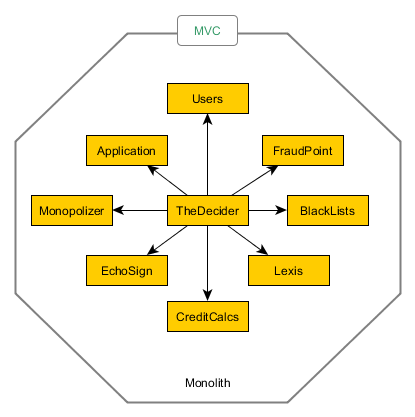
Development Plan

# Refactoring Order

1. ***Applications => Applications Microservice***
2. ***EchoSignServcie => Microservice Platform Build***
3. ***Monopolizer => Microservice Platform Build***
4. ***FraudPoints => FraudPoints Microservice (for integrations)***
5. ***BlackListIps => Black Lists Microservice***
6. ***Lexis => Lexis Microservice***
7. ***CreditCalcs => Credit Calcs Microservice***
8. ***The Decider = > Rules Microservice***
9. ***Login Microservice***

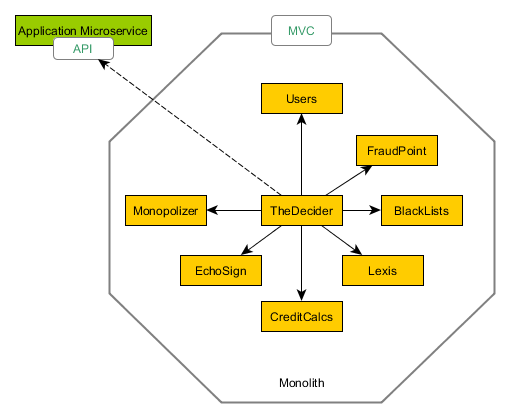
# Remove TheDecider coupling

The image below reflects the coupling between ***TheDecider*** and the rest of the modules.



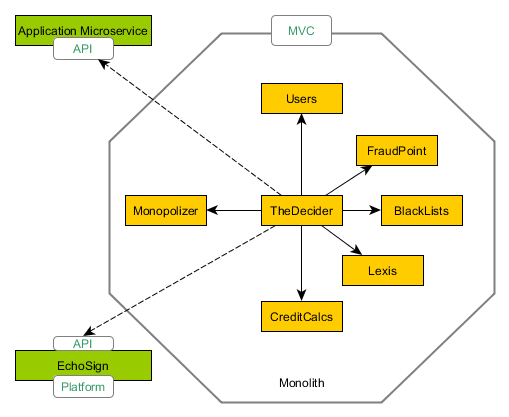
# 1 – Applications Modules to Applications Microservice

Isolate Applications into its own microservice.



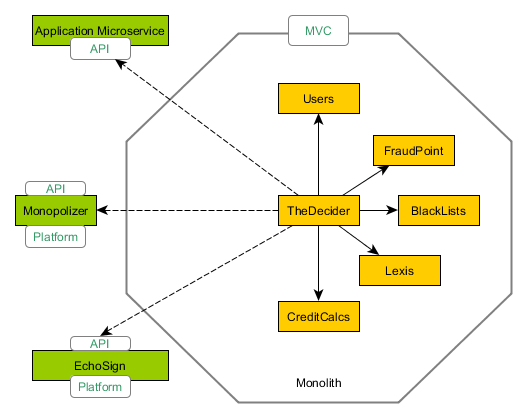
# 2 – Remove Echosign dependency on CommonConfiguration (Tight Coupling)

This step includes removing monolith libraries from EchoSign and moving ***CommonConfiguration*** into a nugget package so every new service has not to have a submodule installed in its repository. That makes development easier.



