Social Sensor Cloud: An Architecture Meeting Cloud-centric IoT Platform Requirements

9th KuVS NGSDP Expert Talk

Thomas Menzel Niels Karowski Daniel Happ Vlado Handziski Adam Wolisz

Telecommunication Networks Group, TU Berlin

2014-04-09

The Internet of Things

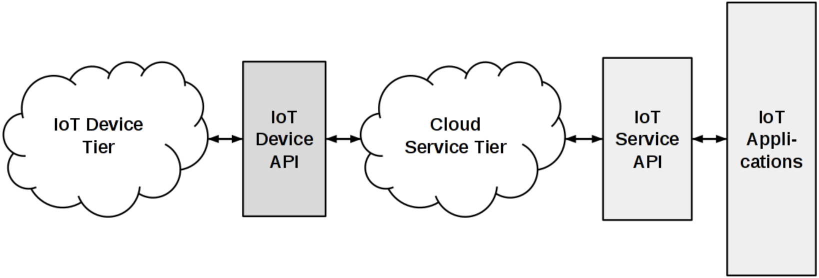
- Connect **everything** and everyone everywhere to everything and everyone else
- Platform abstracting services of **trillions** of interconnected devices
- Providing interoperability in face of **heterogeneity**
- Enabling diverse **applications** over a shared hardware resource substrate

The Cloud

- Computing resources as utility with elastic demand matching
- Core enablers:
  - Server virtualization
  - Reliable distributed storage
  - Fast networking
- Benefits
  - Flexibility
  - Reliability
  - Pay-as-you-go

Cloud-supported Internet of Things

- Cloud services as abstraction layer for IoT
- Popular approach followed by many academic and commercial platforms



Existing Solutions

- ETSI (one)M2M, Xively, Etherios, SENSEI, Sensor Andrew, FI-WARE, OSIOT, Axeda, OpenIoT, EVRYTHNG, RuBAN, openHAP, ioBridge ...
- Architecture
  - Natural decomposition
  - Explicit differentiation: data generators vs. consumers (but: Xively, ETSI M2M)
- Value-added services
  - Data storage
  - Virtual sensors
  - Device management
- Application-side Platform Access
  - Protocols
  - Encoding formats
  - Interaction patterns

Shortcomings of Existing Solutions

- Service APIs:
  - Insufficient support for low-latency real-time data streaming
  - Insufficient system integration for event-triggered interaction patterns
- Rigid decoupling
  - Missing support for cross-layer optimization
  - Device tier operation not adapted to QoS requirements of running applications leading to reduced lifetime of battery operated sensors
- Storage-centricity
  - Focus on archival of all generated data, independent from real level of interest in the data
  - Requires high initial investments and operational costs

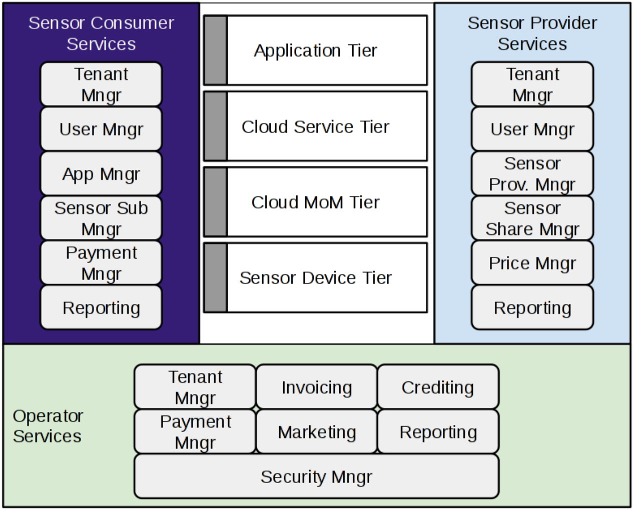
SSC Project Goals

- Reference system architecture . . . addressing these shortcomings
- Prototypical implementation
- Evaluation

