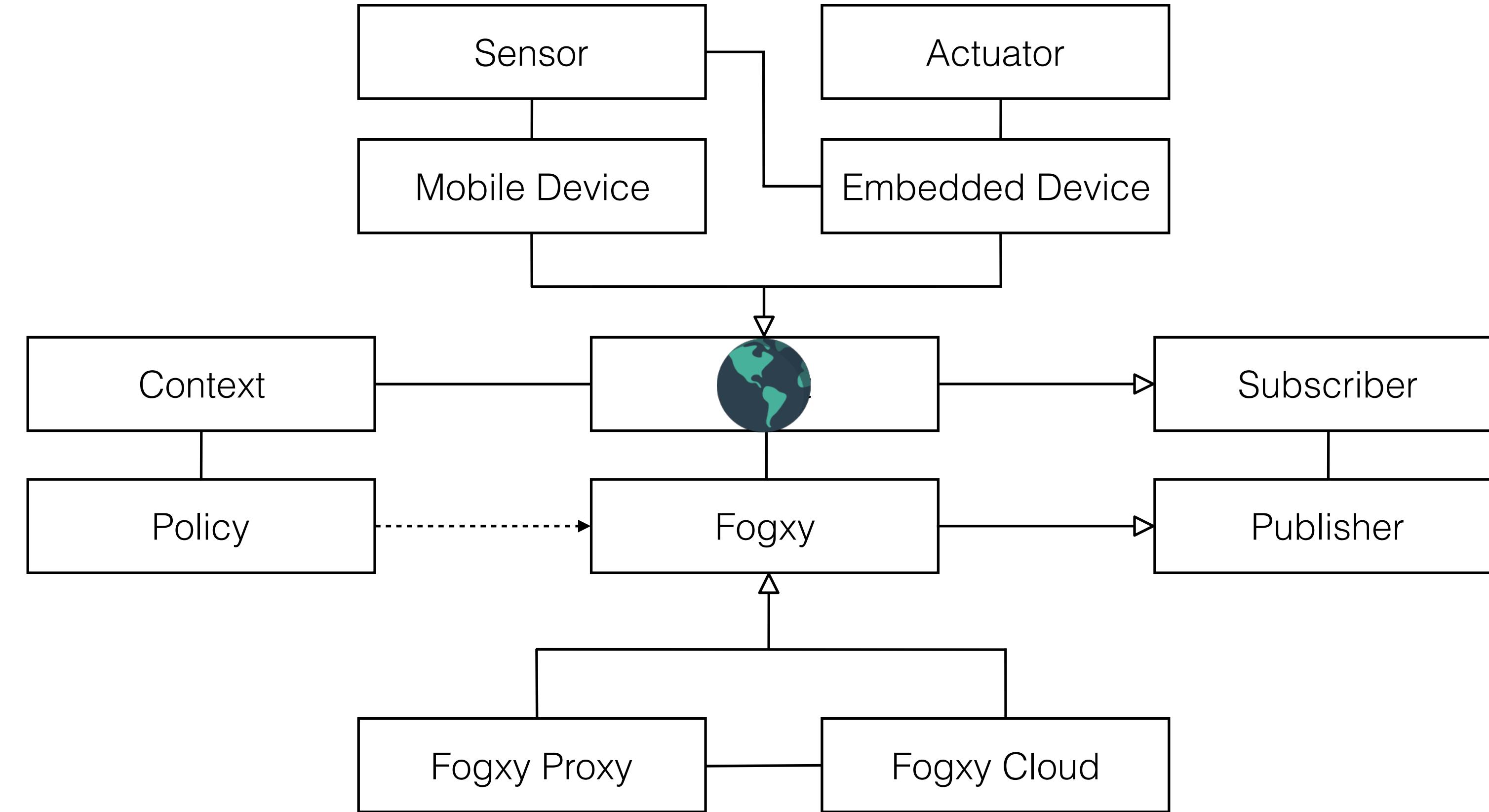


Dynamic View



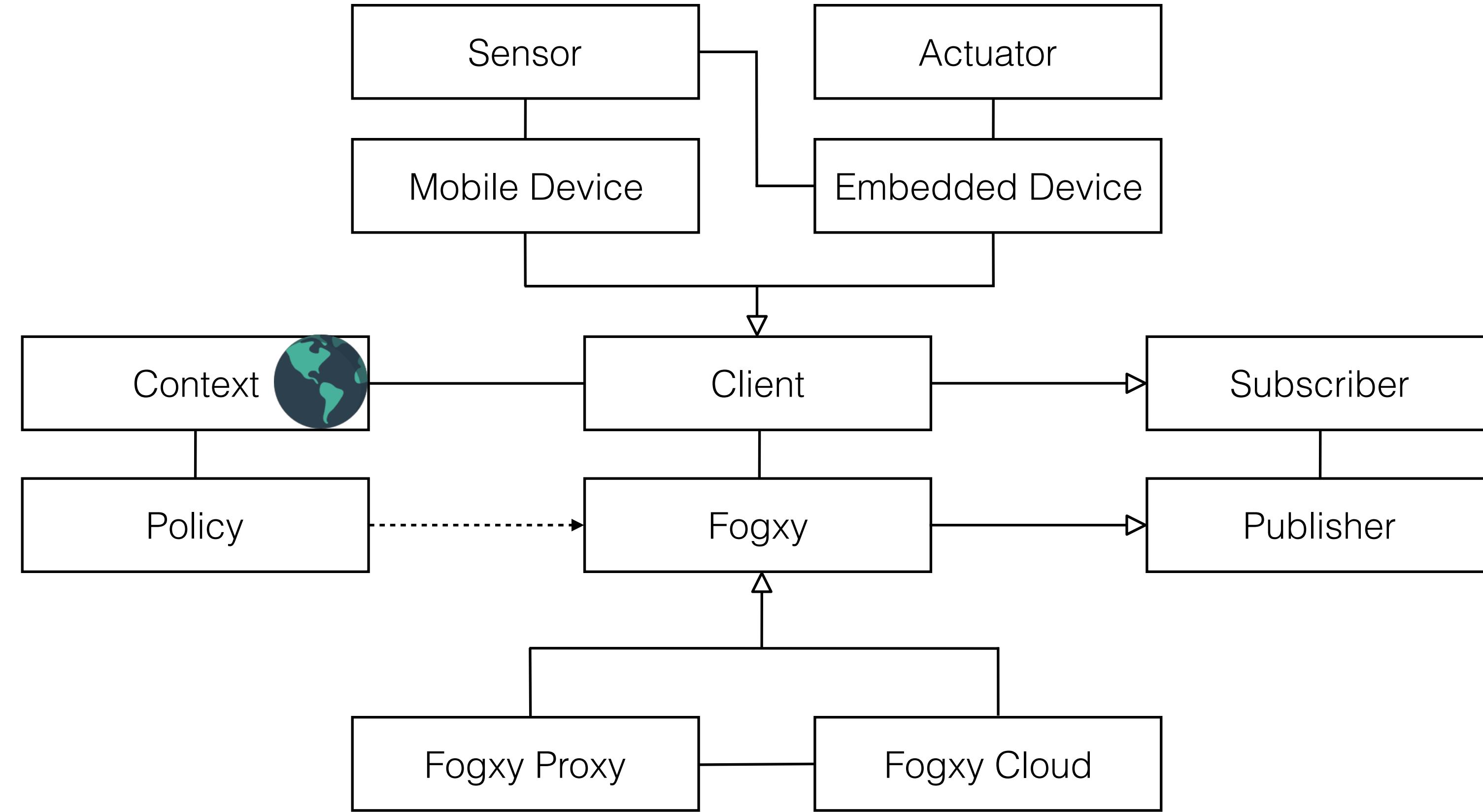


Dynamic View



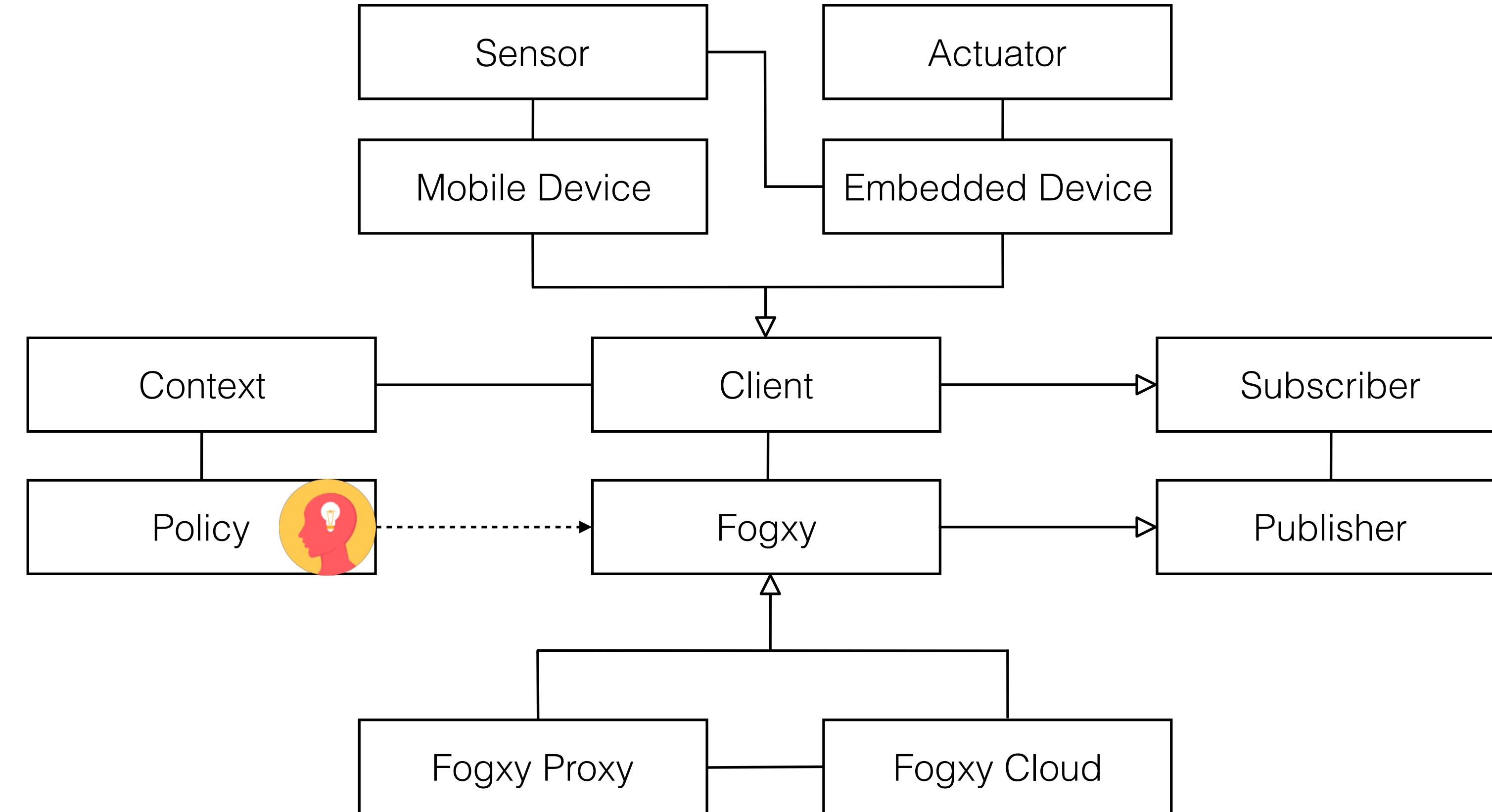


Dynamic View



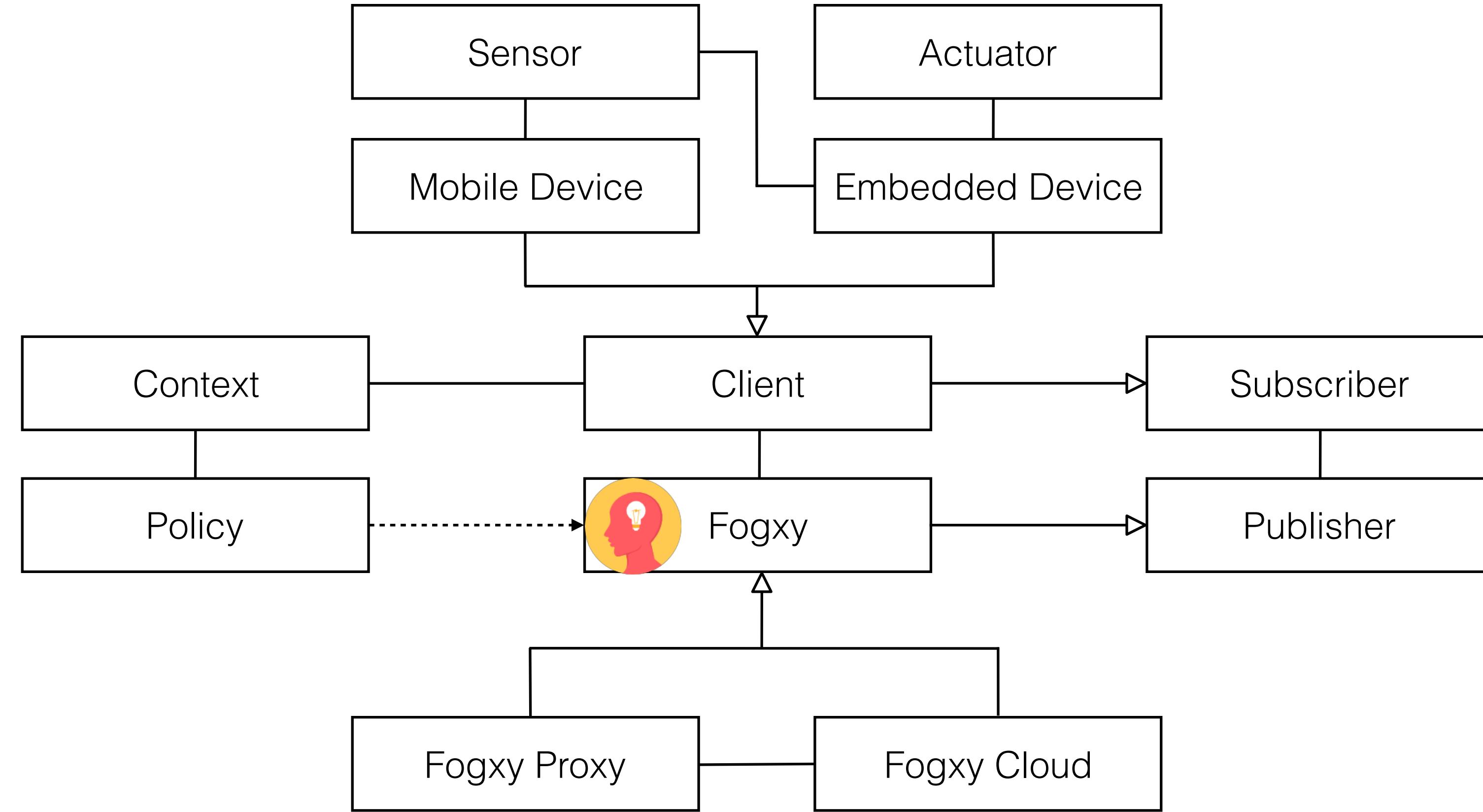


Dynamic View



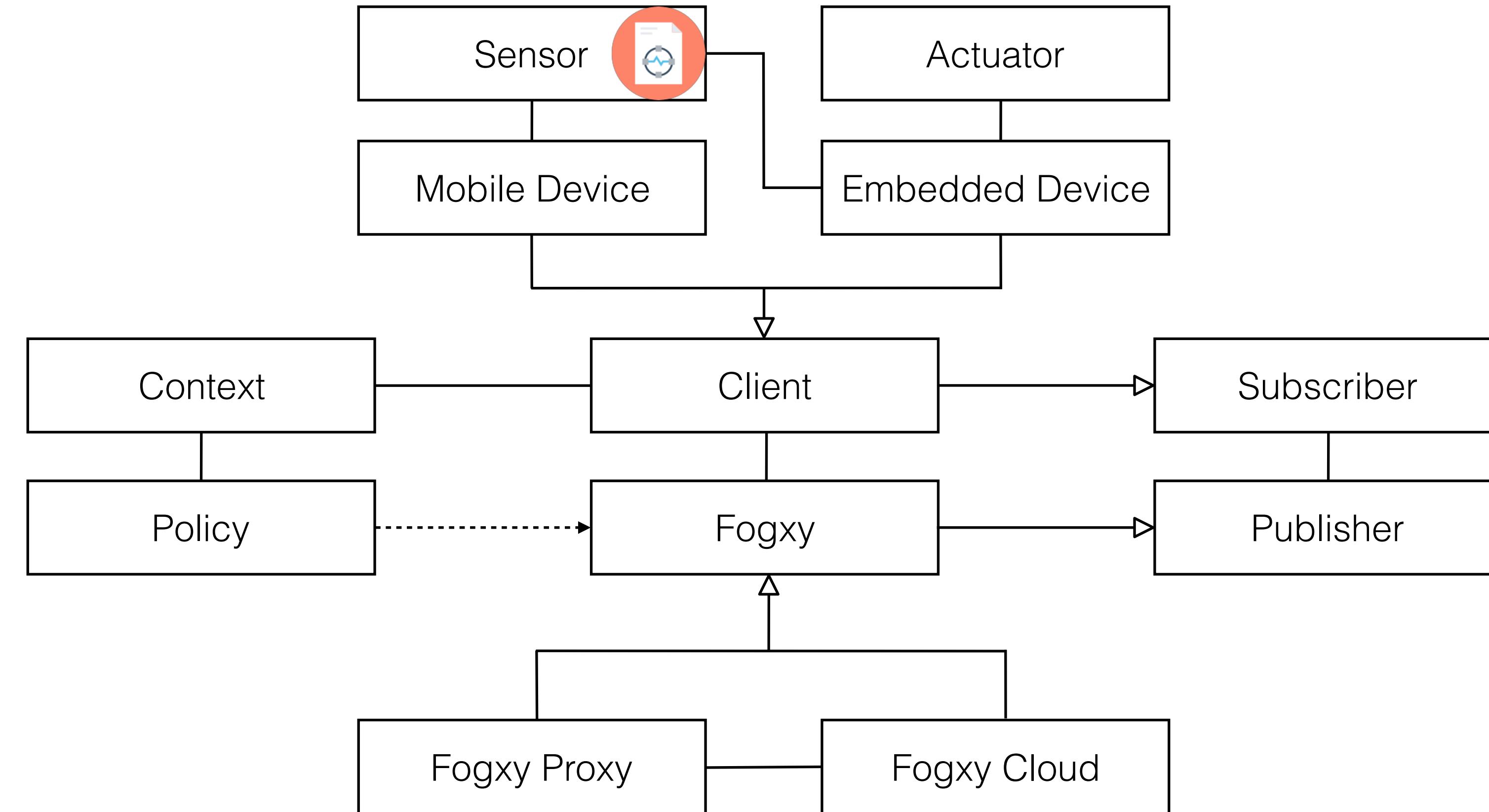


Dynamic View



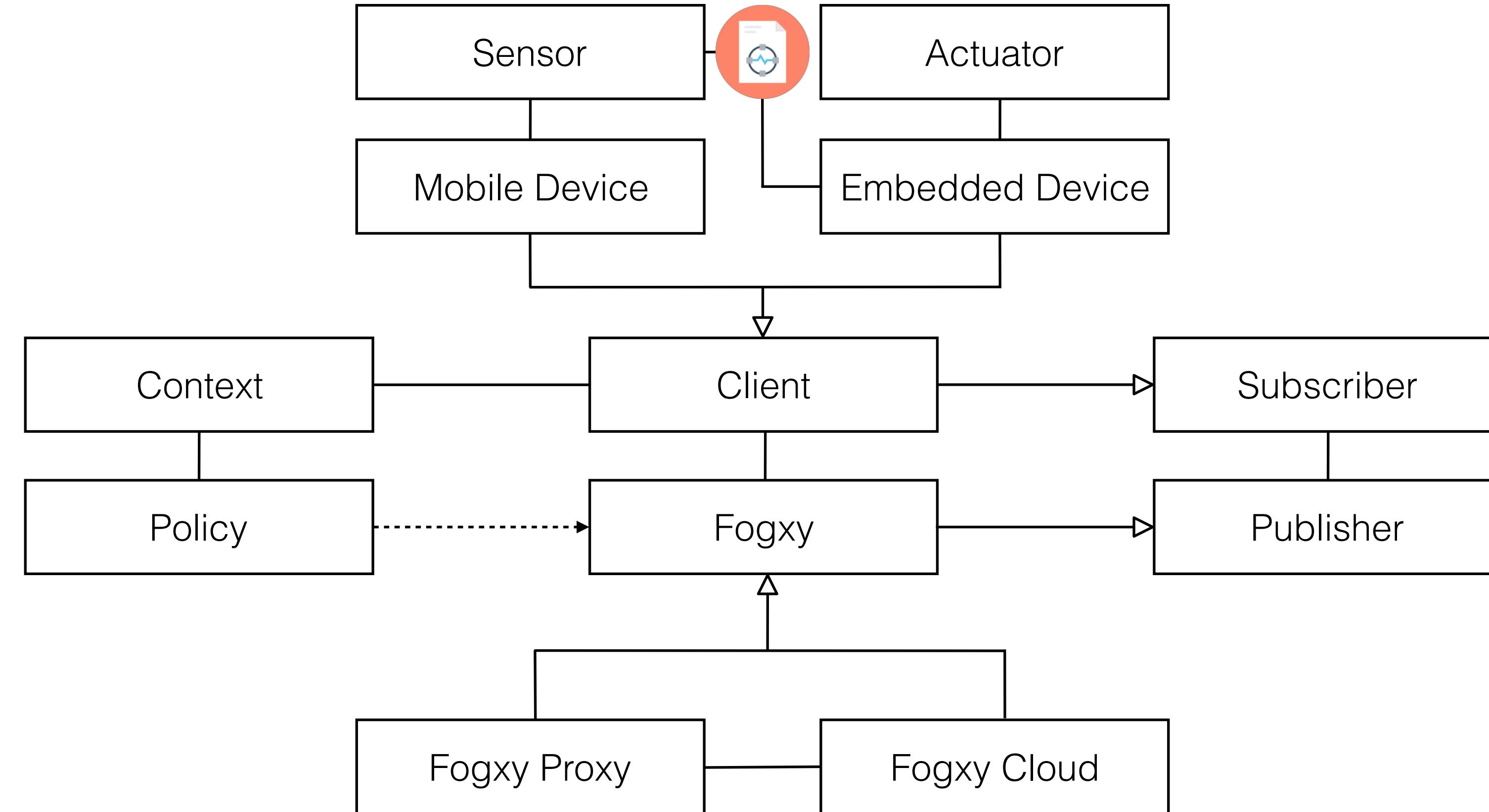


Dynamic View



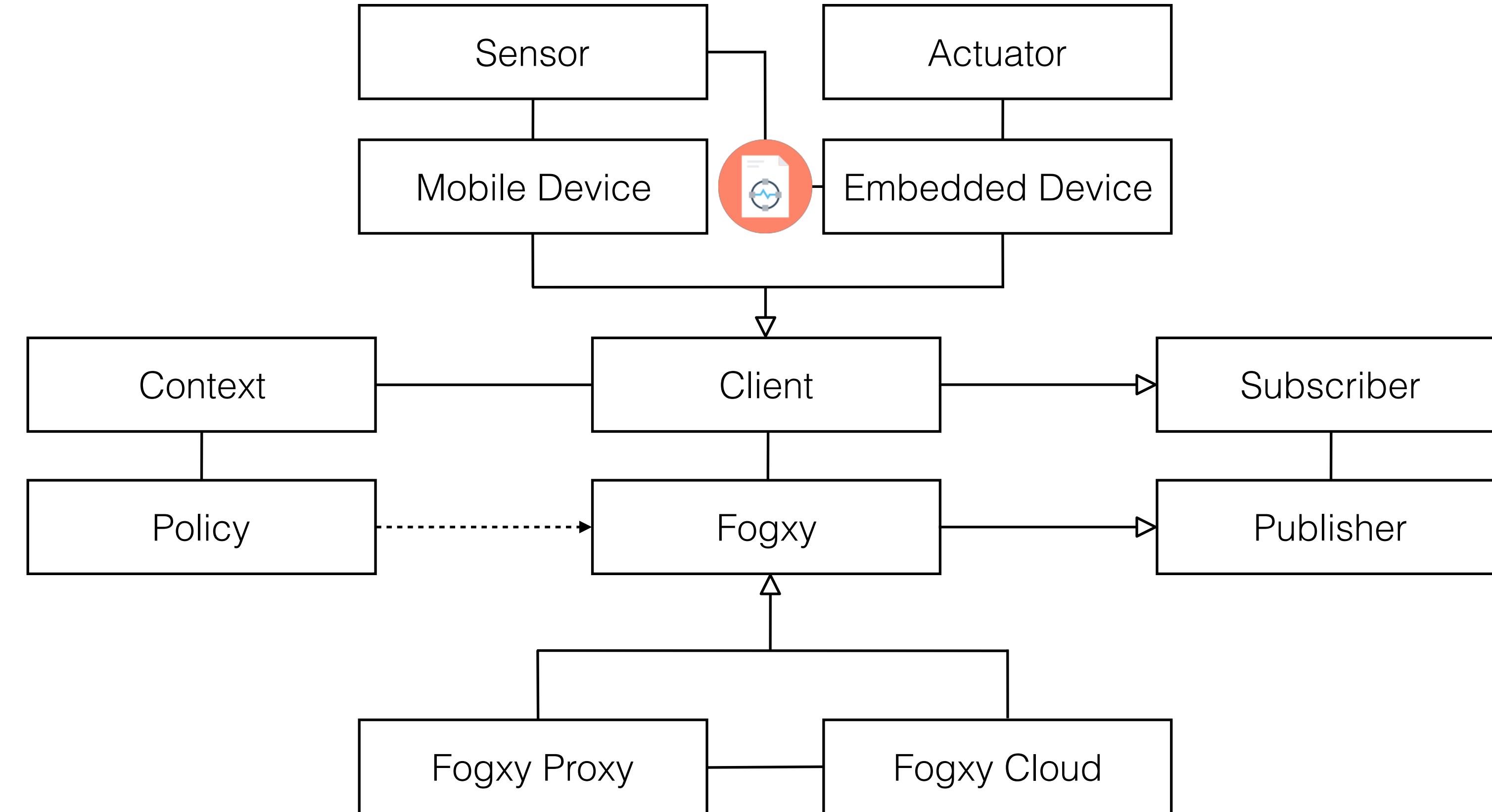


