The MPA uses OrientDB to help fight crime

1. What is your organization and what does it do?

The [Ministerio Público de la Acusación (MPA)](http://www.mppba.gob.ar) is the public prosecutors’ office of the province of Santa Fe, Argentina.

2. What are some of the key technical challenges that the MPA faces?

The amount of data the MPA collects when dealing with criminals is huge. If one wants to map the relationships between intra- and inter-case elements, the amount of data grows even larger. Traditional RDBMS fail to address this critical problem.

3. Tell us more about the application you developed and how graph DBs powered key parts of it. What was the goal of the application? What were important technical requirements?

The application is in the design and development process. We hope to release the first version in the second half of 2018. We need to store brief information within the cases that includes details such as the accused, victims, participants, prosecutors, defenders, elements, sites, times, etc. All relations between entities are mapped as vertexes and edges. In this way, we will be able to perform a search on the type of relations that exist between entities in a case and most importantly, if there is a relations with the entity and other cases. Ultimately, this will enable us to see the entire map of what is going on. Another important thing is that the prosecutor needs to know if there is any existing information about any item that registers in the system and be notified of any future appearances of that item in other cases. There is no clear way of determining how you relate the elements in a case and this implies that the traditional query languages are not sufficiently expressive. With regards to this, the Graph/TRAVERSE statement can be helpful.
4. Why was the application important? How did the application aid in fighting crime?

Currently, a prosecutor is in charge of more than 400 cases and there have been more than 600,000 cases in total over the last three years. This makes it impossible for one to remember without equivocation all relationships between all elements in a case and even less so to relate to elements of other active or closed cases. Here is where our apps can be helpful: they will be able to show all information that existed in the past and register any objects that belong to a case in the future.

5. How did you ultimately settle on OrientDB as your vendor?

We started creating a plan in 2014. At the beginning, we found Neo4J. Our conceptualization of the problem pointed to graph databases as an ideal solution. Following that, we began to research if there were other similar solutions to utilize. At this stage, we found OrientDB.

Our province has a law that states that free software must be used in any development and Neo4J did not have a license that worked for us. However, OrientDB did. So, after switching to OrientDB, we were eager to see if it could do the job. We tested an initial application, posted a few queries on Stack Overflow and everything worked well. From there, I hired my team and directed them to develop an application using OrientDB.

6. From a technical standpoint, why was OrientDB ultimately best suited to power this application?

OrientDB has a very simple way for people that come from SQL to do a switch. The way you write a query is very similar but the power lies in using a specific graph function to search the database. It is open source and written in Java so you can run it on any server. The availability of the out of the box cluster configuration, even in the community version, was the characteristic that ended up being the deciding factor to use OrientDB because it guaranteed that we could grow without problems if we needed to.