SSC Approach

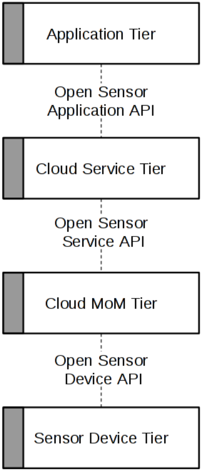
- Natural decomposition ...
- Differentiation:
  - Fine-grained architecture
  - Features of the offered services and APIs

SSC Architecture Overview



High-Level Architecture Concept

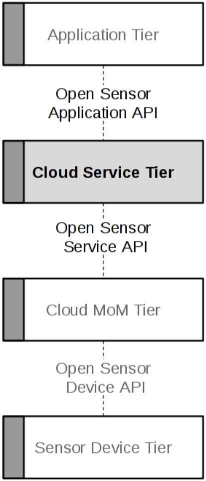
- Functional decomposition in four tiers
  - Application Tier
  - Cloud Service Tier (CST)
  - Cloud MoM Tier (CMT)
  - Sensor Device Tier (SDT)
- Interaction via three well-defined APIs
  - Open Sensor Application API
  - Open Sensor Service API
  - Open Sensor Device API



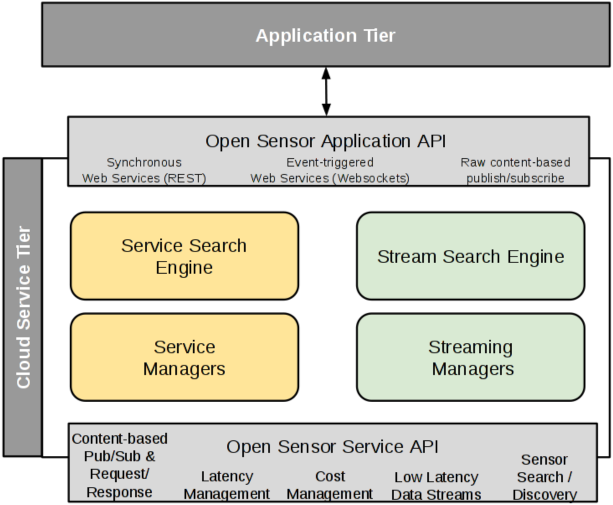
Cloud Service Tier

Simple development of IoT applications

- Fast and flexible discovery
  - Discovery of services
  - Discovery of data streams
- Efficient data delivery
  - Synchronous
  - Event-triggered
  - Low-latency streaming
- Value-added services
  - Aggregation of multiple streams
  - Specification of composite events
  - Query injection and reconfiguration