Dynamic View



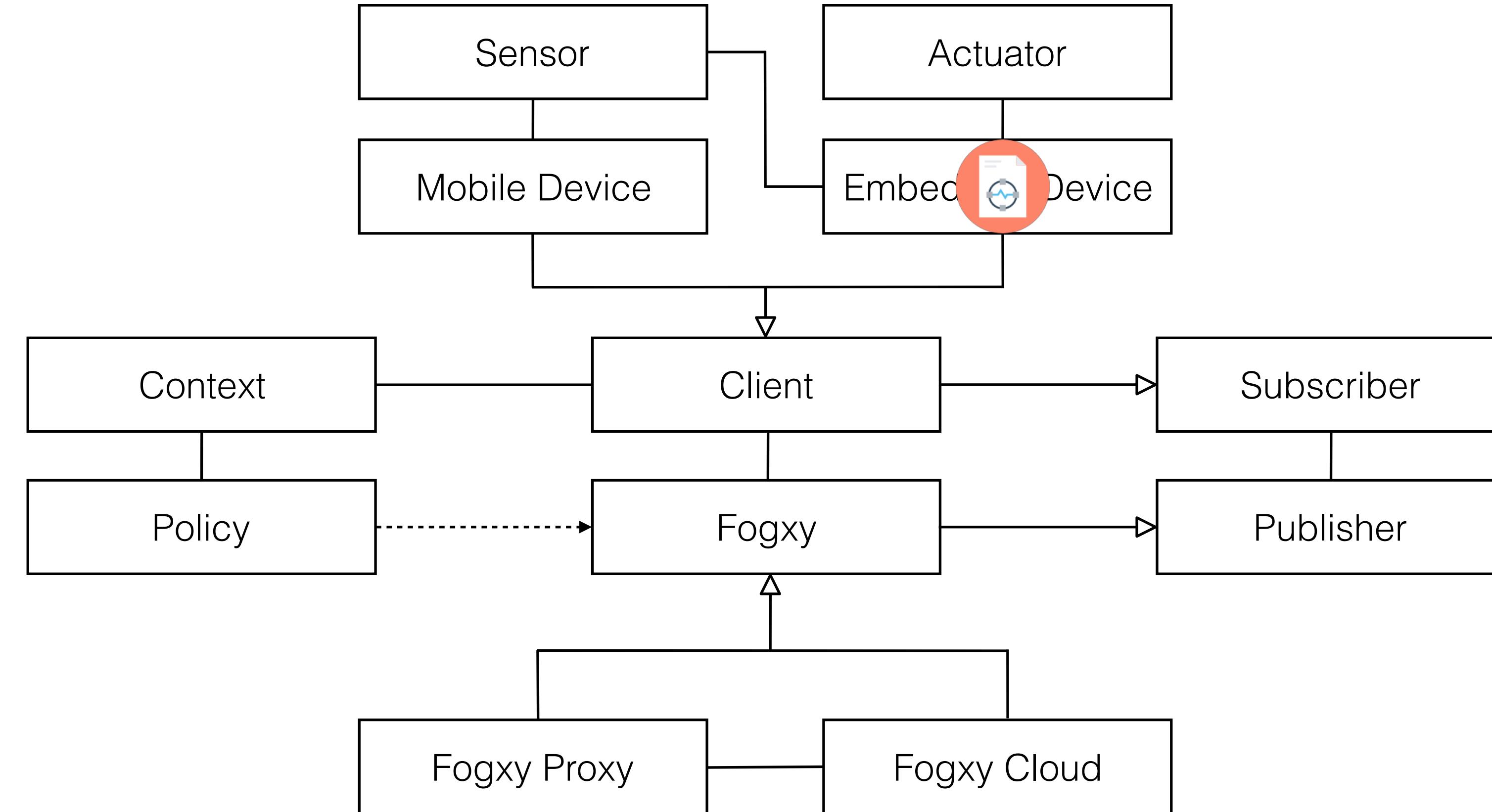


Dynamic View



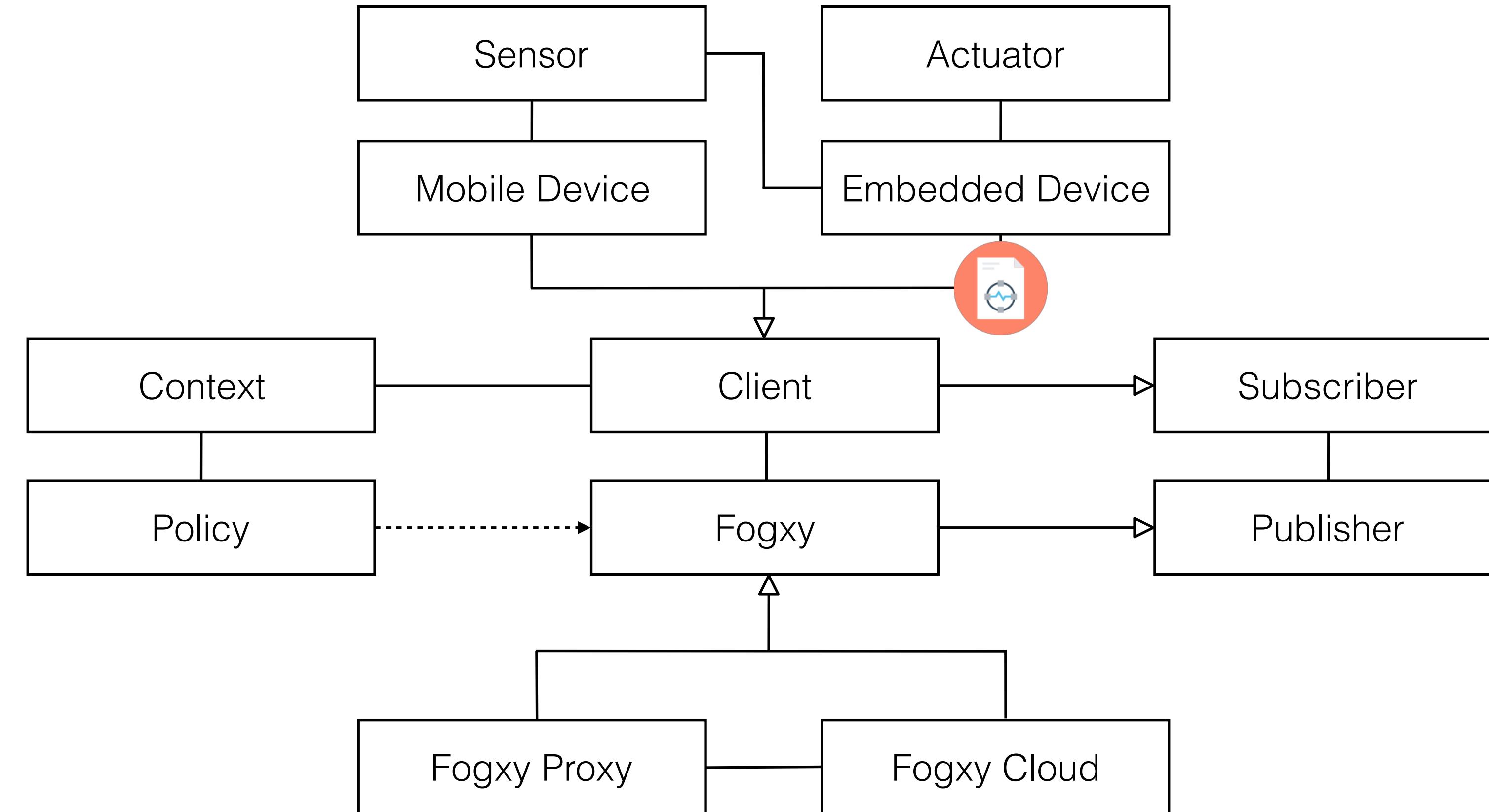


Dynamic View



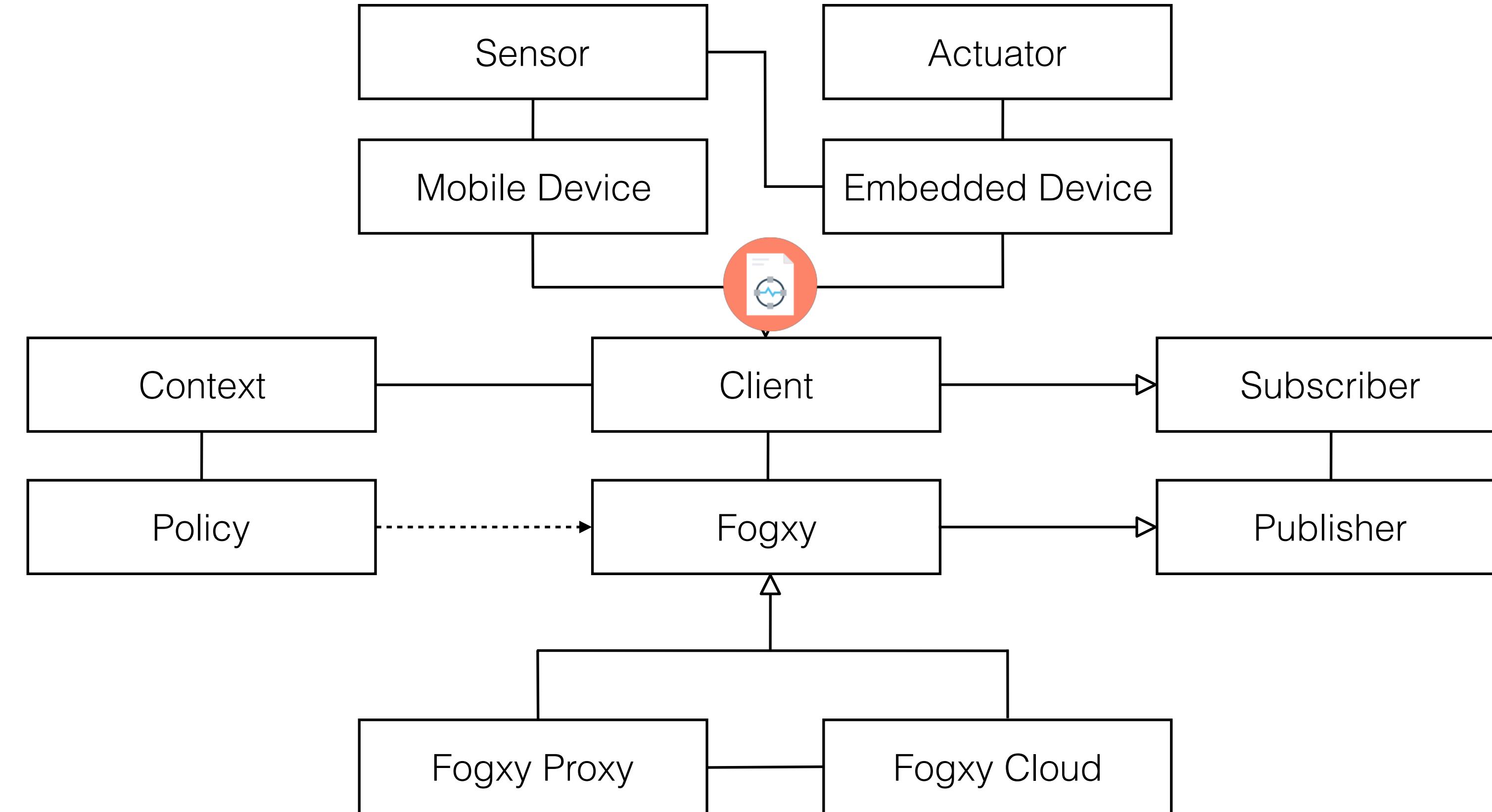


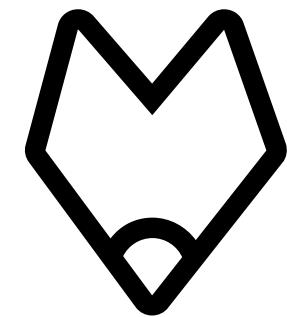
Dynamic View



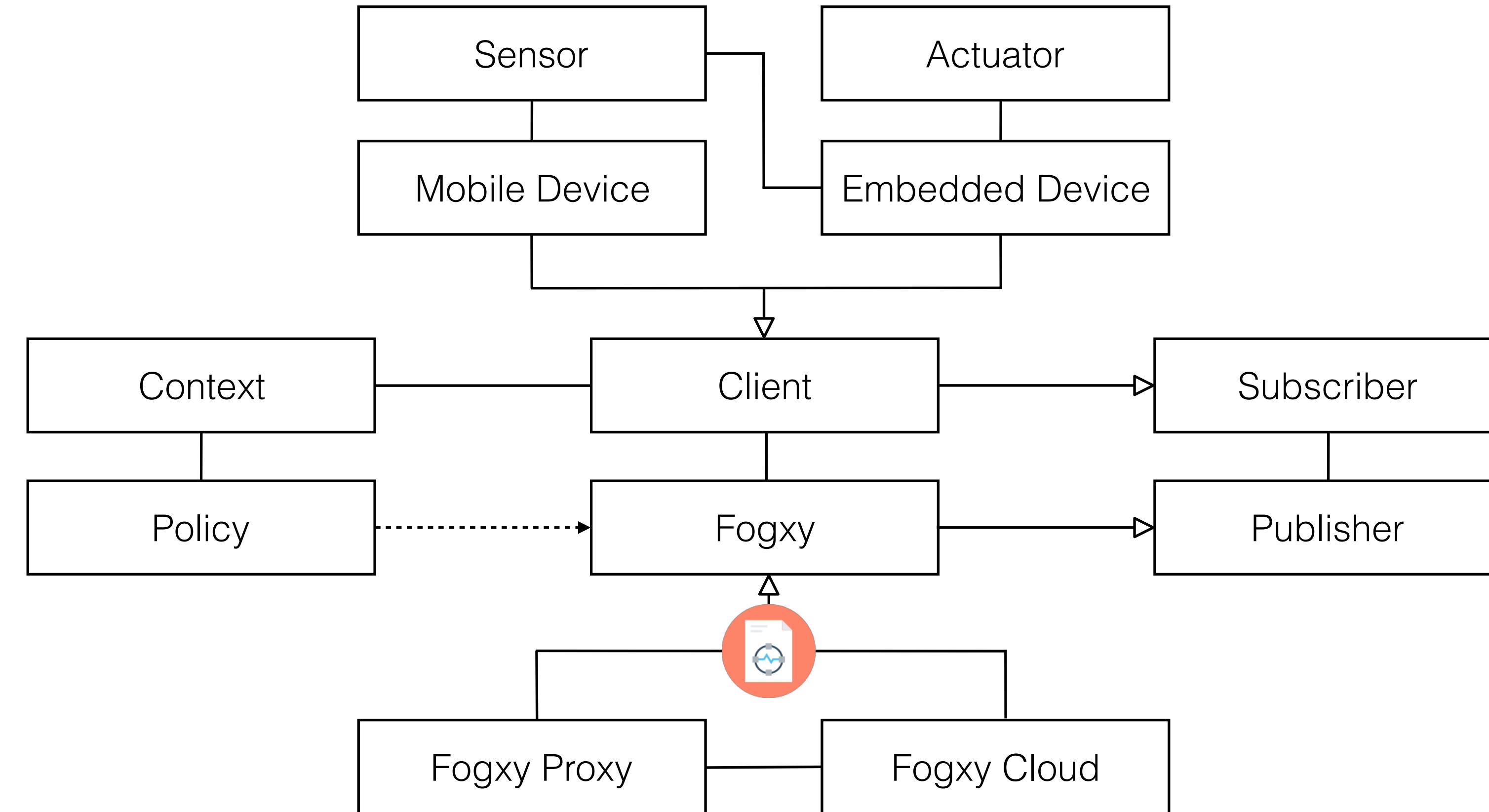


Dynamic View



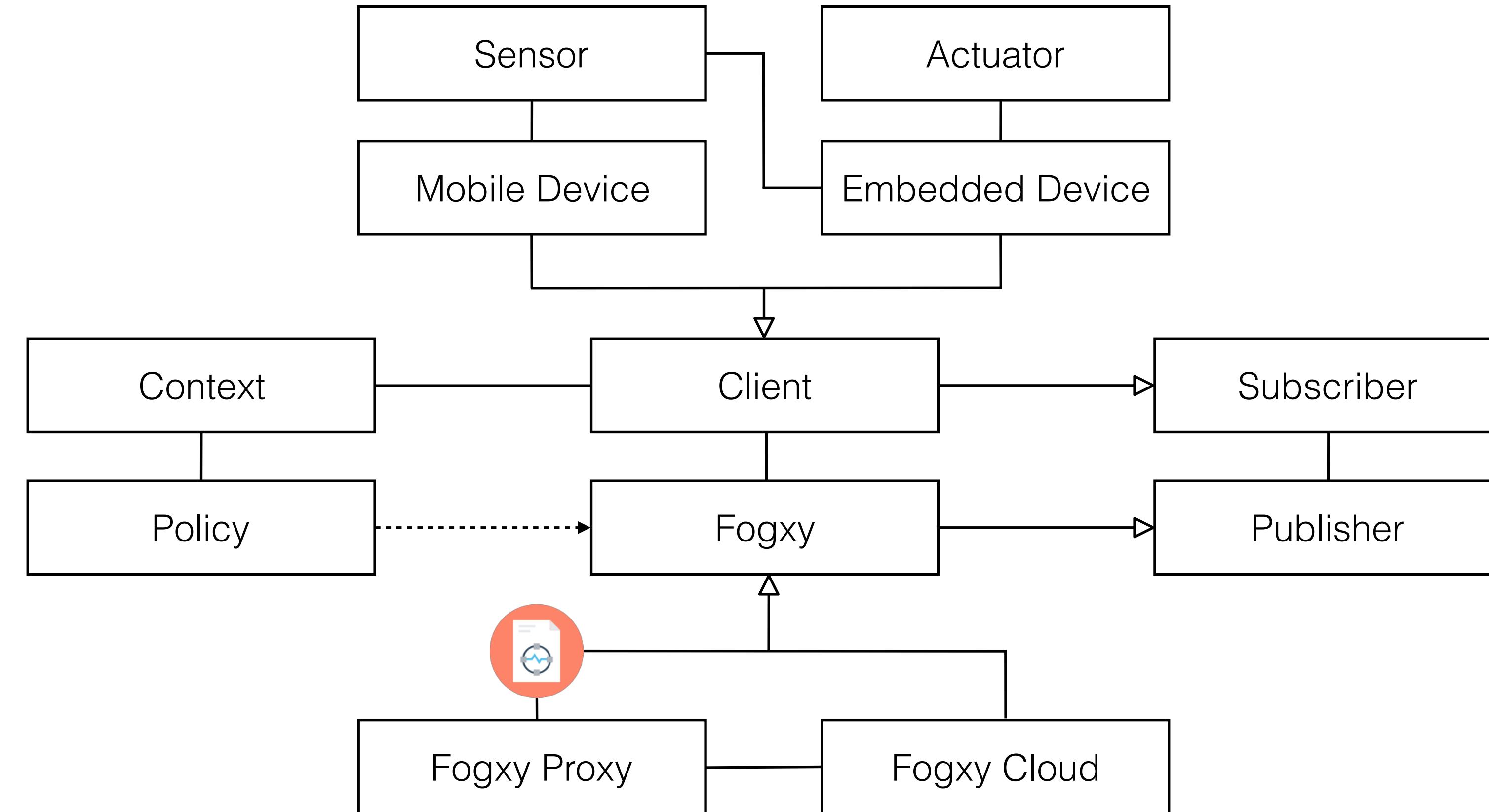


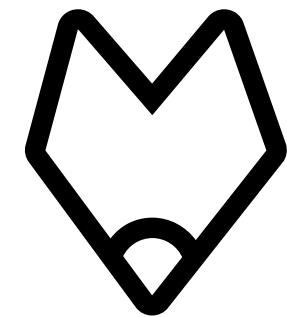
Dynamic View



