Sensor Observation Streams within Cloud-based IoT Platforms: Challenges and Directions

Antoine Auger, Ernesto Exposito, Emmanuel Lochin

ISAE-SUPAERO, *Université de Toulouse*, France

antoine.auger@isae.fr

Motivation

- Internet of Things (IoT) and data explosion
- Many "IoT platforms" for many use cases providing different services and/or APIs
- "Black box approach" so far
  Which requirements? Which software solutions? Why?
- Common challenges when building software platforms, need to explain:
  - Concepts, technical issues
  - How to cope with them, good practices
  - … Not just describe final results!

Motivation

- We focus on:
  - **IoT platforms**: ingest information from *Things*, create value from data, services exposed to users or applications
  - **Cloud-based**: available on demand via the Internet, Software as a Service model (SaaS), other non-functional requirements
  - **Sensor observation streams**: physical-observed phenomena or virtual-occurred events
- Contributions of the paper:
  - Insights for researchers who are interested in IoT platforms capable of handling sensor observation streams
  - Takeaways for developers

Motivation

- "Lessons learned" coming from:
  - State of the Art
  - Study of available tools/components for IoT platforms
  - Attended conferences (in particular, see **[Kamienski2016]**)
  - Own experience: development of an Integration platform for QoI Assessment as a Service (iQAS) **[Auger2017]**
- Disclaimer:
  - Not exhaustive!
  - May contain subjective thoughts…

Agenda

1. Observation-related challenges *What?*

   Observation streams, levels, quality

2. Implementation considerations *What solution? How?*

   Observation collection, processing and consumption

3. Non-functional requirements *With what features?*

   Adaptation, scalability, availability

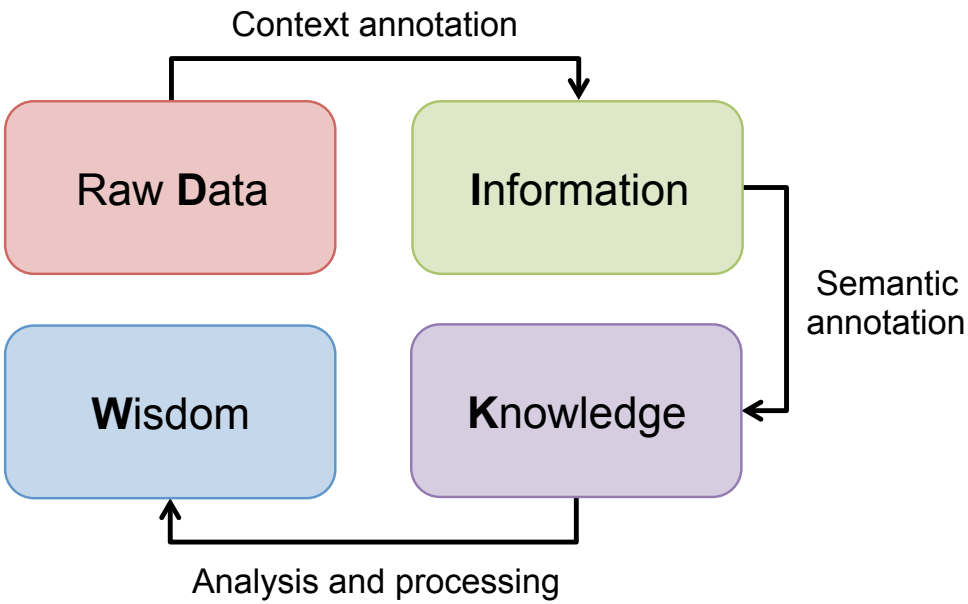
Observation-related challenges

- We assume that an observation is *something* produced by a sensor (either physical or virtual)
- IoT platforms are asked to create value from received observations
  - Data-centric systems
- Heterogeneous observations:
  - Level
  - Quality
  - Sensing rate
  - Coverage
  - …

Observation-related challenges

Observation levels

- Taxonomies to denote different observation levels
  - Useful to estimate the level of complexity required to process and "understand" observations by final consumers



