

Exposing MySQL Shell's backend to multiple programming languages

Ricardo Ferreira

Consultor externo: Miguel Araújo (Oracle)
Supervisor: Fernando Miguel Gamboa de Carvalho

1. Some background

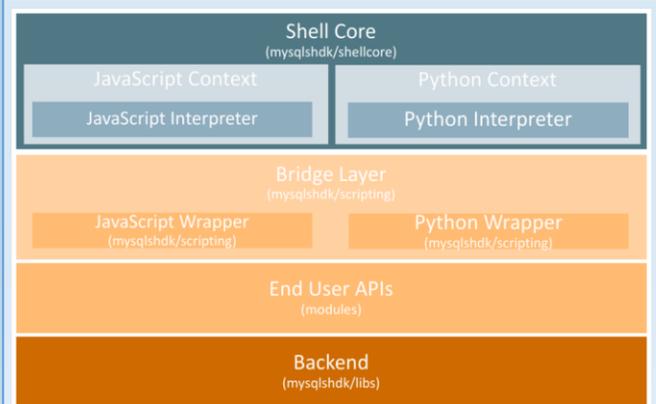
MySQL is an open-source DBMS currently developed by Oracle. In today's world of cloud computing and devops (a term used to denote a methodology in which a software performs both development and system administration/operational duties), there is a need for high-availability and scalability. MySQL has some features for creating and managing highly available clusters of instances, but these are not easy to manage, requiring often specific expertise in MySQL intricacies. This led to the development of InnoDB Cluster - an integrated high-availability solution that makes use of these features and exposes them to the user through a new MySQL client, called **MySQL Shell**.

2. Purpose of the project

The purpose of this project is to allow **MySQL Shell's** capabilities for managing highly available clusters of MySQL instances to be usable outside of the client itself. This will be done by **exposing** these capabilities in a **shared library** and creating **language bindings** for this **shared library**. The languages supported shall be Javascript, Python and Go.

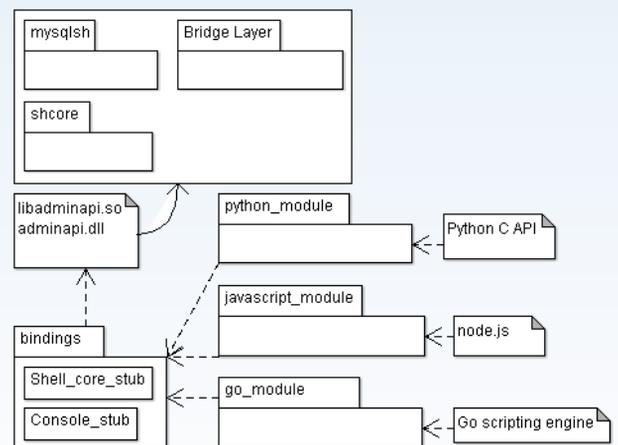
4. Conclusions

Unfortunately, in the current state-of-the-art, only the Python bindings have been developed, and these are still not working. The initial assumptions on the architecture turned out wrong, as well as the experiments on generating bindings and wrappers automatically. So, to summarize, the current beta only fulfils the first functional requirement (exposing **AdminAPI**).



MySQL Shell is separated into layers. This project will expose the logic contained the **End User APIs** and **Backend** layers and will use the parts of the **Shell Core** and **Bridge Layer** to create the language bindings.

3. Progress so far



The high-availability capabilities have been extracted and compiled onto **libadminapi**. Some dependencies were found inside that code that don't make sense in the bindings (like printing to console) and stub classes were created to subvert this behaviour and be able to use the existing code without changing it altogether.