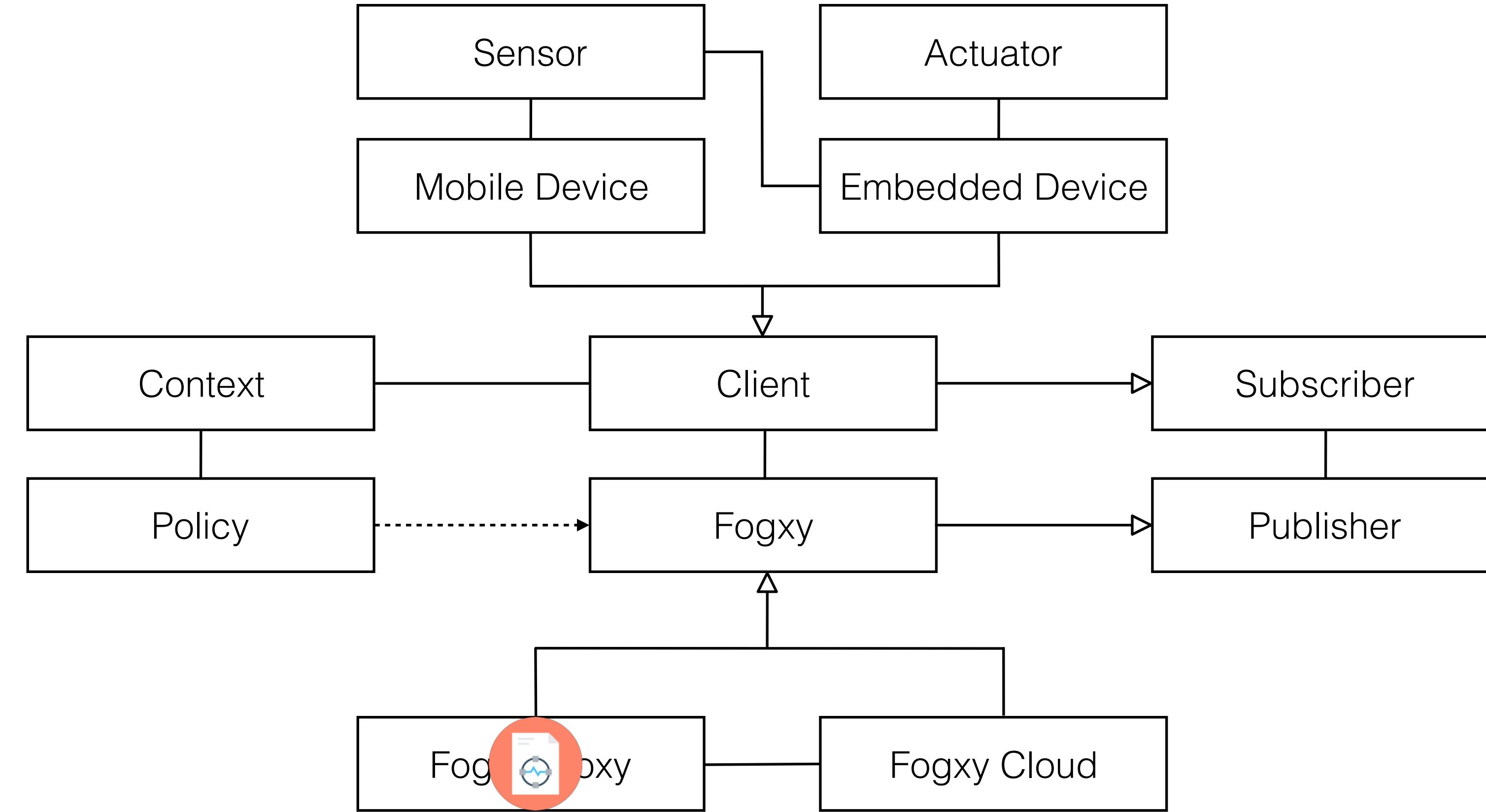


Dynamic View



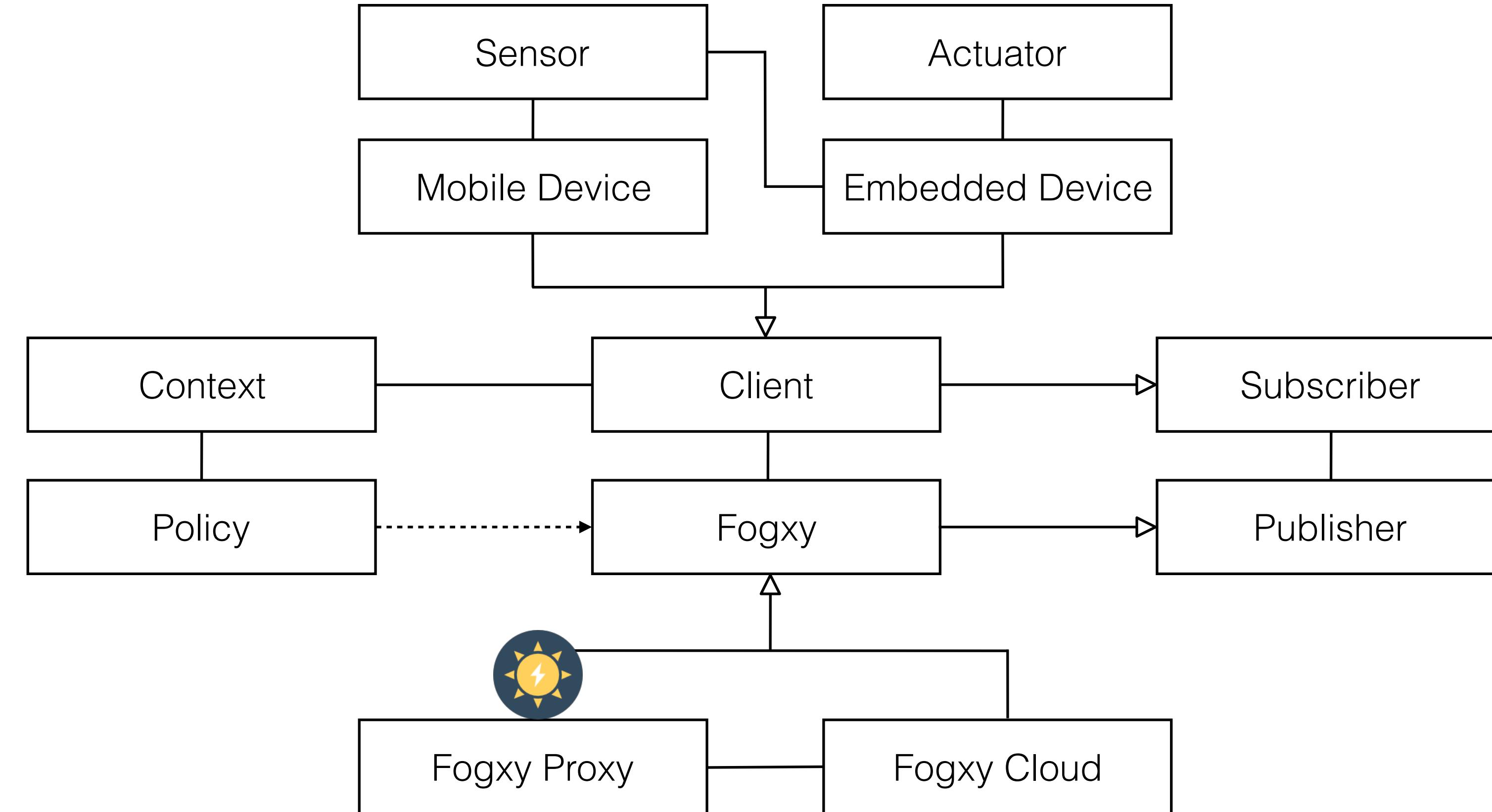


Dynamic View



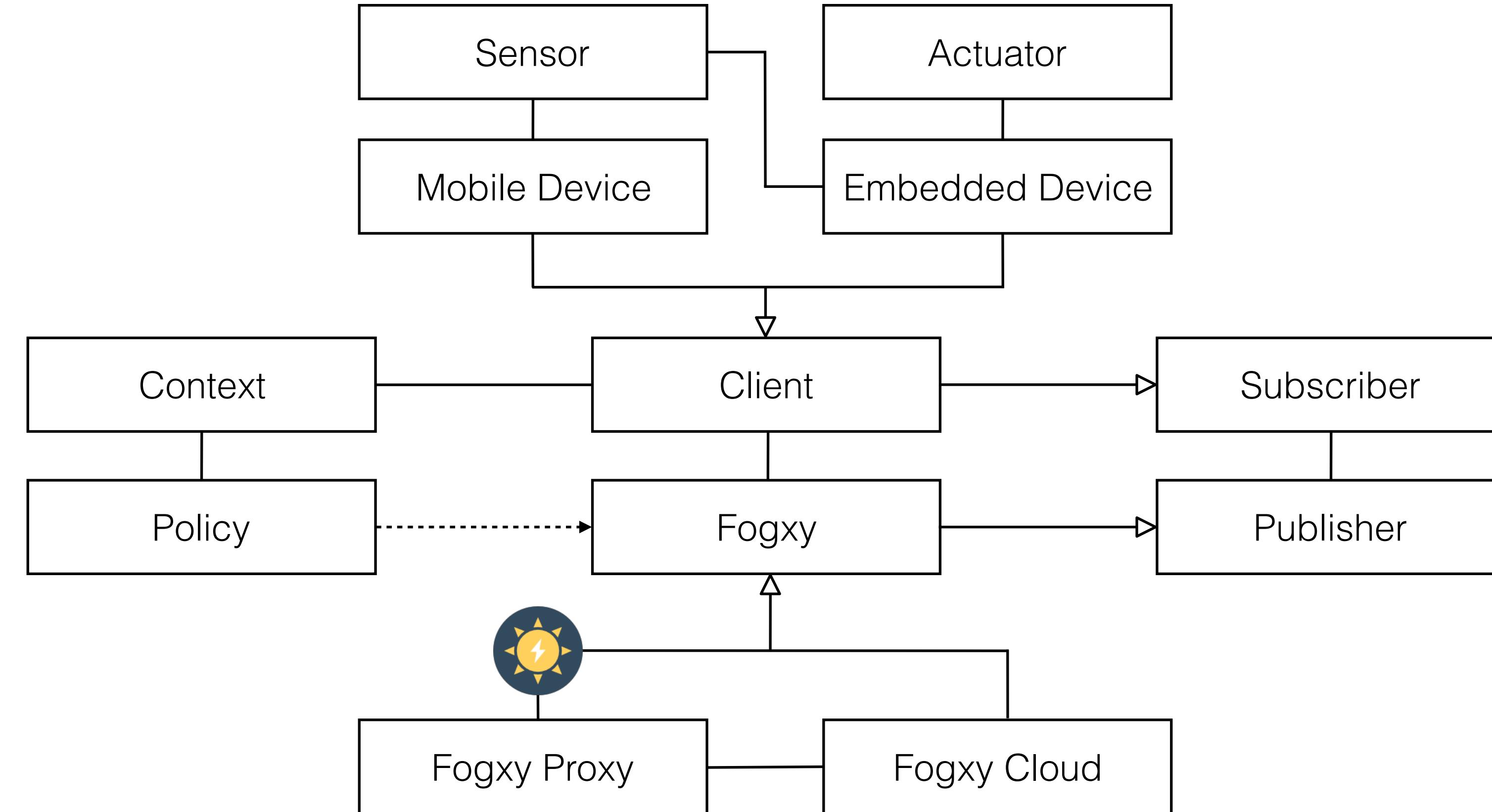


Dynamic View



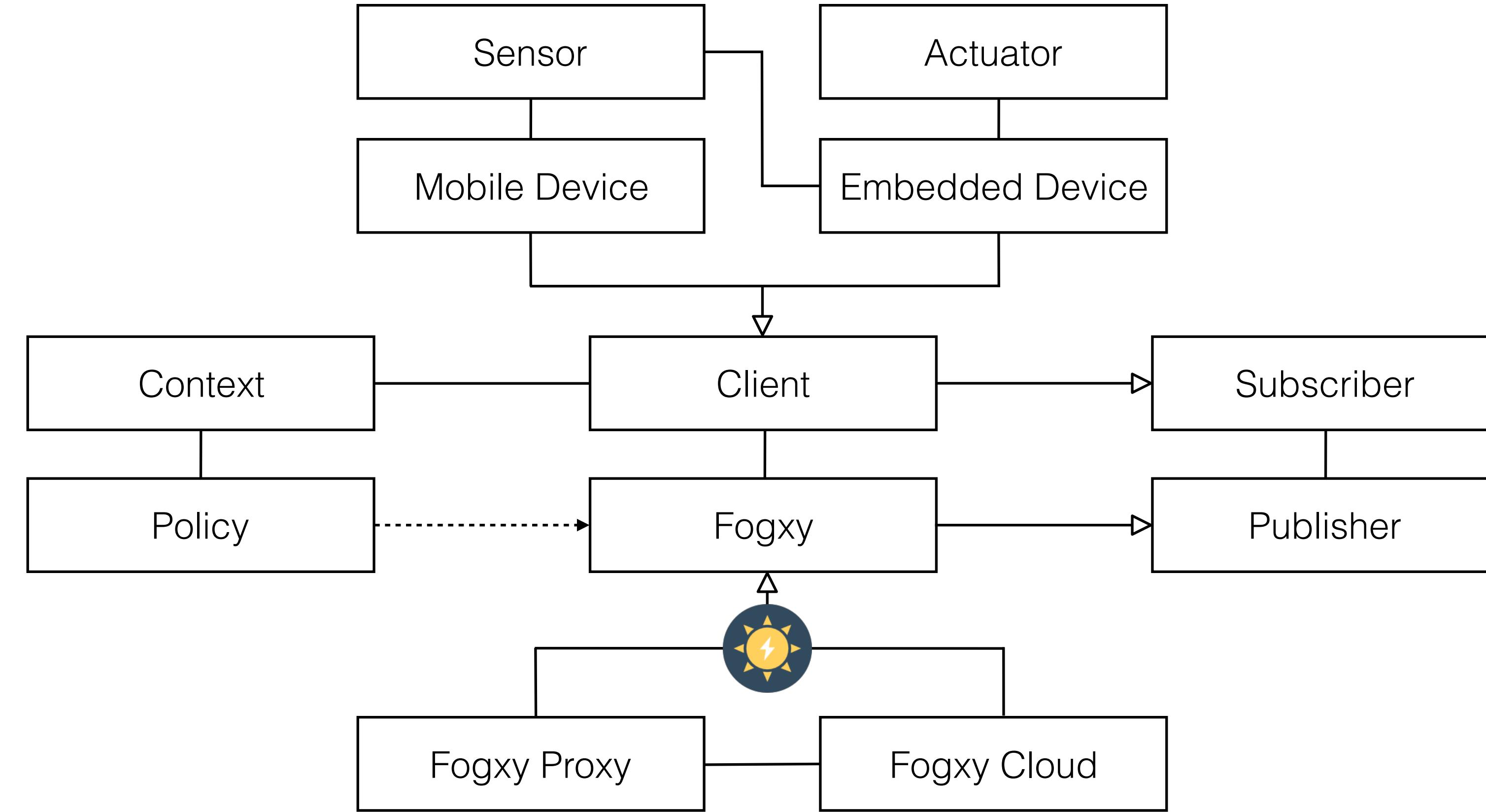


Dynamic View



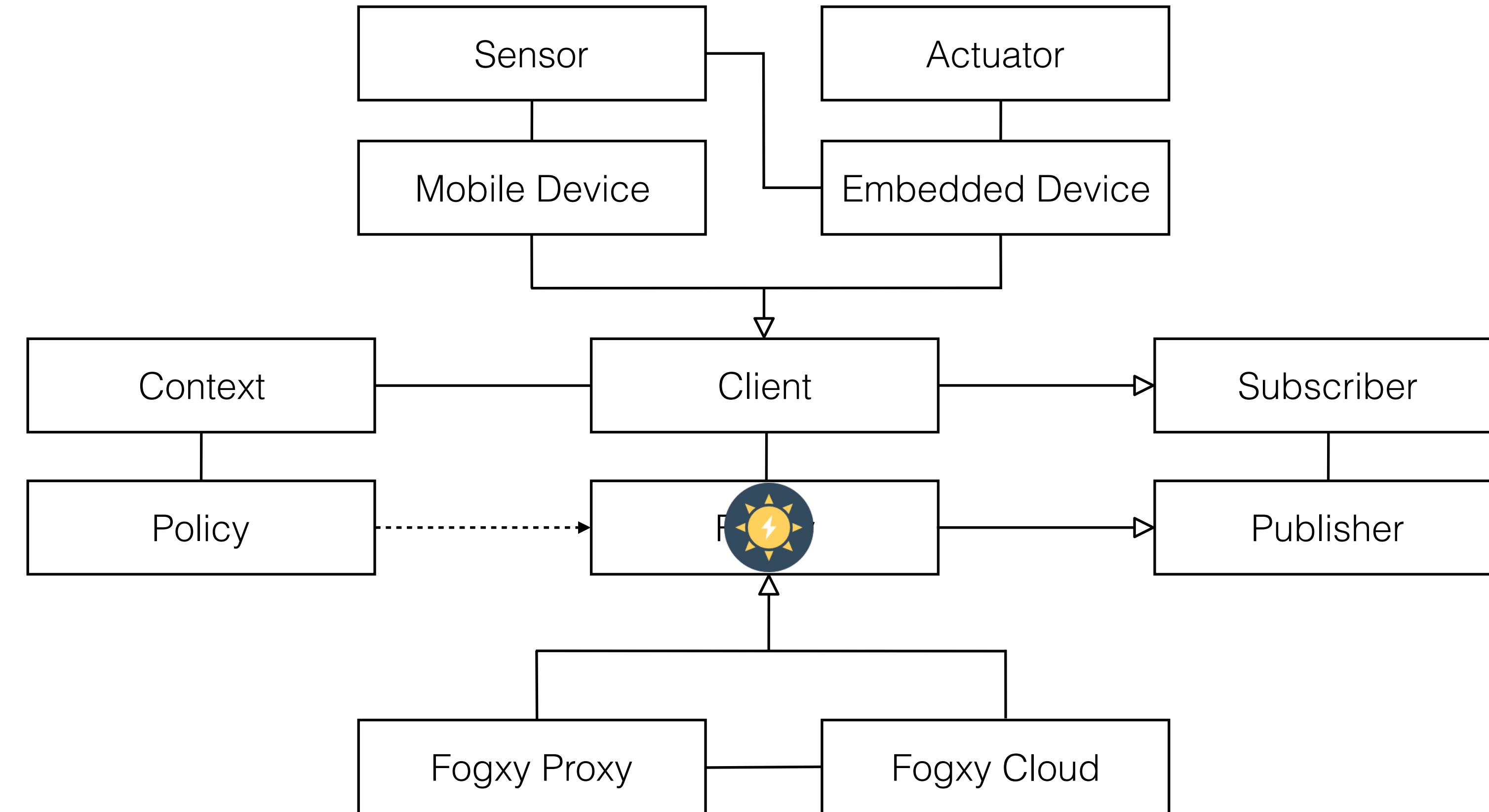


Dynamic View



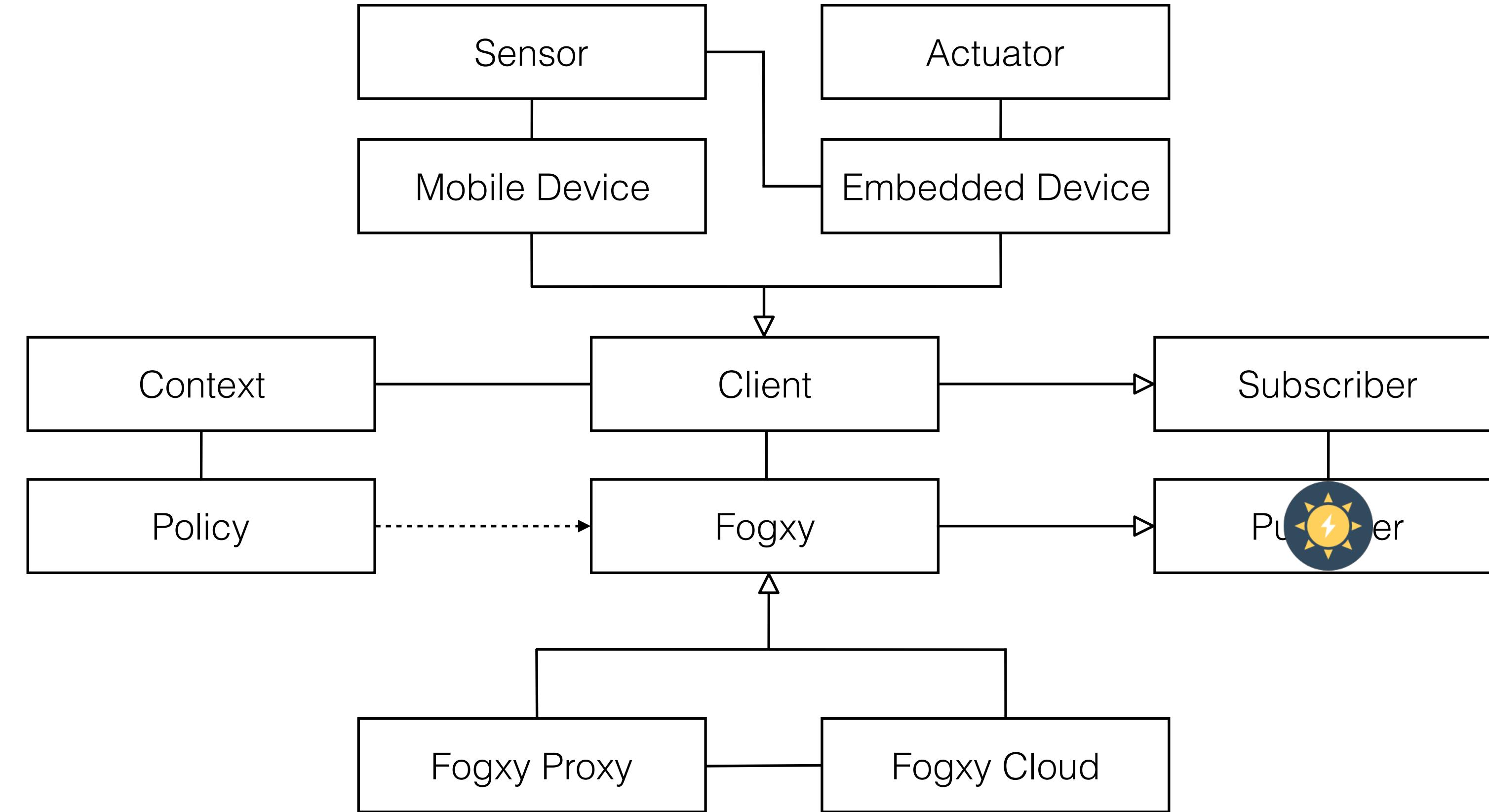


Dynamic View



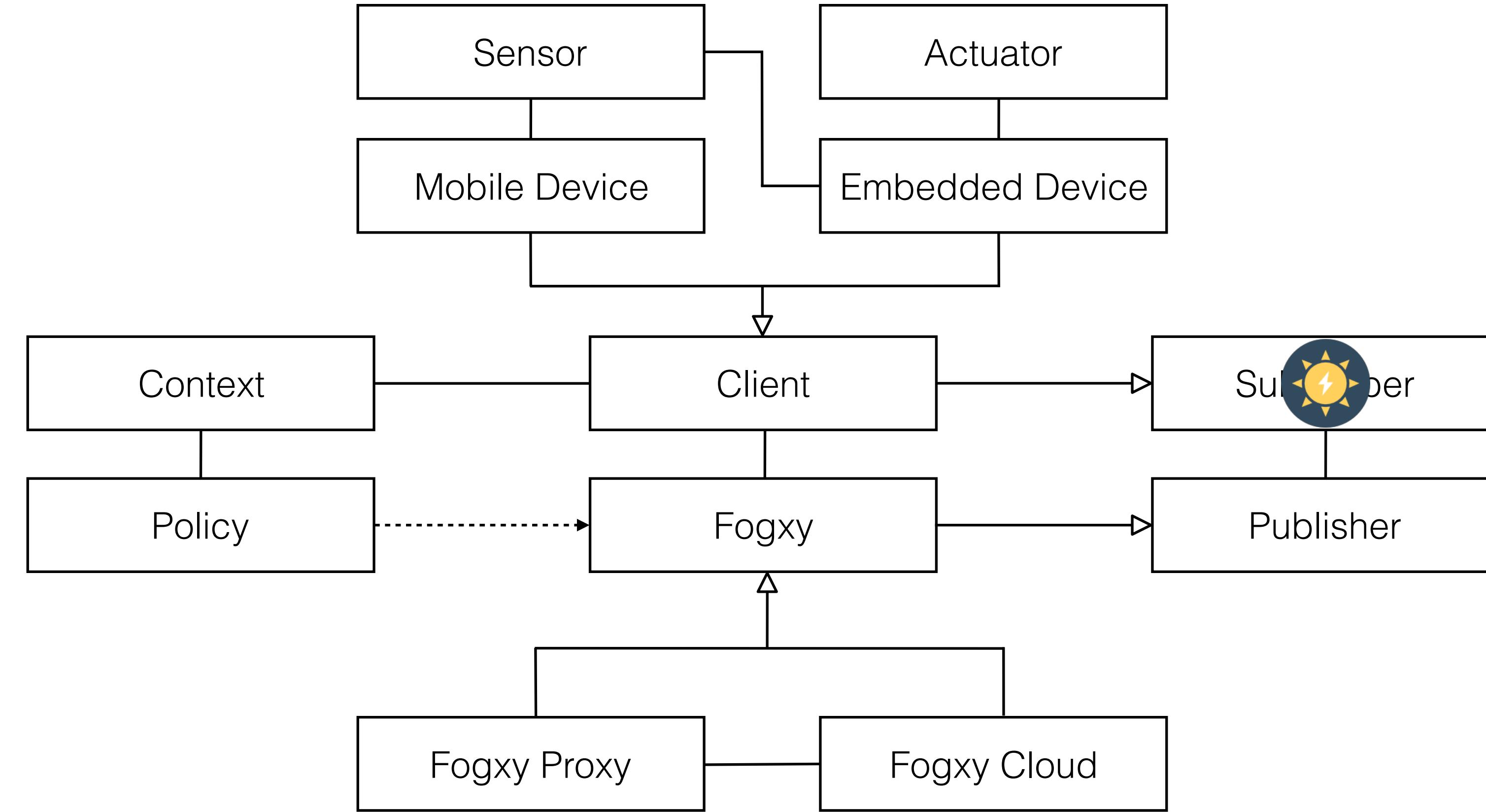


Dynamic View



