

Beyond API management - towards development engagement

Submitted on 06 Nov 2013 by Jarkko Moilanen

In Finland more and more data is opened every day. There are thousands of spreadsheets, and other documents dumped in several open data catalogues such as HRI, Open Data Tampere and the Finnish Meteorological Institute. CSC — IT Center for Science is also planning to launch a CKAN based open data catalogue for scientific data. In other words, data is there.

Another thing is APIs or Application Programming Interfaces. More of these are being released in Finland, not weekly but more likely monthly. API management is a subject that I've tried to raise in discussions in Finland. Some have seen the need for focusing on API management solutions for the future. It is crucial to think about the APIs and identifying needs around them before the legendary stuff hits the fan.

Below are two models which both seem possible for any country, but as an example I've used Finland as a case. The introduced models should not be seen as being mutually exclusive, but more as examples of different approaches. Furthermore, there are plenty of other models, but the intention is to raise awareness and attention to open data API management related issues beforehand. In addition, Finnish national level open data catalog is going to be released in a few months and next logical step is to manage not just data but also APIs.

In my opinion, we should broaden the viewpoint even more. It's not just about API and data management. Instead, it's more and **more about development engagement**, which refers to:

- civic hacking and
- civic engagement to service and application development.

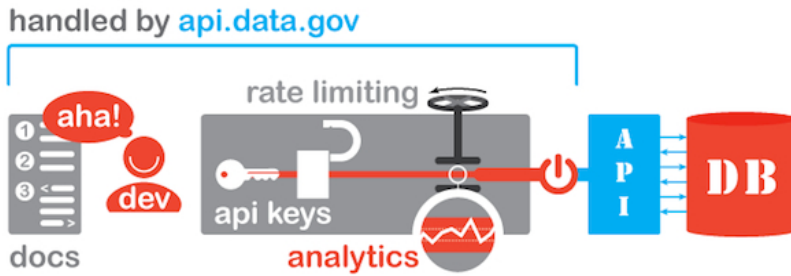
APIs play a crucial role in enabling civic hacking, which is according to Gartner, is [expected to produce 25% of business applications next year](#). That's why we all should be so enthusiastic about APIs and civic hacking. Those two items will most likely be the foundation for the next generation of services and applications.

Founding values

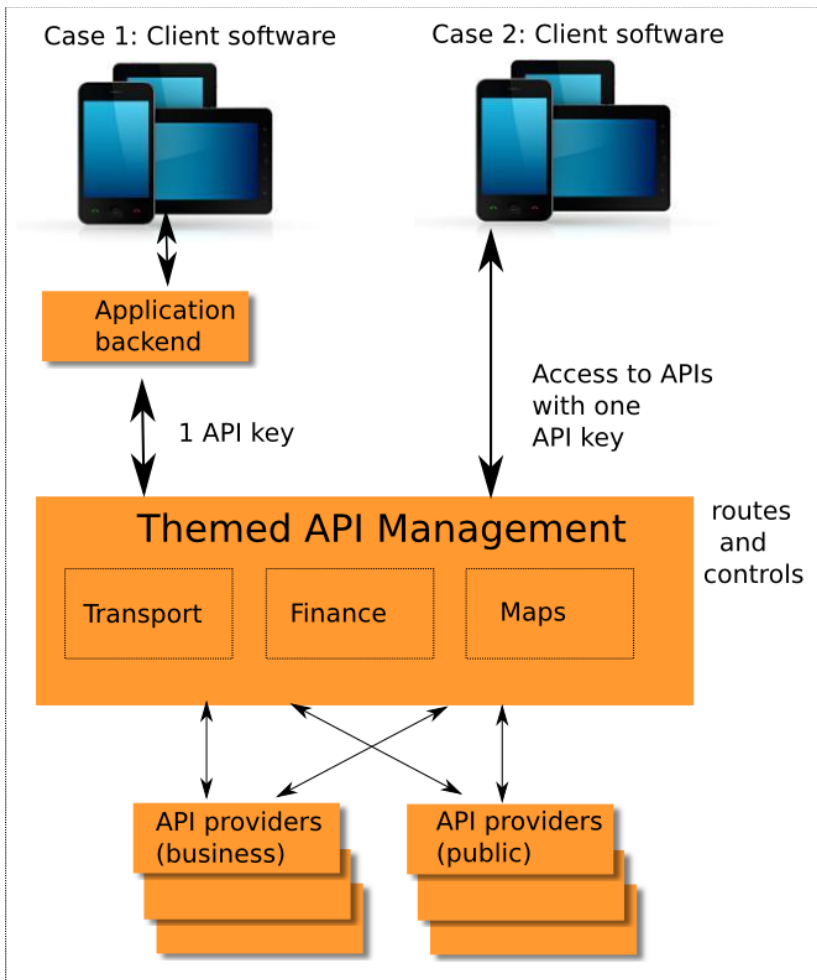
But let's take a look at the API management models. Whatever the selected management model is, the starting point must be the application and its developer. The starting point must not be the political structure or other invisible boundaries such as the borders between counties. The premise cannot be that public sector and business originated APIs are differentiated under some miscellaneous management solutions. Application developers or apps do not care about the data public sector or business sector origin. The difference rarely makes any added value to apps or developers. Of course there will be "rogue" APIs, which are under no management.

