

Faceted Data

What is faceted data?

Various functions could inspect data. However, we can design a system that, if we label a security level to the data, a mechanism in the system can determine function accessibility. Only a function with adequate permission can operate on sensitive fields in the data; otherwise, it restricts only access to nonsensitive information. For example, a decrypted password can be accessed by the comparison function. But tasks like printout, write to file part, or assigned to another variable are disallowed. When the system detects the violating cases, the calling function can be terminated, or the protected data returns a non-inference of public nonsensitive data to outside.

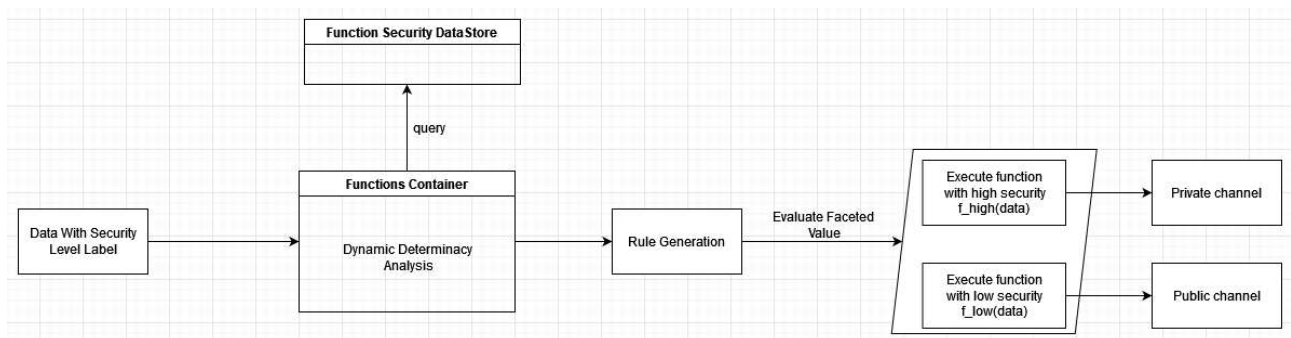
People can get benefits from this approach:

1. Data will not be leaked by accident, for example, logging sensitive data to the log or returning sensitive information by the wrong function called.
2. People can fully control data in whole processing. Even data with sensitive information passed into a lower security function, no need to worry about data leaking.

But faceted data could have drawbacks like:

1. "Statically computing determinacy information, however, is very challenging, especially for dynamic languages like JavaScript". (Schäfer, M., Sridharan, M., Dolby, J. and Tip, F., 2013)
2. Introduce extra overhead when evaluating data security, especially if it could run multiple times when having various security policies.

Facted Data in Python Way



Reference:

1. Schäfer, M., Sridharan, M., Dolby, J. and Tip, F., 2013. Dynamic determinacy analysis. *Acm Sigplan Notices*, 48(6), pp.165-174.