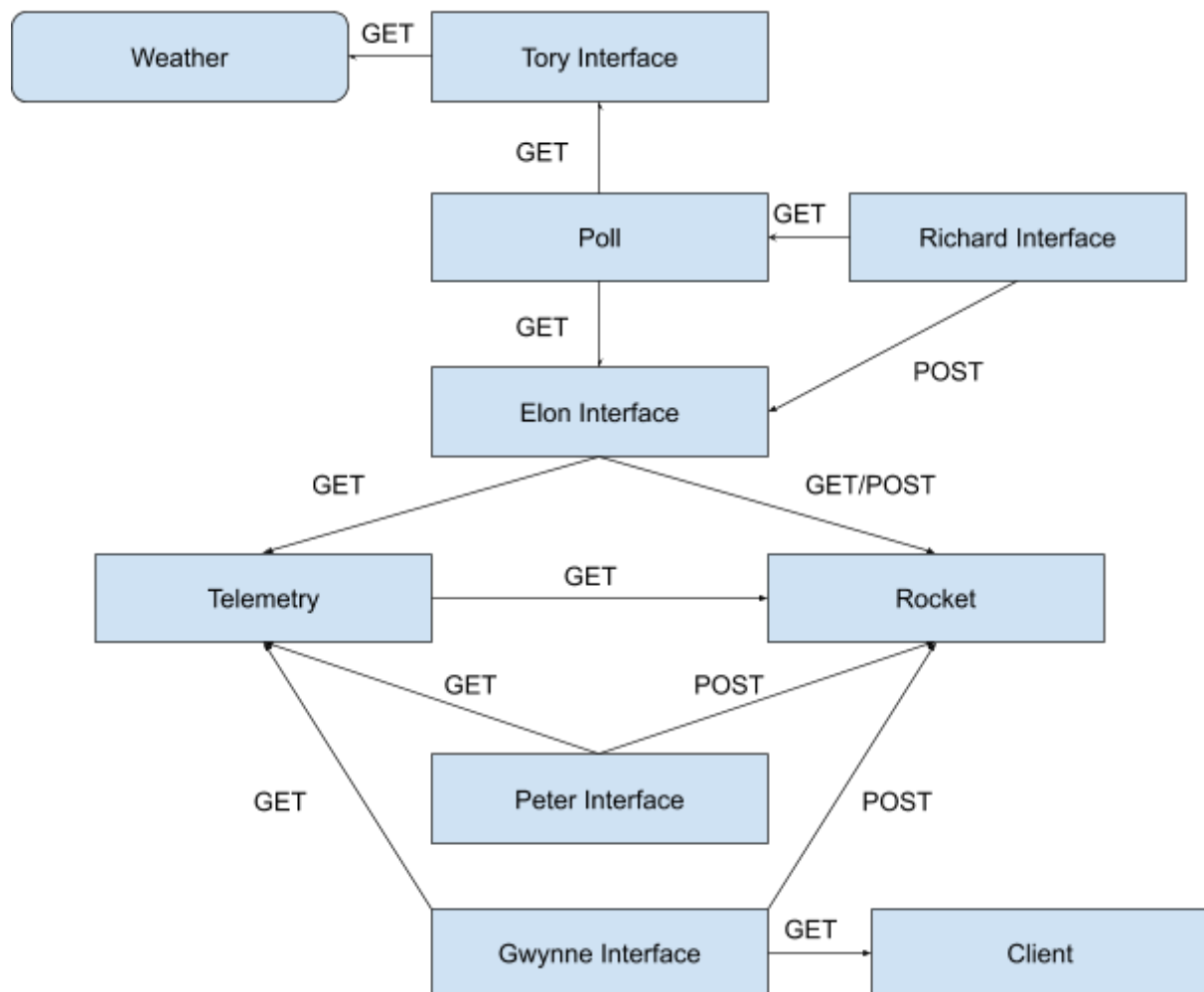


# Blue Origin X

Team E

First Delivery Report

## Architecture Actuelle



Pour arriver à cette architecture nous avons, lors de la première semaine créé une classe pour chaque User Story. Dans une problématique de responsabilités et d'objet métier, nous avons ensuite séparé le département de Richard en deux services : *Poll* qui s'occupe de faire le sondage et *RichardInterface* qui communique simplement avec *Poll* et le *ChiefRocketDepartment*. *RichardInterface* n'est alors qu'un petit ensemble de routes qui font appel à d'autres routes. Sur notre architecture actuelle nous savons que nous pouvons encore découper nos services pour mieux correspondre à une vision métier, en séparant par exemple le *ChiefRocketDepartment* en plusieurs services. Dans les défauts d'implémentations, nous pensons que Gwynne et Peter devraient passer par le *ChiefRocketDepartment* pour envoyer des ordres à la Rocket plutôt que de les envoyer eux-mêmes.

## Liste des services

Services	Ports	Routes	Paramètres possible du body	Retour de route	Dépendances
Rocket	4001	/status GET		"GO"	
		/order POST	"LAUNCH", "SPLIT", "TRAJCHANGE"		
		/data GET		rocketData	
Chief Rocket Department	4002	/status GET		"GO", "NO GO"	4001/status
		/order POST	"LAUNCH", "SPLIT"		4001/order, 4007/rocket Data 4001/order
Poll	4003	/poll GET		"GO", "NO GO"	4002/status 4004/status
Launch Weather Officer	4004	/status GET		"GO", "NO GO"	4005/status
Weather Supplier	4005	/status GET		"Clear", "Raining", "Windy", "Thunderstorm"	
Richard Interface	4006	/rocketLaunch GET		"NO GO"	4003/poll 4002/order
Telemetry	4007	/start GET			4001/data
		/rocketData GET			
Payload Department	4008	/sendPayloadInformation GET		"Payload in place"	4007/rocket Data 4001/order
Client	4009	/getPayloadInformation GET		"Traj" : "Orbital" "FutureSpeed": 100, "FutureAngle":90	

## Choix d'implémentation spécifiques

Nous avons décidé pour la mise à jour de la télémétrie qu'elle ferait des requêtes de façon régulière à la fusée afin de savoir son état. Ce choix a pour conséquence que si la fusée n'a plus de carburant il faut attendre que la télémétrie lui demande l'état de son carburant pour que d'autres personnes puissent être au courant, il faut donc que la fréquence des requêtes de la télémétrie soit assez élevée pour avoir une assez bonne précision, mais nous avons décidé de le développer comme ça car cela nous semblait plus logique et que si la rocket explose accidentellement la télémétrie le remarquera car elle ne recevra pas de réponse à ses requêtes au contraire de l'autre implémentation.

Nous avons décidé que l'ordre de destruction de la fusée devait passer par l'interface de Elon car même si Richard veut détruire la fusée, c'est à Elon de décider définitivement de lancer cet ordre à la fusée, cela a comme désavantage que la requête est un petit plus long mais nous pensons que c'est un risque à prendre.

Nous n'avons pas eu le temps de travailler sur l'atterrissage des propulseurs, pour cela il faudrait que nous découpons le service Rocket en deux service pour que l'un s'occupe de la firstStage et l'autre de la secondStage une fois la fusée divisée. Nous avons tout de même divisé les informations des deux parties de la fusée dans la base de donnée.