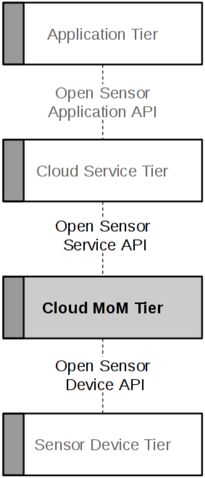


Main Cloud Service Tier Components

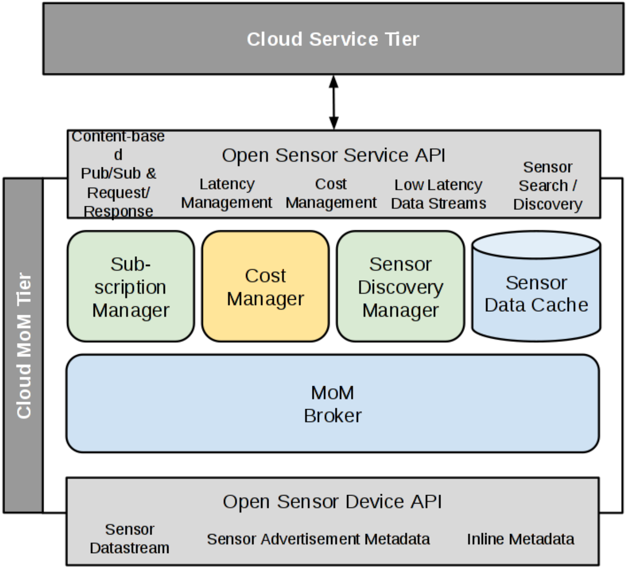


Cloud MoM Tier

- Efficient data dissemination
  - Flexible publish / subscribe service for dissemination of data and control metadata
- Dynamic cross-layer optimization
  - Merging of overlapping sensor data queries and piggybacking of notifications and advertisements
  - Calculation of a query "cost" metric based on expected system efforts
  - Expressing latency requirements allowing intelligent caching and larger freedom to reschedule/combine/defer queries

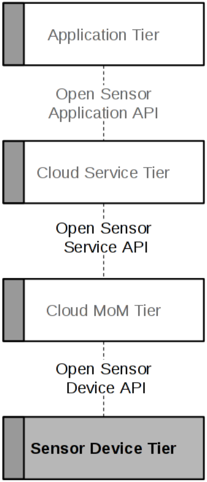


Main Cloud MoM Tier Components

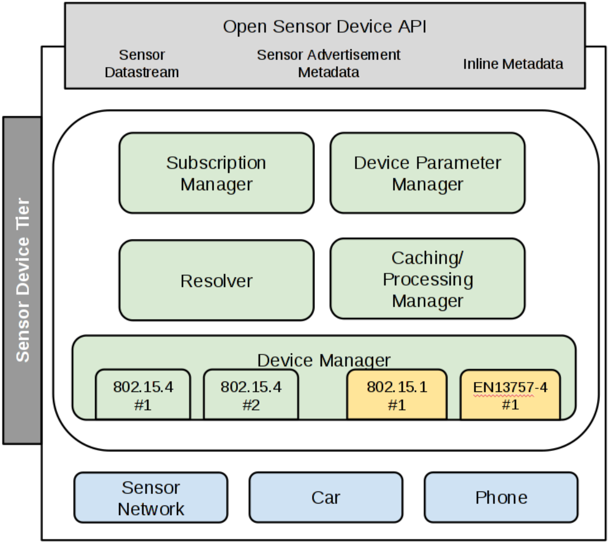


Sensor Device Tier

- Integration of heterogenous sensor technologies
  - Standardized data interface, sensor description and discovery
- Horizontal and vertical discovery
  - Based on proactive advertisements
  - Inverts the current poll/crawling method
  - Metadata for parametric-based search
  - Low-frequency data for data-based search
- Device network optimization
  - Flexible data caching
  - Dynamic device parameter adaptation



Main Sensor Device Tier Components



End.

Thank You!

Comments/ Questions?

Survey of Cloud-centric IoT Platforms

Thomas Menzel, Niels Karowski, Daniel Happ, Vlado Handziski, Adam Wolisz
Technische Universität Berlin, Telecommunication Networks Group (TKN)
{menzel, karowski, happ, handziski, wolisz}@tkn.tu-berlin.de

***Summary*—The Internet of Things (IoT) platforms support rapid development of applications focused on gaining enhanced awareness and control of the physical environment. In contrast to the traditional, vertically integrated model, IoT platforms aim at supporting a much broader set of applications on top of a shared hardware base of diverse sensing and actuation devices. In this work we survey several IoT solutions and extract their features along different design dimensions. We address several limitations in these existing platforms. Finally, we present our recent work on a novel IoT platform that deals with these shortcomings, developed in the context of the Social Sensor Cloud project.**

I. Cloud-centric IoT Platforms

In the envisioned Internet of Things (IoT) solutions, huge amounts of data from billions of interconnected devices will have to be distributed in an efficient manner, imposing new challenges on the communication and processing infrastructure. Cloud computing is well positioned to meet gracefully these requirements thanks to its flexibility, reliability and usage-based cost model. We therefore expect future IoT platforms to be cloud-centric with a similar general architecture as the one shown in Figure 1 [1]. At high-level, sensor data from the device tier is sent to a cloud-based service tier, which provides access to the data as well as additional services like data storage, analysis, aggregation, etc. to user facing applications.