An augmented version of the "DIKW ladder" of Sheth **[Sheth2016]**

Observation-related challenges

Observation quality

- Several quality dimensions:
  - Quality of Service (QoS)
  - Quality of Information (QoI) **[Bisdikian2013]**
  - Context, Quality of Context (QoC)
  - Quality of Knowledge (QoK)

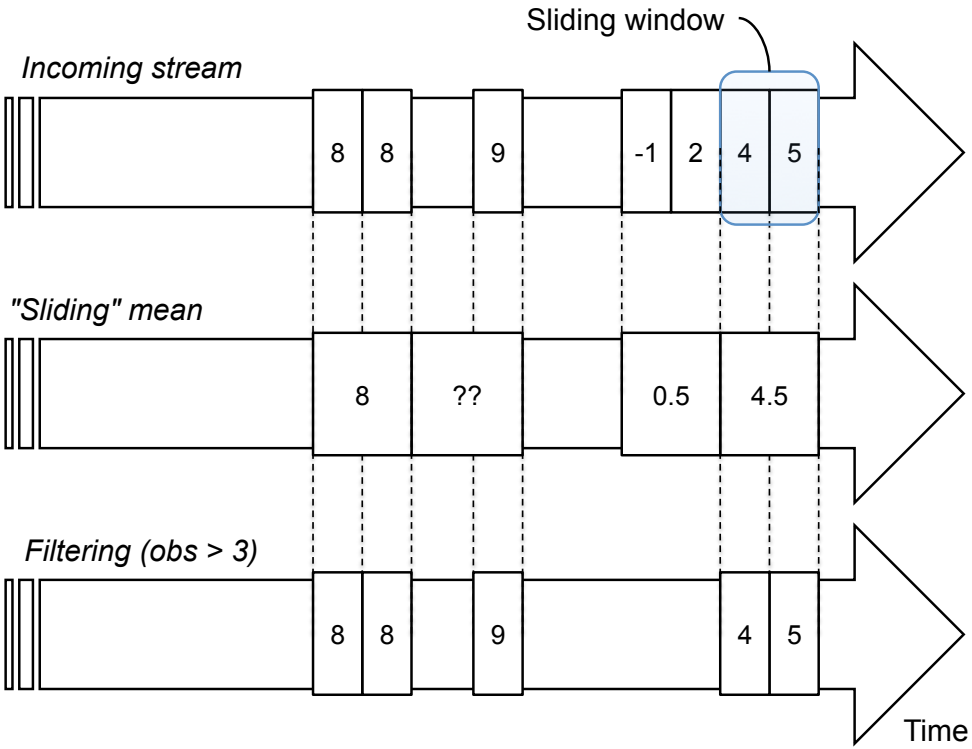
In short:

- ✓ All dimensions not always needed
- ✓ Attributes to consider may vary according to use cases (*delay*, *obs. lifetime*, *accuracy*, *trust*, …)
- ✓ Recent works consider mainly QoS-QoI (e.g., FP7 CityPulse)

Observation-related challenges

Unbounded observation streams

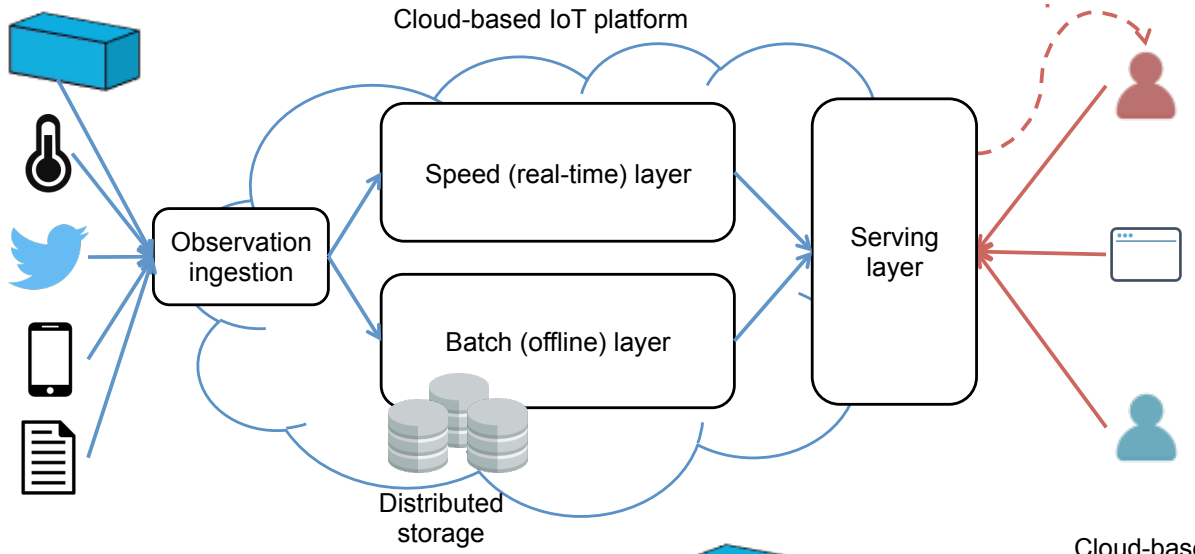
- Operations on streams ≠ databases
- Observation streams
  - Unbounded
  - Heterogeneous
  - Real-time
- Reactive Streams
  - http://www.reactive-streams.org
  - Asynchronous stream processing with non-blocking back pressure
  - Java APIs



