

Future platform for Internet of things

This article presents (summarizes) a vision for the future Internet platform based on the results of the long time cooperation between MSU and AbavaNet company. During the long time we were involved into telecom services development, recently our activity switched towards smart metering and sensors. And all that time the main areas of our interest were the software platforms as well development tools for services creation. So this article describes our current vision based on our past and current experience in this area.

In our vision the new services /applications will be based around the Internet of things concept. Practically it is a set of sensors / intellectual devices that capable deliver some real time measurements and/or context-related data by either by user's requests or by the requests from other devices.

Of course, the proper platforms as well as the right development tools are key point. Let us see the history of telecom services development. We took telecom services as they are most natural predecessors for Internet of things. There we can see the following stages: the complete lack of API, proprietary API's from various vendors, efforts to create some united API (e.g. Parlay), but all that, to be honest, without the big success. The main reason from our point of view is that API's (proprietary or public – does not matter) were always outside of the modern trends in the development. Everything was switched to the web, telecom API's does not. Web development does REST, Telco's are still pushing SOAP etc.

And despite any efforts to unify API's they are still fragmented.

The only possible solution by our mind is change for the paradigm. We really need to switch from the set of individual API's to the holistic platform. The future Internet platform by our mind will play the same role as modern middleware applications. Simply, we see the platform as a middleware expanded to the whole Internet. So there is simply no space (read – no reasons) for the private API's. And because there is no single company that could develop all, as well as no one service that could be positioned as a killing service, the core technology platform should be a mashup platform. It should allow easily mix various data sources/services (via respective access methods) and quickly build (configure) end-users services.

Let us describe the attendant modern technologies that will form a foundation for the future platform.

At the first hand it is cloud. Cloud based services should form a foundation. As the amount of services (read – sensors) will grow the scalability inevitably becomes the key point. There are no ways anymore to get a reliable service from any standalone host. And we definitely need a middleware cloud. Suppose we've created some new function (library). What if, instead of repeating that function each time a new server is created, you could generate one version that saved somewhere in the middleware and provided the same capability to all. Instead of needing some new functionality embedded in each our server, we can describe this functionality just once and then let each server pickup it from a net, when needed. It is a basic principle for cloud software that any software resource will be instantiated just once but used many times.

No-Sql approach for databases. It is absolutely explainable too. There is no way to predict the possible processing or specify once all the attributes, most of the services will be formed “on the fly” – so we definitely need to follow schema-less way

JSON as a prevailed data-type. Even more – JSONP. To be the true dynamic most of the processing in the future services will be performed on the end-points (end-users) devices, ad-hoc service composition and mashups will require the same. Dynamic languages (read – JavaScript), no server side processing – JSON is definitely a winner here. And JSONP let us process data at end-points, using servers just for transport initiations.

Meta formats and self descriptive devices. Here, from our point of view, such things as SNMP and its MIB approach especially are underestimated. If you have many sensors and new types added monthly (if not weekly) the only solution to add them to the net (integrate them into services) in the reasonable time (for the reasonable money) is to have some form of formal descriptions for their features.

Examples. Lets us list some examples for the services we are talking about.

- Environment maps. E.g. city officials or citizens can see maps where each building colored by its environment impact (e.g. energy usage and environment influence). And live aggregating data are available as well. For the technological point of view we need to automatically discover home sensors as well as create and support large databases (data stores) with flexible schemas
- Parking lot sensors let us pickup free spaces info right on the mobile
- The same sensors automatically “check-in” visitors at places so we can see “buzzed” areas in the real-time

- live population/traffic spread across the city (e.g. based on the presence sensors and/or active phones)
- the above mentioned maps (snapshots) with the time scale (how the thing are changed over the time)
- the various geo-fences that inform us during a day about events, offers, conditions etc. in areas we are in (about to enter or leave). And setting rules (triggers) for them technically is a task for cloud

Dmitry Namiot – currently is Senior Researcher at Moscow State University. Ph.D in Computer Science. 20+ year professional experience in software development and telecom services.

Development awards: Simagine Best of Technology, Java Magazine readers choice, Simagine Innovation Award and others.

Business areas: JEE applications, web development, databases and data processing, open interfaces for telecom (Parlay, SIP), mobile services and mashups.

E-mail: dnamiot@gmail.com