In Section II, we present the results of our survey of several prominent academic and commercial IoT platforms. We extract the most important characteristics of their Application Programming Interfaces (APIs) that capture the diverse requirements imposed on the service tier. In Section III, we briefly introduce the Social Sensor Cloud (SSC) [2] project in which we have defined a new IoT architecture that addresses identified shortcomings in the existing solutions.

II. IoT Platform Analysis

We have surveyed several prominent IoT solutions and extracted their features along different design dimensions, e.g. platform architecture, offered value-added services, and application side platform access. Table I provides a list of the surveyed IoT platforms.

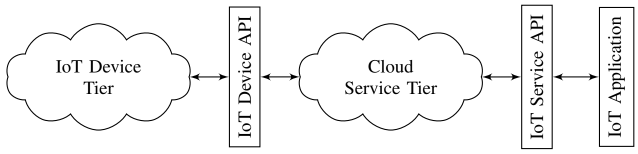


Fig. 1: General architecture of a Cloud centric IoT platform.

The vast majority of surveyed platforms explicitly *differentiate between data generators and consumers*, offering separate APIs for interacting with the devices from one side, and the served applications from the other. As a result, their coarse architecture is similar to the one depicted in Figure 1. Xively is an example of a different approach that uses a unified API towards the generators and consumers of data. In almost all platforms, *gateways* are used to mediate and integrate devices that lack native IP connectivity. Despite their importance for implementing Cyber-Physical Systems applications, *actuator* devices are explicitly supported only by few academic solutions.

Value-added services such as data storage, virtual sensors and device management, represent another important dimension for comparison.

Data storage services archive information generated by devices as part of the offered services of the IoT platform. The storage of data allows IoT providers to offer additional functionalities such as historical search operations and data aggregation. This service is supplied by the majority of the analyzed IoT platforms, even though some platforms (e.g. openHAB) do not provide integrated storage solutions but resort to transparent integration with external services to achieve the same goal.

The concept of *virtual sensors* enables the combination of multiple sources of information into a composite data source. Data generated by different physical devices can be processed and presented as a new source of information. A common example of a virtual sensor is the computation of the dew point. Information about temperature and relative humidity measured by different sensors can be combined in order to compute the dew point and the result can be offered as a new data source, a virtual dew point sensor. In the surveyed platforms, the concept of virtual sensors is mostly prevalent in the area of research-oriented solutions, e.g. Sensor Andrew, and standardization activities (ETSI M2M, OSIOT and OpenIoT), while it is less represented in the commercial area (with the exception of the open-source openHAB platform).

Due to the envisioned high number of connected devices an important service of an IoT platform is to support users in managing their devices, starting from adding and removing single devices to handling batches of devices. Such *device management* services are offered by all analyzed IoT platforms.

Providing uniform access to the gathered, processed and stored device data is one of the core responsibilities of every IoT platform. We identified *protocols* to interact with the platforms, *encoding* format and *interaction* pattern as key features for characterizing *application-side platform access* of the IoT platforms as shown in Table I:

- Protocols – All current platforms support RESTful HTTP access. Only a selected few also offer more efficient access via (Web)Sockets or raw MoM interfaces.
- Encoding – XML and JSON are the leading data encoding formats and many platforms support both.
- Interaction – In addition to simple request/response commands, many platforms also offer more flexible interaction patterns and mechanisms such as subscription to configurable data feeds or streams and notification in case of application defined events using push messages.

TABLE I: Application side platform access.