Implementation considerations

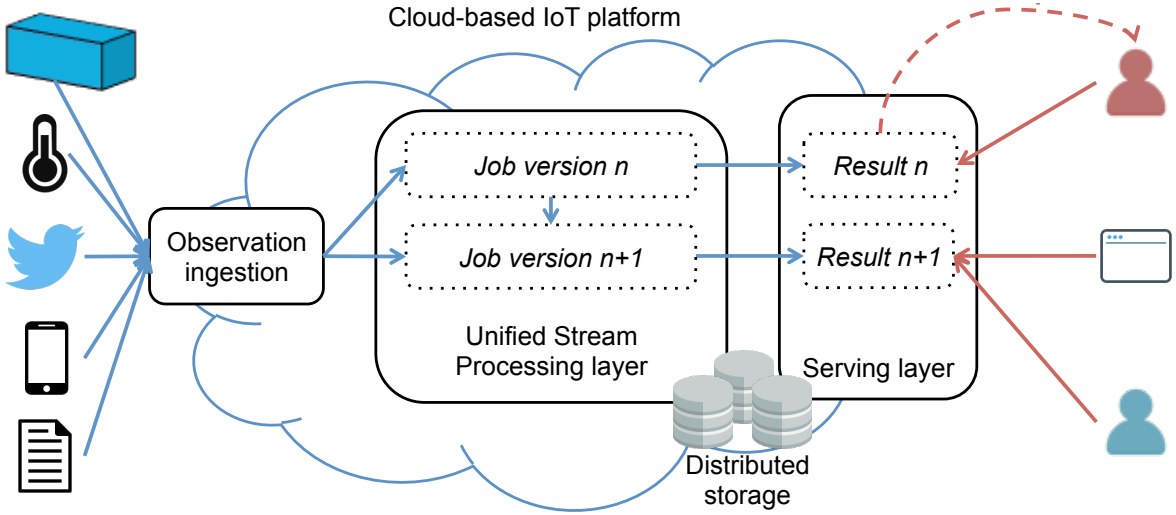
- A jungle of IoT platforms
  - Open-source solutions, projects:
    FP7 INSIGHT, FP7 OpenIoT, FP7 CityPulse, H2020 OrganiCity, …
  - Commercial or proprietary solutions (SaaS):
    AWS IoT, IBM Watson IoT, …
- Researchers may want to add new features:
  - By extending existing solutions: requires a learning phase
  - By developing new prototypes from scratch (preferred)
- Most popular data architectures:
  - Lambda Architecture http://lambda-architecture.net
  - Kappa Architecture http://www.kappa-architecture.com