Central API management model with API Umbrella

First thoughts about API management in small countries like Finland tend to go towards a central national solution which is handled by government. This is also the model that the US government has chosen. The US government chose the "API Umbrella" as a technical solution for managing APIs under <http://api.data.gov>.



[API Umbrella is an open source](#) API management platform for exposing web service APIs. The basic goal of API Umbrella is to make life easier for both API creators and API consumers. API Umbrella allows admins to create documentation for individual web services and organize them into hierarchical collections. In the below illustration is an example of central API management. The image contains two developer examples. Case 1, which contains apps with backend (server), and case 2, which is simple app/apps that access APIs directly. Both cases are normal use cases for APIs. If all available APIs are put under one management (for example one API Umbrella instance), the APIs could be divided to collections (themes).



In either case, developers need to get an API key or possibly multiple keys from a single point. In the case of Finland, there is a question regarding who this organization is running API management for. For government? And with which resources?

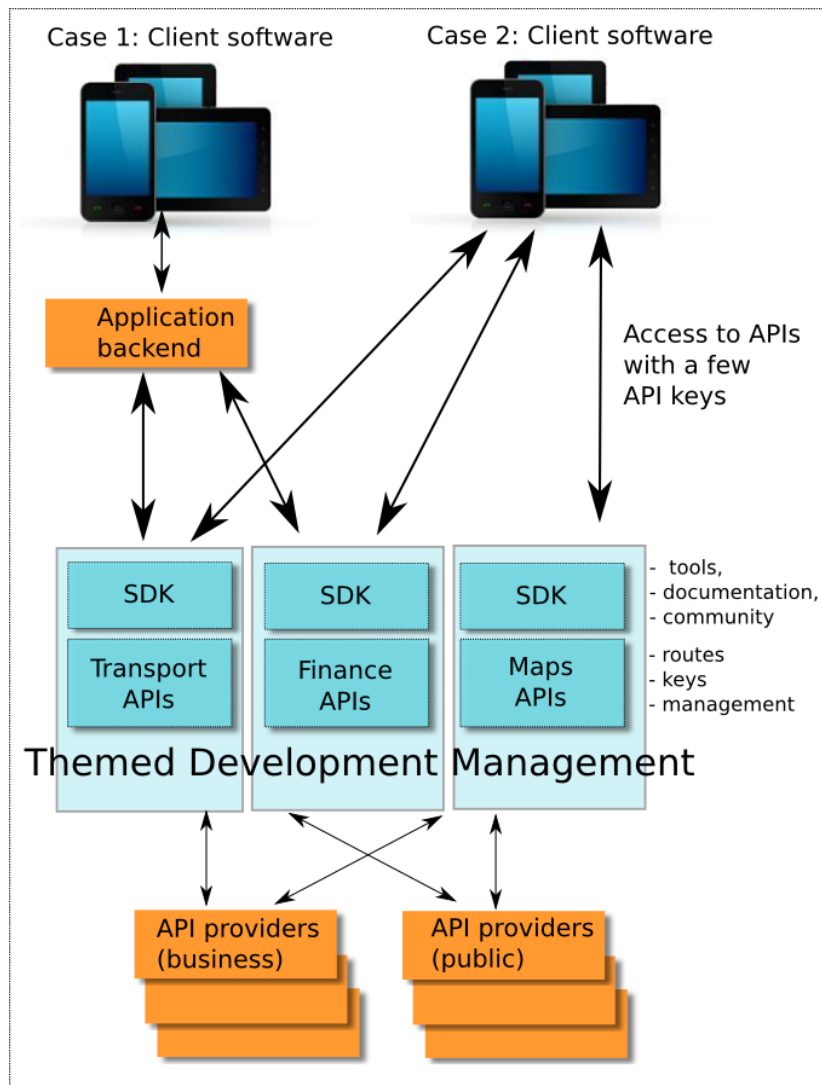
I highly doubt that government will be able to handle API management in the near future. Establishing CKAN based open data catalogue has shown us that the Finnish government is not very agile and API management discussion has not even emerged yet in Finland. Even if the government

were the organization maintaining the API management, I doubt that they would allow lightweight processes to occur on that platform. Refusing API key for one app or developer probably would require some multi step process which takes days. In addition, having single point to access all APIs would also pose risks. Nevertheless, this model is one option.