	Protocols			Encoding			Interaction		
	HTTP	RawMoM	Other	XML	JSON	Other	Req/Resp	Pub/Sub	Push
ETSI (one)M2M [3]	✓		CoAP	✓	✓		✓	✓	✓
Xively [4]	✓	✓	Socket	✓	✓	CSV	✓	✓	✓
Etherios [5]	✓			✓	✓		✓		✓
SENSEI [6]	✓		SIP	✓			✓	✓	✓
Sensor Andrew [7]	✓	✓		✓		CSV	✓	✓	✓
FI-WARE [8]	✓			✓	✓		✓	✓	✓
OSIOT [9]	✓	✓	CoAP		✓		✓	✓	✓
Axeda [10]	✓	✓	propr.	✓	✓	ASN.1	✓	✓	✓
OpenIoT [11]	✓		Socket	✓	✓	CSV	✓	✓	✓
EVRYTHNG [12]	✓				✓		✓		
RuBAN [13]	✓			✓			✓	✓	✓
openHAB [14]	✓	✓		✓	✓		✓	✓	✓
ioBridge [15]	✓				✓		✓	✓	✓

While some of the evaluated platforms make use of the underlying networks' QoS capabilities, only Axeda uses message priorization and only OpenIoT targets *QoS support* between the application and device tiers. Furthermore, only SENSEI and OpenIoT try to increase the platform efficiency by leveraging *cross layer/tier optimization* approaches.

III. Social Sensor Cloud

We have so far described several academic and commercial IoT platforms. Nevertheless, we believe that there are some important shortcomings in these existing solutions. Most platforms offer limited service APIs that lack support for event-triggered interaction patterns. Their request/reply and polling focus leads to unnecessary sampling, high latency and communication costs especially over bandwidth limited links and reduced lifetime of battery-powered devices. The majority of platforms also follows a very rigid decoupling between the different architectural tiers, thus missing on opportunities for cross-layer/tier optimization such as expression of latency, reliability and cost requirements that can be used to improve the efficiency of the platform, especially in the device tier with many battery-operated devices operating under constrained communication and energy budgets. Finally, most of the platforms follow a *storage-centric* operation model focused on archival of all data generated by the connected devices, an approach that is not scalable to the massive number of envisioned interconnected devices, due to the high initial investment and operational costs for storage.

In the context of the Social Sensor Cloud (SSC) [2] project, we are currently working on a novel IoT platform

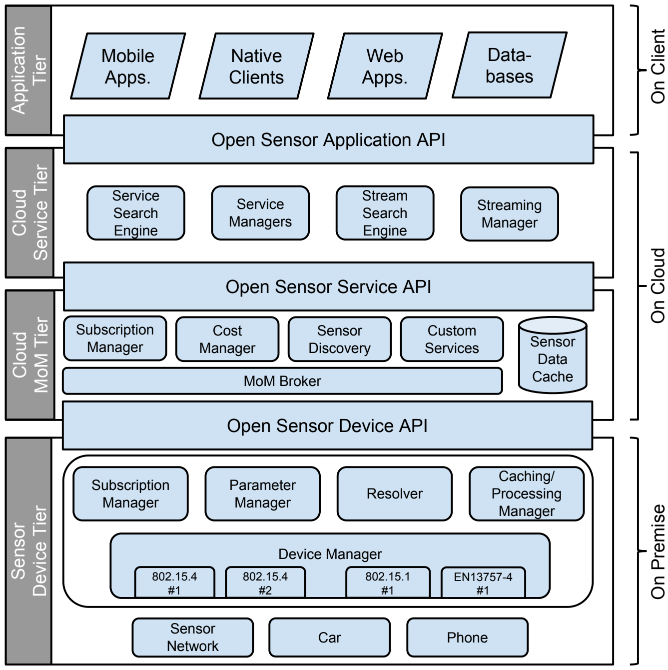


Fig. 2: SSC architecture overview.

that addresses these shortcomings from the point of view of scalability, efficiency and supported services. The architecture of the platform is based on the functional decomposition in four tiers as shown in Figure 2.

The highest tier of the architecture, the *Application Tier*, encapsulates the different user-facing applications to be developed on top of value added services provided by the Cloud Service Tier. It therefore technically does not belong to our proposed platform. To meet different application requirements, developers may express latency requirements as well as other QoS parameters using the Open Sensor Application API.