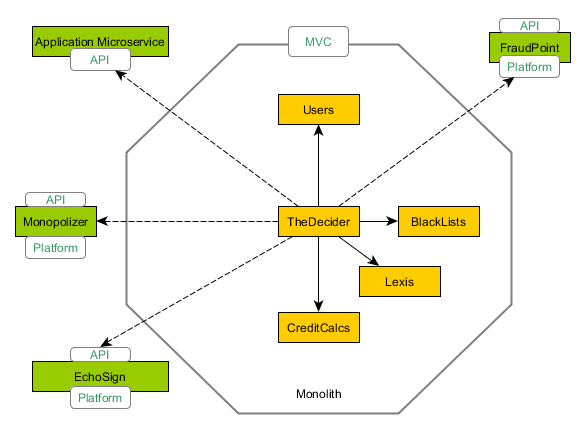
# 3 – Remove Echosign dependency on CommonConfiguration & CommunityChest (Tight Coupling)

* Remove Monopolizer CommunityChest Dependency
* Replace the reference to the CommonConfigurationSubmodule by the nugget package implemented in step 2.

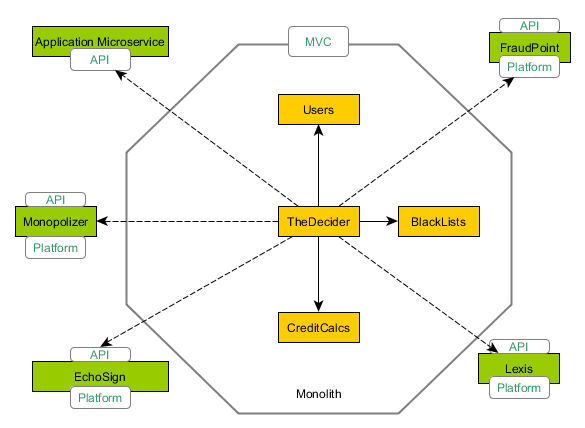


# 4 – Move FraudPoint to a Microservice

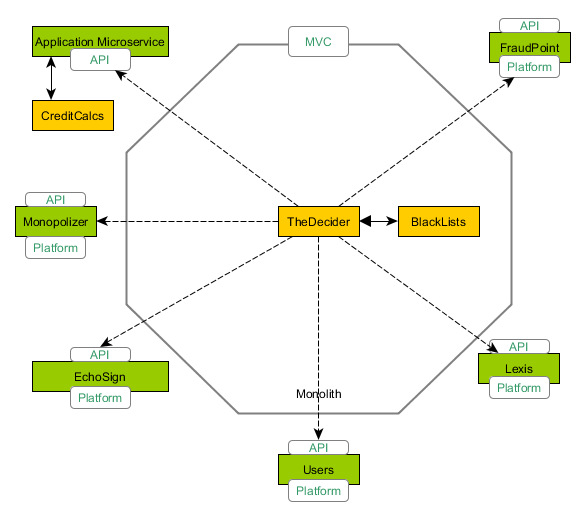
This is a ***technical capability*** that represents “integration” and it should be moved out from TheDecider.



# 5 – Move Lexis Module to Lexis Microservice

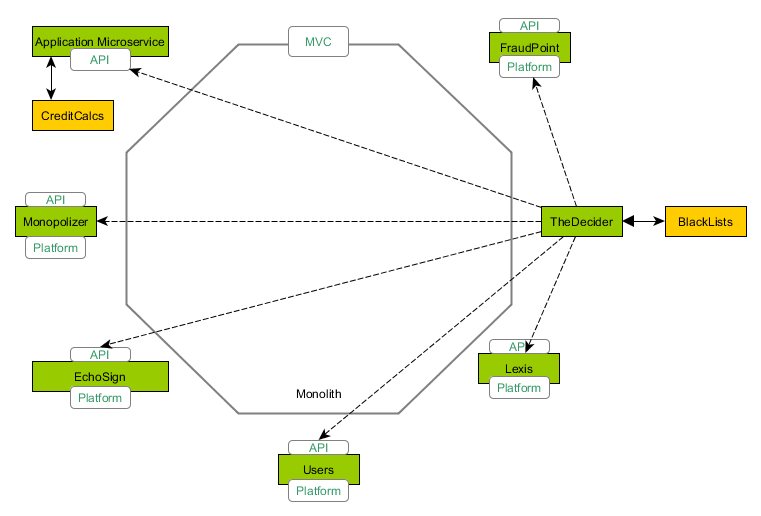


# 6 – Create Identity Microservice



# 6 – Create Rules Microservice

Remove TheDecider outside the monolith.



Once all functionalities starts to be isolated the MVC application will start becoming the API Gateway i.e. the Public API.

***Next step includes:***

* ***Moving out Contracts***
* ***Service to service security***