DTDM - Distributed Themed Development Management with SDKs

Yet, I argue that focusing purely on API management solutions is only a partial fix. APIs are key elements in future applications and therefore we should encourage API SDK development. Software Development Kit (SDK or "devkit") is typically a set of software tools that allows for the creation of [applications](#) for a certain [software](#) package, [software framework](#), hardware platform, [computer system](#), [video game console](#), [operating system](#), or similar development platform. APIs are essential, but not enough on their own. Building API SDKs will make application development faster and more accessible for variety of developers.

Distributed themed development management is one option of organizing API and development in the fields of open data. We should not separate public and private sector data flows (APIs), nor should we enforce separation of public sector data and data created by crowd-sourcing. Instead we should seek solutions in which all data can be easily combined and used via some set of tools (SDK).

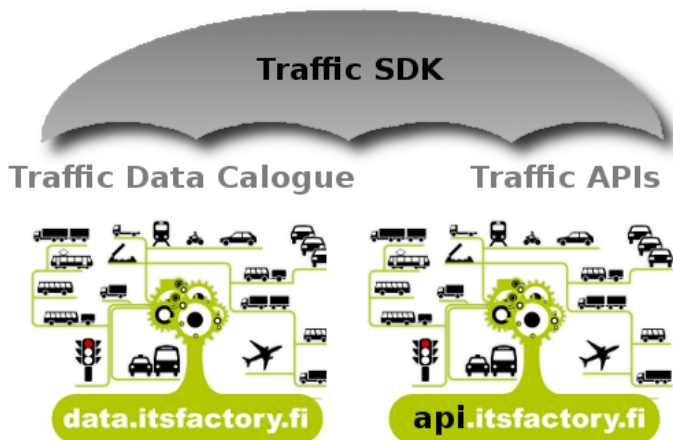


CitySDK is one example of "themed" SDK. The CitySDK aims for a step change in how to deliver **services in urban environments**. With governments around the world looking at open data as a kick start for their economies, CitySDK provides better and easier ways for the cities throughout Europe to release their data in a format that is easy for the developers to re-use. The project of taking the best practices around the world will foresee the development of a toolkit – CitySDK v1.0 – that can be used by any city looking to create a sustainable infrastructure of “city apps”. The above API management model follows the same logic, but is not limited to urban environments; it acts more in national level. After all, apps are often targeted at a national or global level.

In DTDM we need several organizations that are willing to take responsibility of management services needed (infrastructure). Each of the themes is a separate system, which can discuss with each other just like CKAN nodes. API's are added to suitable themes by API owners, which also manage their APIs in the service. There would not be any kind of centralized body to manage all APIs even in this model. It would be wise to list APIs at the "Finnish Open Data Portal" under APIs, just like datasets are listed. Yet, the management would not be in the hands of central governmental organization. Again, a suitable open source solution in this case as well as in above is "API Umbrella". These themed API management nodes would be quite similar to "themed open data catalogues" which are [one open data catalogue arch type](#).

Selecting hosts for themed development management

An example of such organization for handling one themed API management node is [ITS Factory](#). Their intention is to become "the traffic solution node" in Finland. They could establish API management solutions in their servers for "Traffic/Transport APIs".



Every developer who creates apps which need "traffic APIs" would acquire the API key from them. Of course the organizations which would maintain such important management nodes, would have to have solid base and ability to guarantee continuum so that after one year of service, they would not be forced to pull the plug because they ran out of funding.

The hosts would:

1. handle the infrastructure for selected theme API management, and also
2. lead the theme related SDK development.

Leading the development does not equal doing all the work. Instead, it refers to building roadmaps for the SDK, using some resources (Code 4 Europe?) for developing the SDK and getting community engaged. This kind of broad SDK development should be shared among all countries

and thus broaden the developer and user base.

Shared user base

User authentication and rights management could be shared among the API management nodes (themes). In other words, developers need to register just once and reuse the same account on all themed API management nodes.

But what about the APIs? Should we be able to tell a little more about each API to developers? The answer is yes; we need open data API evaluation tools, about which I have modeled. I will publish the model in another blog entry later on (on the ePSI Platform).