Implementation considerations



Lambda Architecture

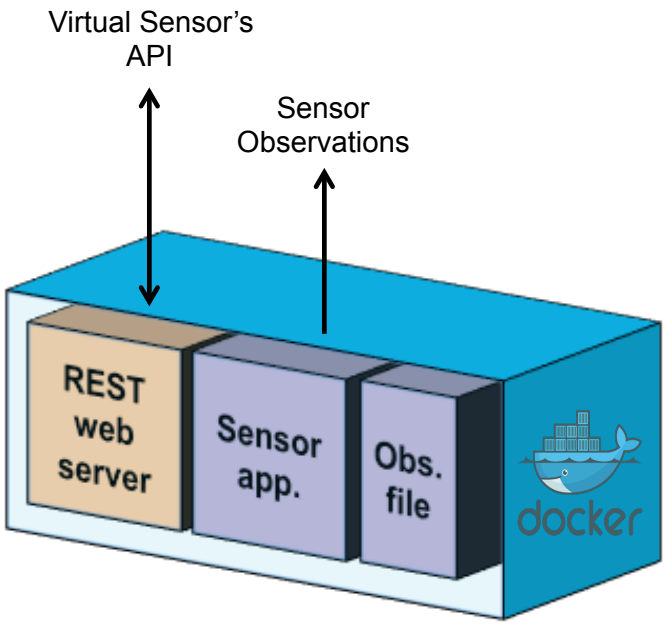
Kappa Architecture



Implementation considerations

Observation collection

- Virtual Sensor Containers (VSCs)
  - For preliminary tests
  - To assess platform scalability
  - APIs to modify sensor behavior on the fly

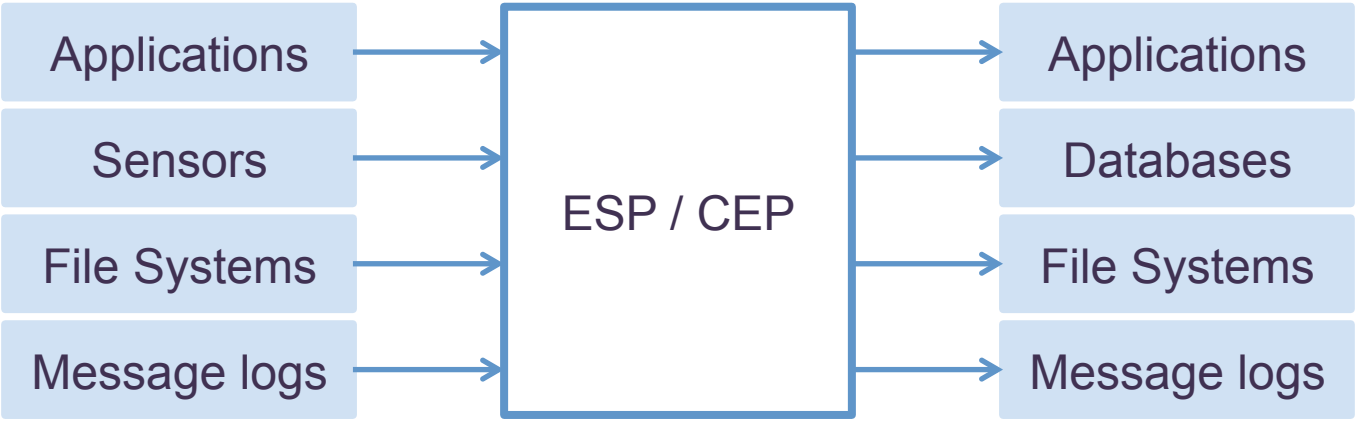


- Apache Nifi
  - https://nifi.apache.org
  - Open source, web-based GUI
  - Run in local/cluster mode
  - Supports directed graphs of data routing, transformation, and system mediation logic
  - Allows you to develop your own processors

Implementation considerations

Observation processing

- Event Stream Processing (ESP)



- Pick one framework according to your needs
  - Final choice may depends on many features **[ESPcomparison]**

Implementation considerations

Observation consumption

- Message Brokers
  - "Shock absorbing" technology
  - Compliant with Reactive Streams initiative
- Trend: Kafka for the IoT
  - Developed by LinkedIn, open sourced in 2011
- Benchmarks for comparison but biases
- Real bottlenecks: final consumers

	RabbitMQ	ActiveMQ	kafka
First release	2007	2012	2014
Solution based on	AMQP	JMS	N/A
Distributed	yes (cluster)	yes (Zookeeper cluster)	yes (Zookeeper cluster)
Exchange types	Queues, Topics	Queues, Topics	Topics
Routing support	yes	yes	no
Written in	Erlang	Java	Scala
Producer performance (messages/sec.)	25000	2000	50000
Consumer performance (messages/sec.)	4800	5000	22500