The *Cloud Service Tier* (CST) reduces the complexity of developing IoT applications residing in the Application Tier by hosting cloud-based value added services and providing fast and flexible data stream and service discovery. The *Service Search Engine* provides the search possibility of the value added services, e.g. aggregation of multiple data streams, possibility to specify composite events as well as query injection and reconfiguration. Each provided service is represented by a *Service Manager*. The discovery of data streams is a native service supplied by the CST through the *Stream Search Engine* and *Streaming Manager*. The *Open Sensor Application* API of the CST offers a synchronous REST-based web-interface as well as a Websockets interface which allows applications to receive event-based push notifications using pre-defined expressions instead of applying continuous polling to check for updates. Furthermore, in order to guarantee efficient data delivery besides supporting synchronous request/response interaction and event-triggered notifications, a raw content-based publish/subscribe interface provides access for high-performance applications.

The *Cloud MoM Tier* (CMT) is the central tier supporting efficient data dissemination using a flexible publish/subscribe service as well as point-to-point communication for the distribution of data and control metadata. The central component of the publish/subscribe service is the *MoM Broker* which connects producers and consumers of data. A major goal of the SSC platform is the shared utilization of sensors by multiple applications. Requests of applications with specified and similar QoS requirements are merged by the *Subscription Manager* to avoid unnecessary sensor sampling by devices. To prevent applications from using the highest QoS always, the *Cost Manager* may be utilized to associate "costs" to e.g. each data retrieval. A *Sensor Discovery Manager* subscribes to pro-active advertisement messages transmitted by devices in the Sensor Device Tier and stores the obtained data in a *Sensor Data Cache* in order to provide sensor search functionality. Historical search is enabled by low-frequency data streams periodically sent from devices. To support the huge number of envisioned IoT devices, only aggregated data, e.g. minimum, maximum, average, is stored in the CMT.

The *Sensor Device Tier* (SDT) enables not only the integration of heterogeneous device technologies using a standardized data interface, but also provides the functional basis to support the novel services of the upper SSC tiers. Application-side QoS requirements are managed by the *Subscription Manager* while the *Parameter Manager* stores the controllable parameters of the connected networks and devices. The matching is performed by the *Resolver* and the resulting parameters are applied by the *Device Manager* and the *Caching/Processing Manager* who forwards the data accordingly to the upper tier via the *Open Sensor Device API*. Instead of applying the often used method of polling and crawling for new devices, the device discovery is based on pro-active advertisements.

IV. Conclusions

In this paper we have surveyed several prominent IoT solutions and extracted their features along different design dimensions such as the properties of their platform architecture, offered value-added services, and application side platform access. All current platforms support RESTful HTTP access and few also offer more efficient access via (Web)Sockets or raw MoM interfaces. XML and JSON are the leading data encoding formats. We claim that there are several limitations in the existing IoT platforms such as lacking support of QoS and cross layer optimization. In the context of the Social Sensor Cloud project, we are currently working on a novel IoT platform that addresses these shortcomings from the point of view of scalability, efficiency and supported services.

References

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, 2013.

[2] Ssc. [Online]. Available: http://www.azeti.net/social_sensor_cloud.html

[3] Etsi m2m. [Online]. Available: http://www.etsi.org/technologies-clusters/technologies/m2m

[4] Xively. [Online]. Available: http://xively.com

[5] Etherios. [Online]. Available: http://www.etherios.com

[6] Sensei. [Online]. Available: http://sensei-project.eu

[7] Sensor andrew. [Online]. Available: http://sensor.andrew.cmu.edu

[8] Fi-ware. [Online]. Available: http://www.fi-ware.eu

[9] Osiot. [Online]. Available: http://osiot.org

[10] Axeda. [Online]. Available: http://axeda.com

[11] Openiot. [Online]. Available: http://openiot.eu

[12] Evrythng. [Online]. Available: http://evrythng.com

[13] Ruban. [Online]. Available: http://www.davranetworks.com

[14] openhab. [Online]. Available: http://openhab.org

[15] iobridge/realtime.io. [Online]. Available: http://www.iobridge.com