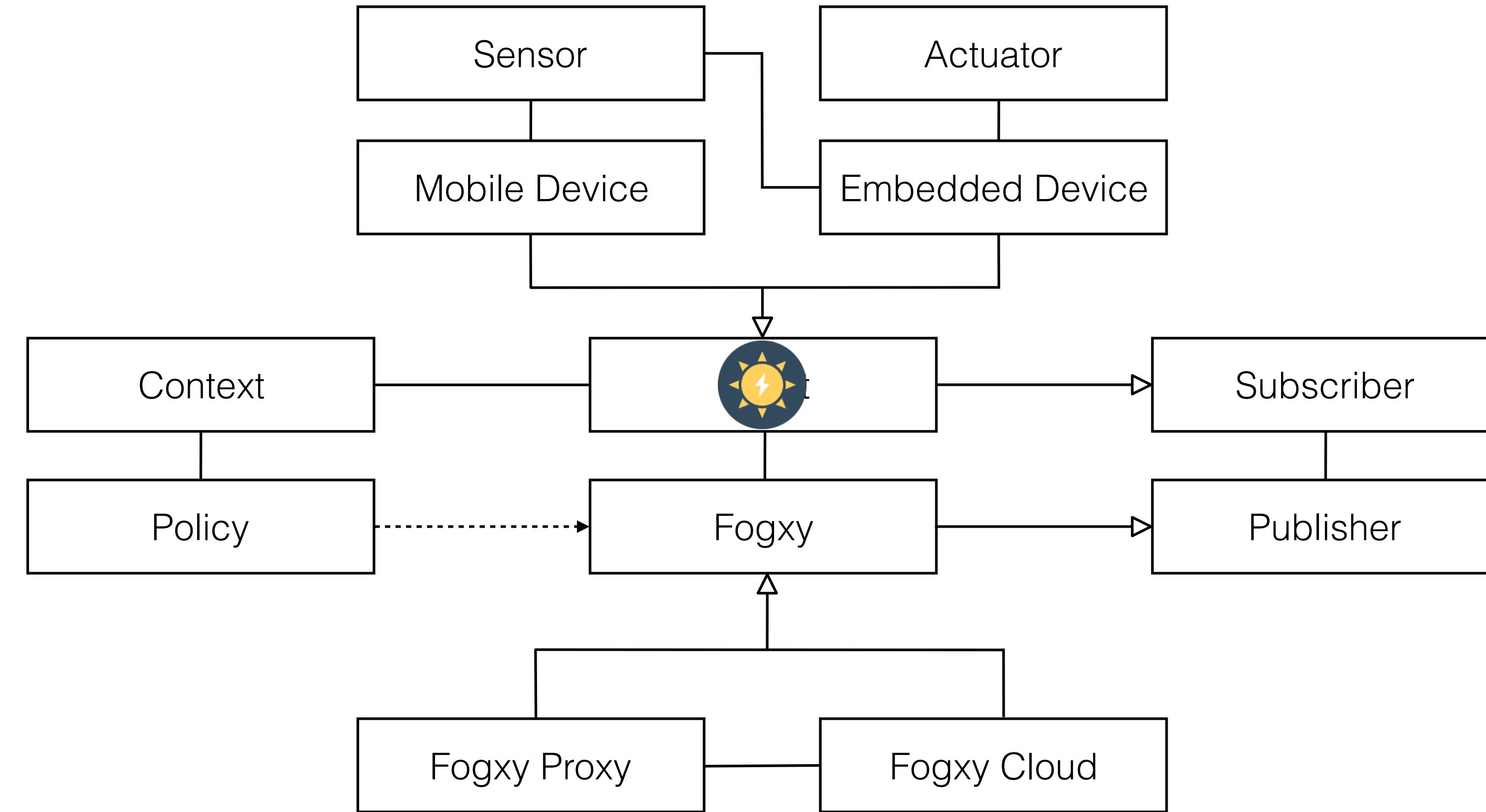


Dynamic View



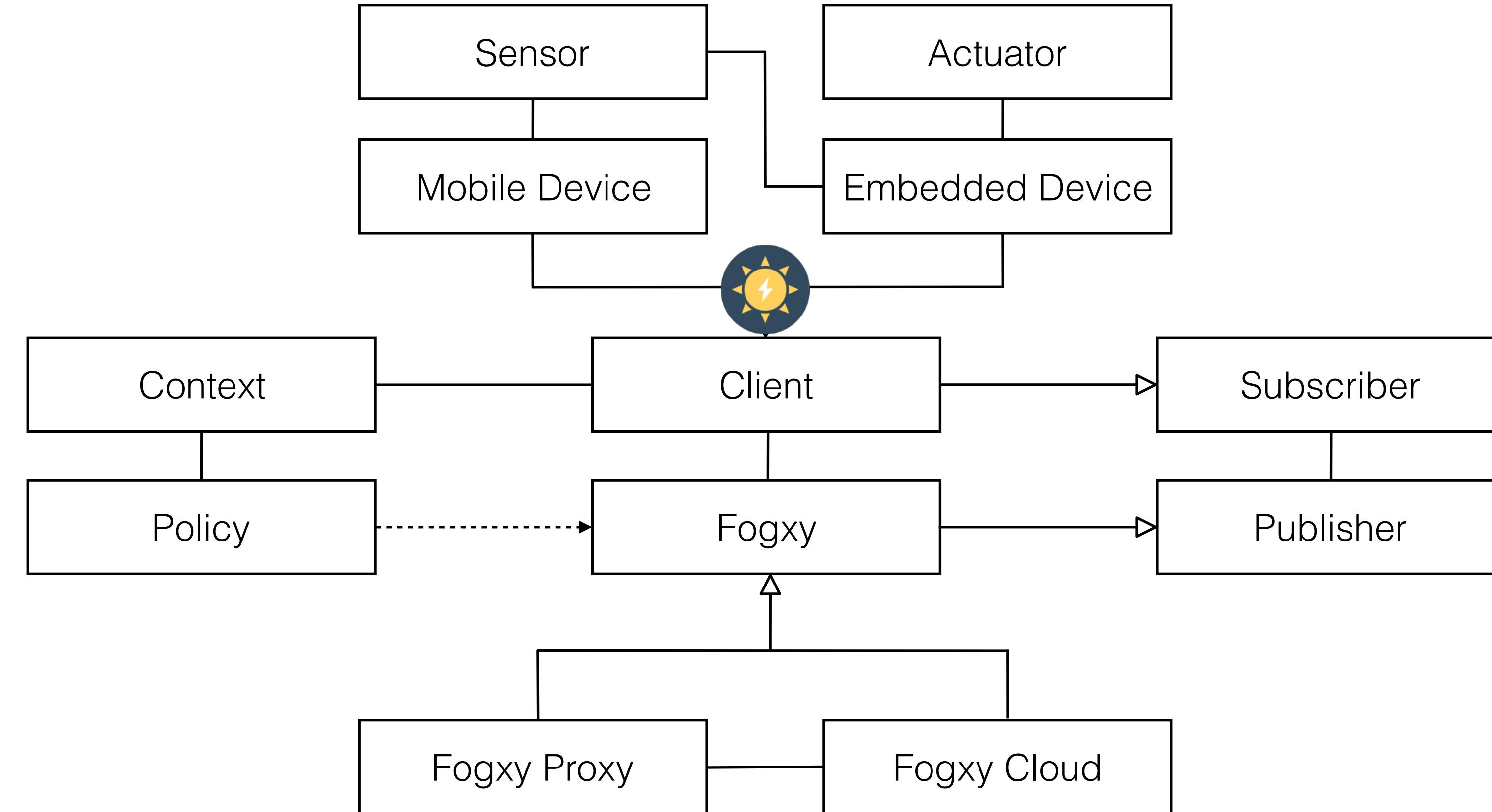


Dynamic View



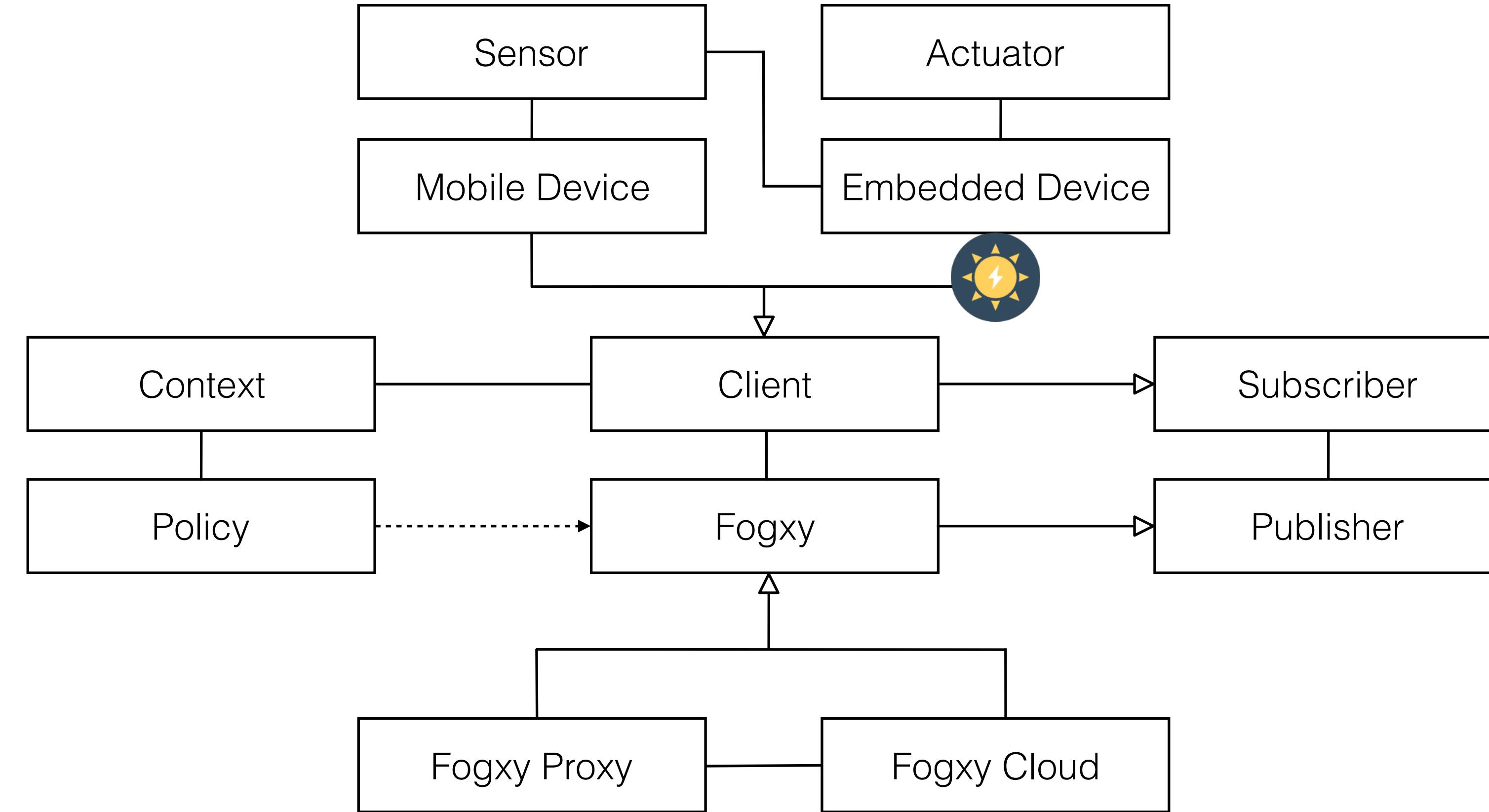


Dynamic View



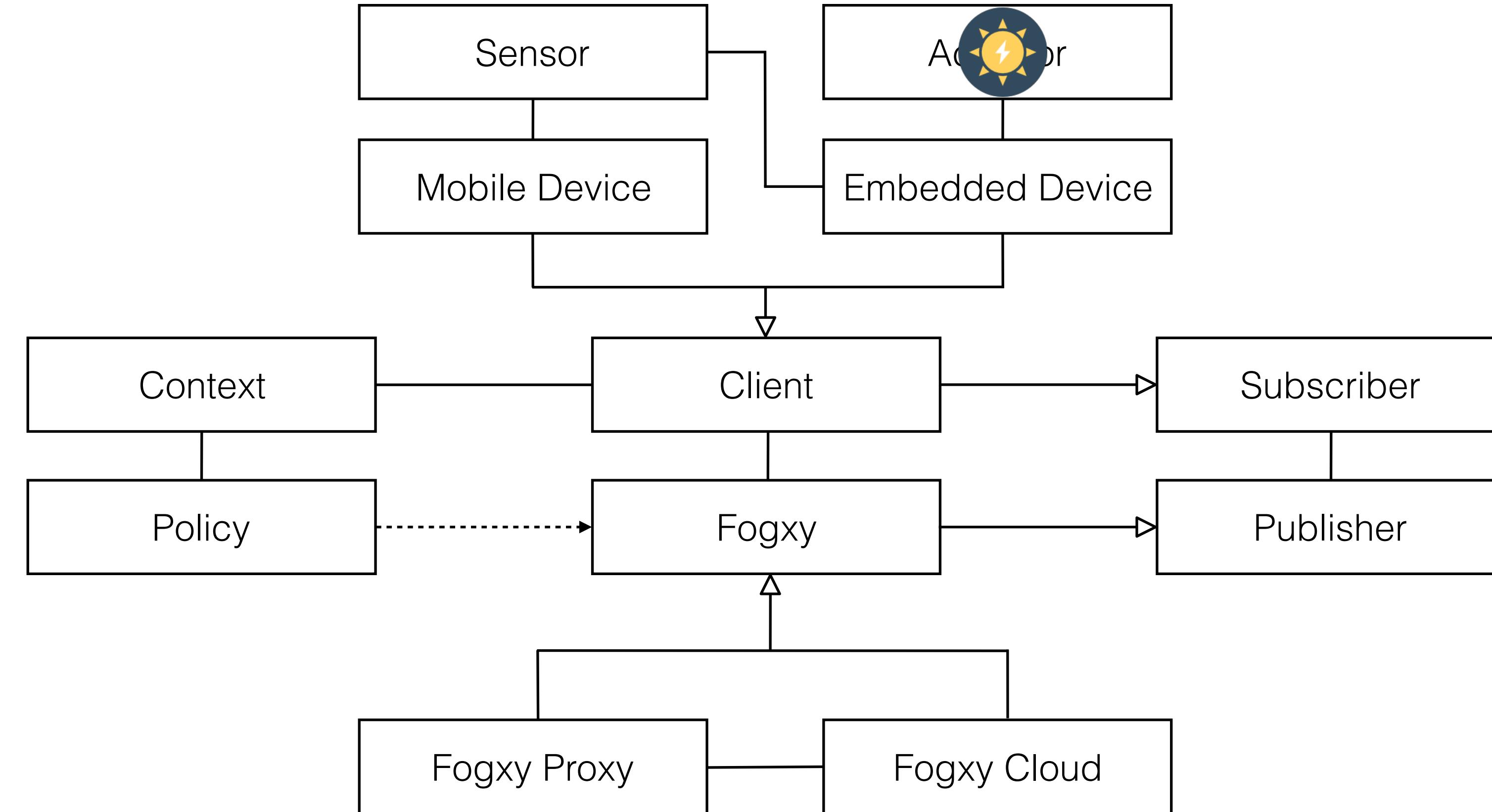


Dynamic View



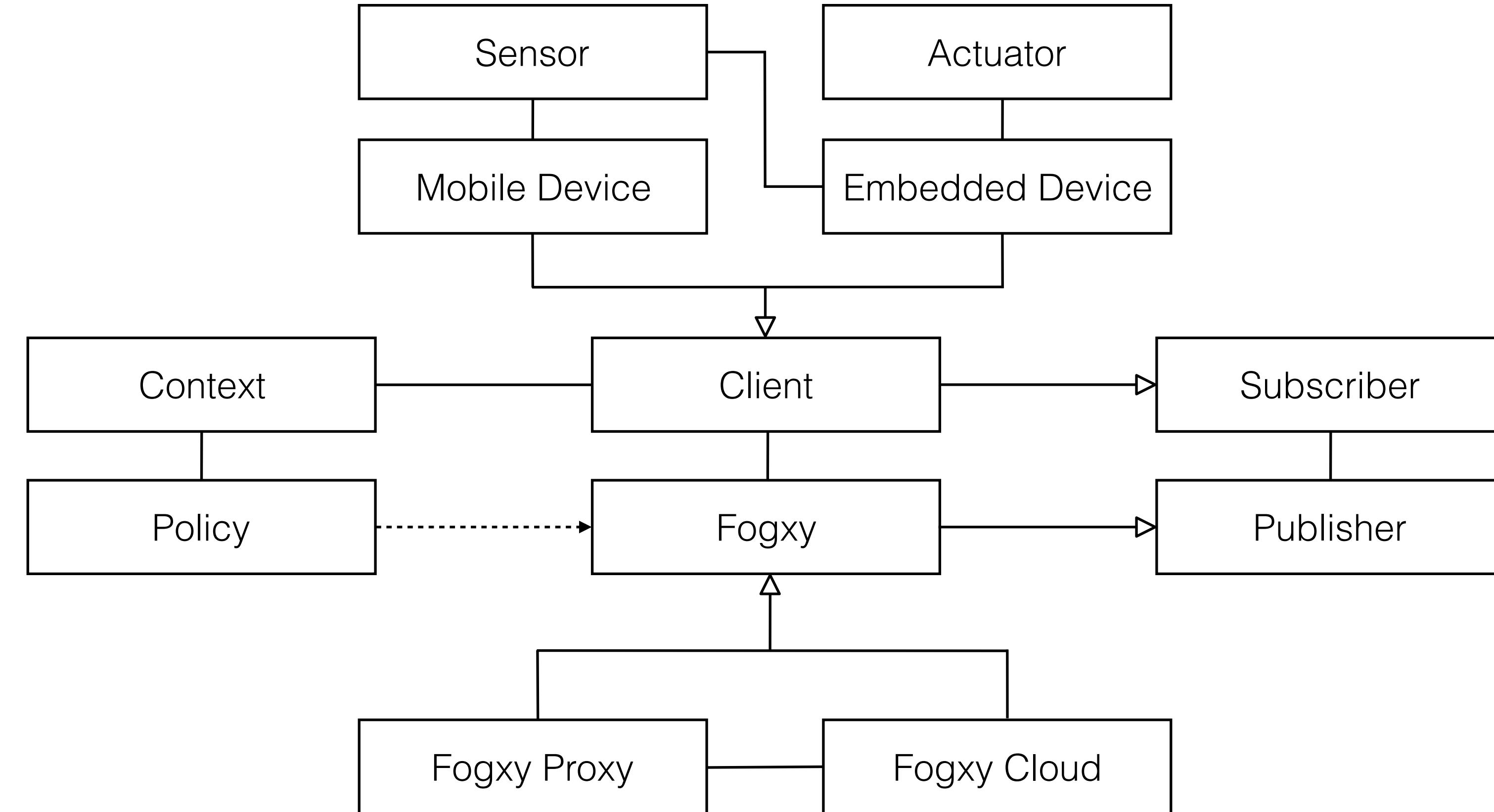


Dynamic View



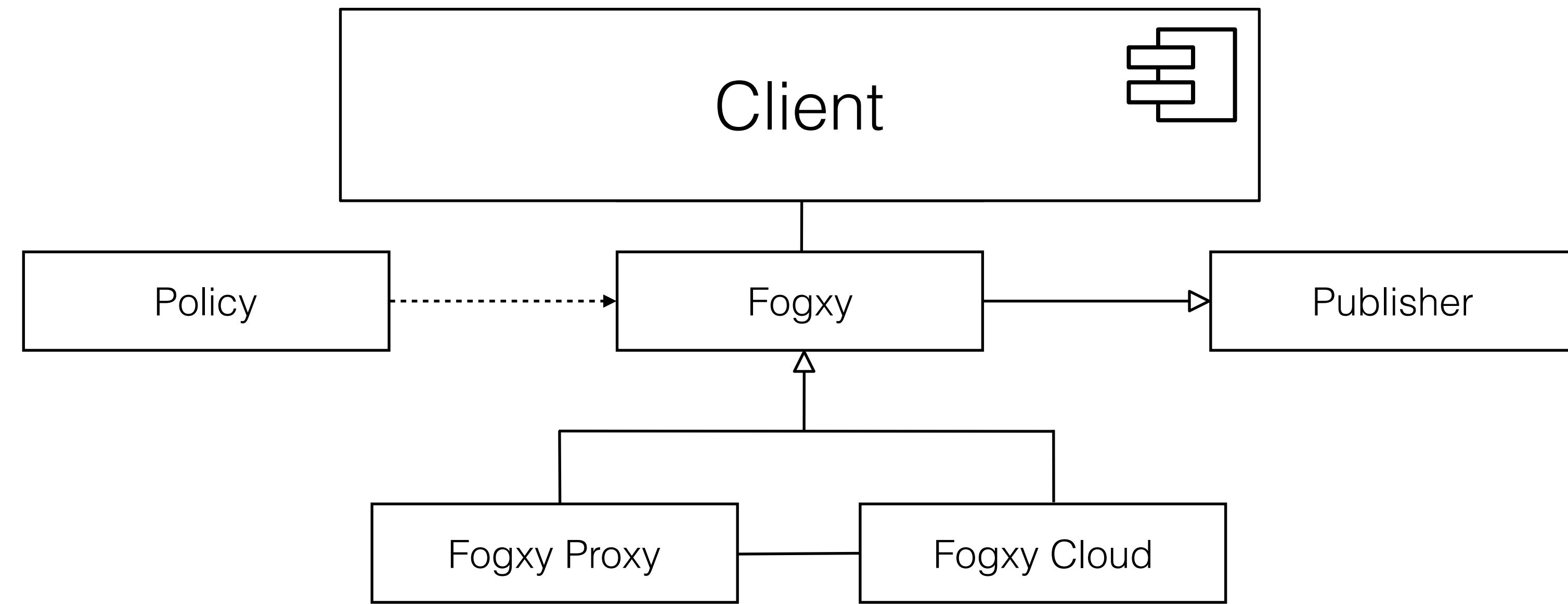


Component View



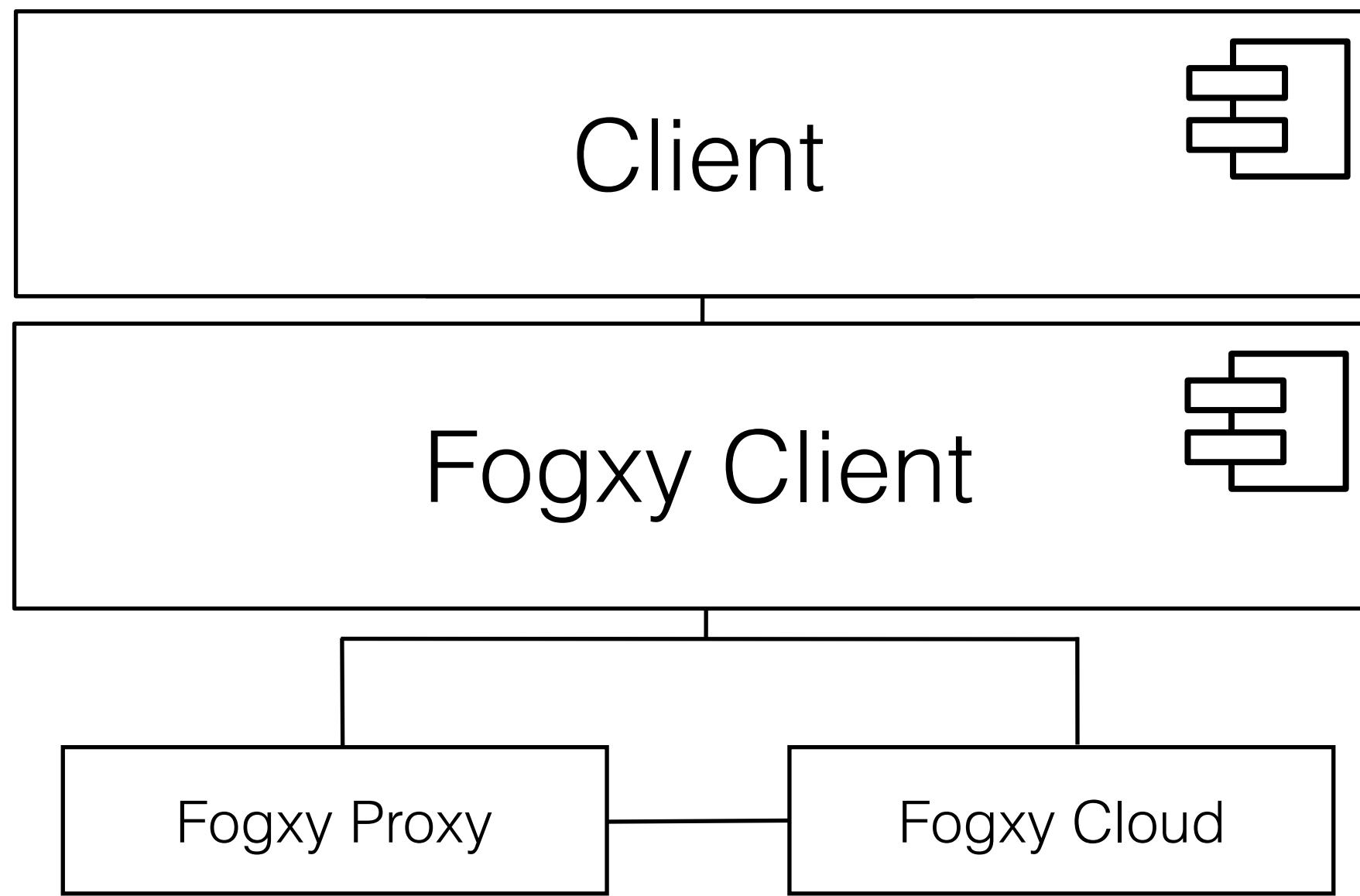