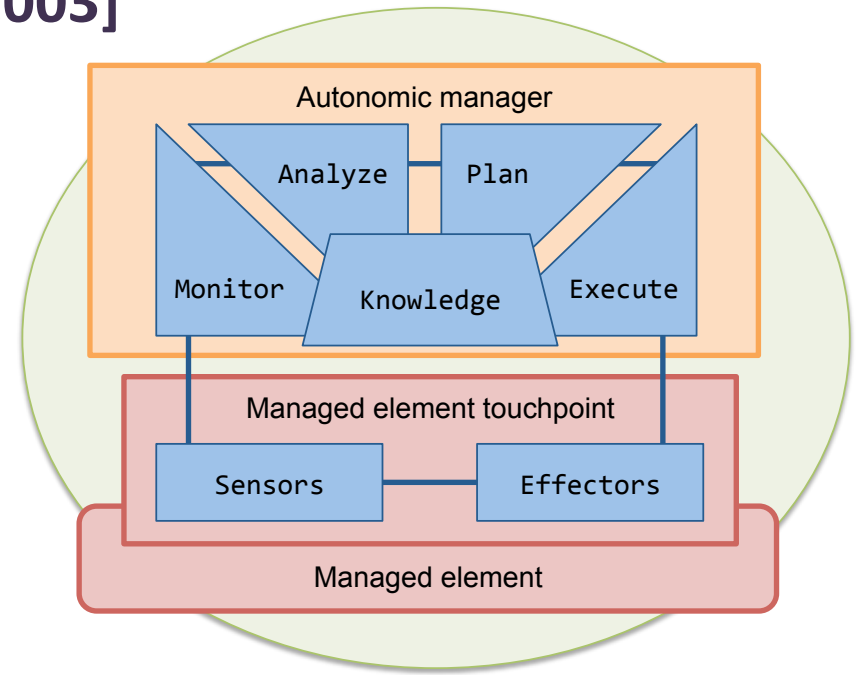
Non-functional requirements

- Functional requirements depends mainly on use cases
- Some non-functional (NF) requirements are recurrent regarding Cloud-based IoT platforms
- Among them, some NF requirements may be fulfill by relying on the Cloud Computing paradigm

Non-functional requirements

Platform adaptation

- Autonomic Computing **[Kephart2003]**
  - IBM MAPE-K control loop
    - Monitor
    - Analyze
    - Plan
    - Execute



Taken from **[Kephart2003]**

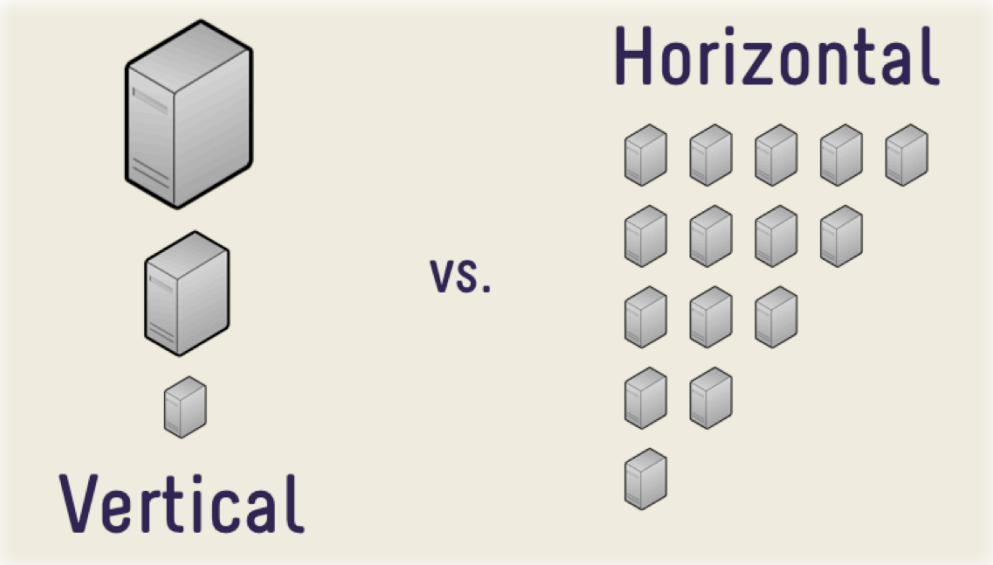
- IoT platforms act as mediators
  - Sensor capabilities
  - Consumer needs
  - Service Level Agreements (SLAs)

- Adaptation is needed
  - E.g., self-configuration, self-optimization, self-healing, …

Non-functional requirements

Platform scalability

- Vertical scalability
  - Allocate more resources
  - VM migration
- Horizontal scalability
  - Load balancing
  - Cluster
- Scalability ≠ elasticity
  - Scalability = add resources on-demand
  - Elasticity = add or remove resources on-demand



Taken from http://pudgylogic.blogspot.fr/2016/01/horizontal-vs-vertical-scaling.html

Non-functional requirements

Platform availability

- CAP theorem **[Brewer2000]** proposed by Eric Brewer
  - *"You can only have 2 of 3 features among"*:
    - **C**onsistency
    - **A**vailability
    - Tolerance to network **P**artitions

- CAP theorem has evolved **[Brewer2012]**:
  - Network partitions can be handled
  - Tradeoff instead of exclusive choice (latency considerations)

- IoT platforms raise new issues:
  - Platform availability may depend on the Cloud Provider (SLA)
  - Sensors availability
  - Is data consistency critical?

Other challenges

Many other challenges such as:

- General architecture for the IoT **[FP7 IoT-A]**
- Interoperability **[FP7 Fiesta-IoT] [FP7 OpenIoT]**
- Observation collection (e.g., CoAP protocol)
- Development tools to create IoT applications or services
- Specific challenges for Smart City platforms (many stakeholders)
- "Big Data" challenges: *Volume, Value, Velocity, Variety*, *Veracity* and more…

=> Not covered in this paper

Conclusions and perspectives

- IoT data explosion
- A jungle of IoT platforms but:
  - Complex to add new features to them, requires a learning phase
  - SaaS = “Black box approach”
  - Need to sensitize researchers and developers about concepts, issues and good practices
- In this paper, we focus on sensor observation streams within Cloud-based IoT platforms
  - Observation-related challenges
  - Implementation considerations
  - NF requirements and benefits of using Cloud Computing

References

[Kamienski2016] Carlos Alberto Kamienski. Development of IoT-based Applications for Smart Cities. Tutorial given at *IEEE WF-IoT 2016* in Reston (USA). Available at: https://www.researchgate.net/publication/311607654_Development_of_IoT-based_Applications_for_Smart_Cities

[ESPcomparison] Available at: https://www.madewithtea.com/images/streaming.png

[Bisdikian2013] C. Bisdikian, L. M. Kaplan, and M. B. Srivastava, "On the Quality and Value of Information in Sensor Networks," *ACM Trans. Sen. Netw.*, vol. 9, no. 4, pp. 48:1–48:26, 2013.

[Sheth2016] A. Sheth, "Internet of Things to Smart IoT Through Semantic, Cognitive, and Perceptual Computing," *IEEE Intelligent Systems*, vol. 31, no. 2, pp.108–112, Mar. 2016.

[Auger2017] Auger, A., Exposito, E., & Lochin, E. (2017). iQAS: An Integration Platform for QoI Assessment as a Service for Smart Cities. *IEEE World Forum on Internet of Things 2016*.

[Kephart2003] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, 2003.

[Brewer2000] Brewer, E. A. (2000, July). Towards robust distributed systems. *PODC* (Vol. 7).

[Brewer2012] Brewer, E. (2012). CAP twelve years later: How the" rules" have changed. *Computer*, *45*(2), 23-29.

20th Innovations in Clouds, Internet and Networks

PARIS

March 7 - 9, 2017

Sensor Observation Streams within Cloud-based IoT Platforms: Challenges and Directions

Antoine Auger

antoine.auger@isae.fr

#ICIN2017

This research was supported in part by the French Ministry of Defense through a financial support of the Direction Générale de l'Armement (DGA).