Component View



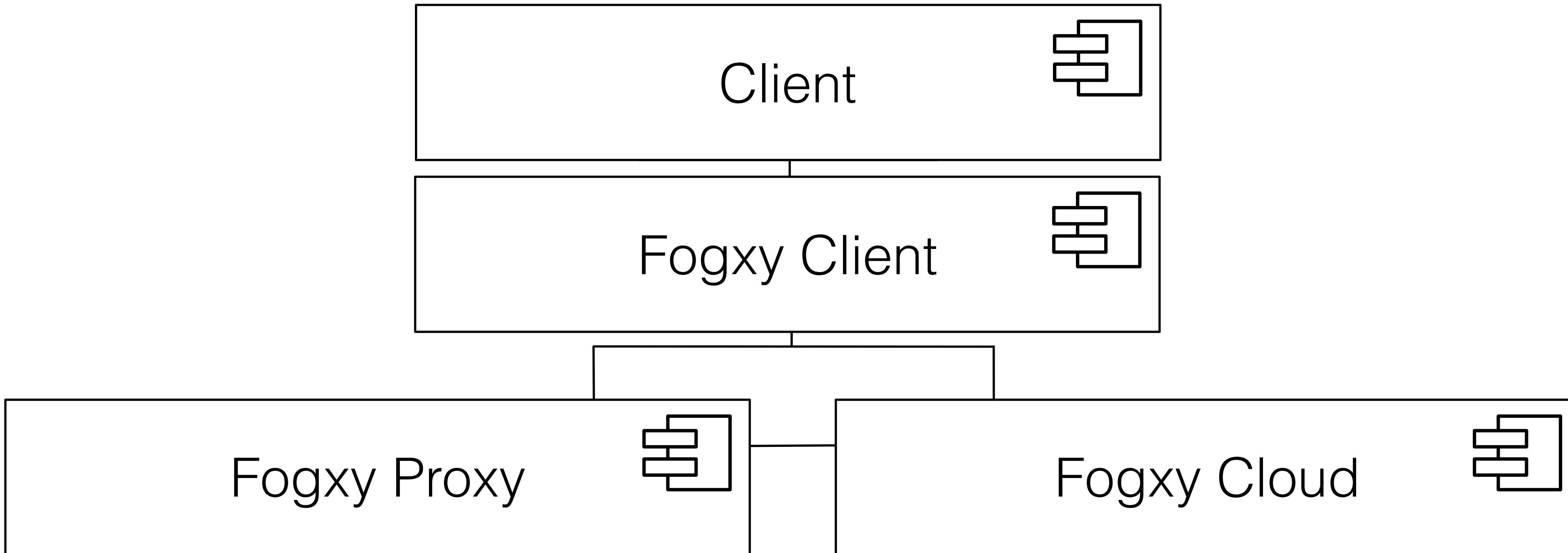


Component View

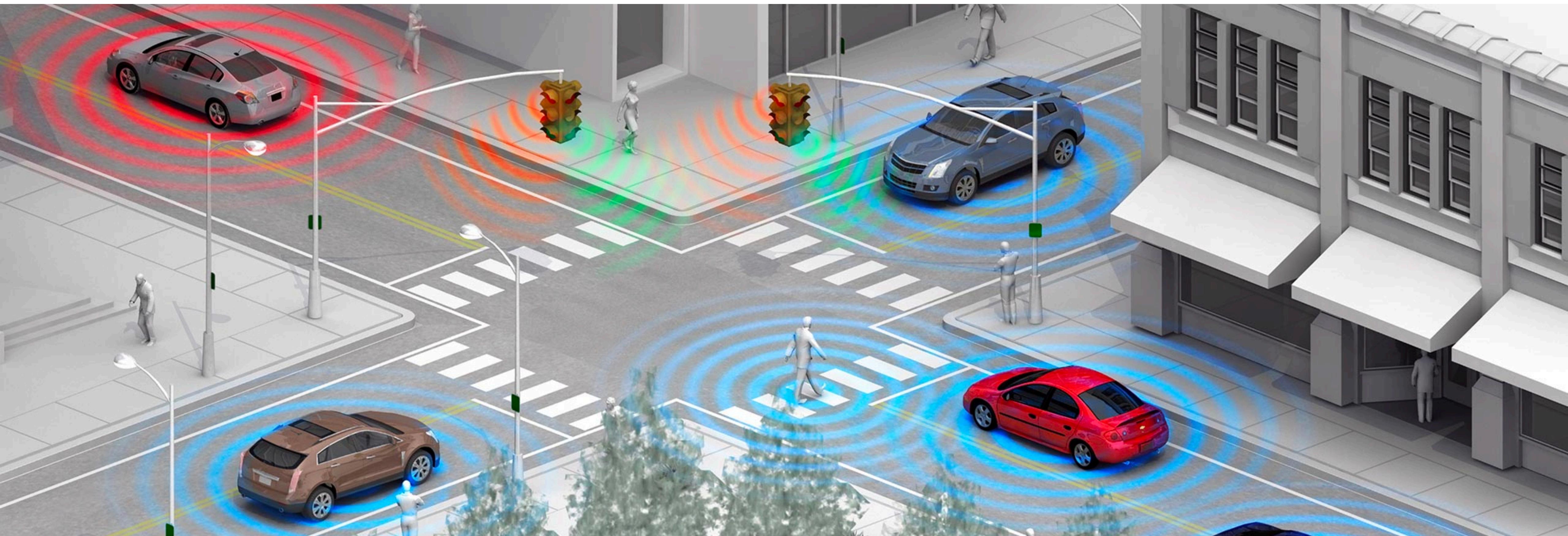




Component View

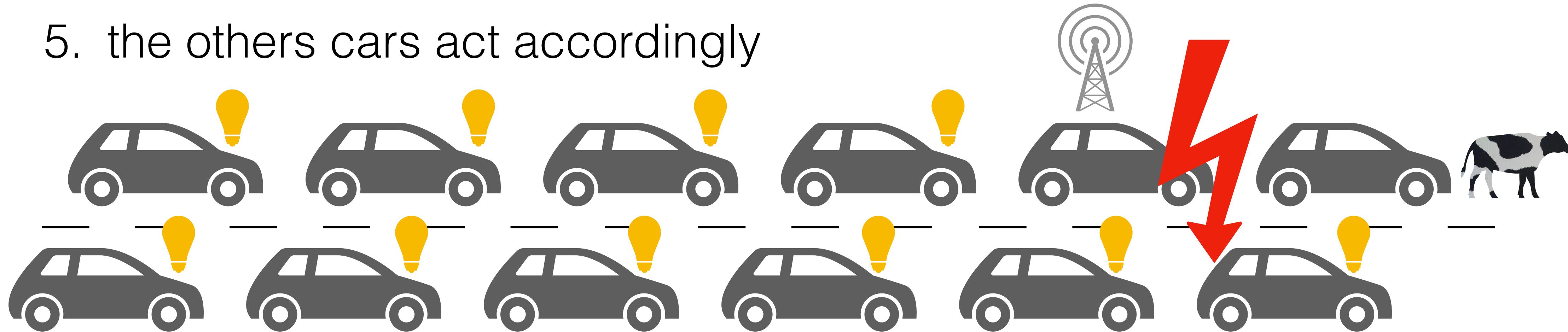


Problem Domain - Autonomous Driving



Scenario - Crash Evasion

1. Car A crashes
2. It communicates this information
3. Some central component computes actions for the best outcome
4. Some central component takes action to prevent traffic jams
5. the others cars act accordingly



Non-Functional Requirements

- The system needs to be available 99,9999% of the time (highly available)
- The decision on whether to break or to evade shall be made in less than 10ms

Task 1: Subsystem-Decomposition

- As a group come up with a subsystem-decomposition to fulfill the scenario
- All classes of Fogxy have to be included in the system

Task 2: Hardware- / Software-Mapping

- Please show, how your components map to hardware

Task 3: Apply the Demo-Scenario

- Use your Subsystem-Decomposition to show how the system carries out all required steps for the scenario

Questionnaire

- What is the idea and purpose of Fogxy?
- What are the shortcomings of Fogxy in this scenario?
- Where did you have problems implementing it?
- Which non-functional requirements could not be fulfilled? Do you have any proposals on how to fulfill them?



Thank you!

Andreas Seitz
seitz@in.tum.de

Technical University of Munich
Chair for Applied Software Engineering
Department of Informatics

Felix Thiele
felix.thiele@tum.de

Technical University of Munich
Chair for Applied Software Engineering
Department of Informatics