Battle Plan

# Algemene flow

Het doel is om een packager te hebben waarmee het mogelijk is om een executabel te maken. Deze executabel wordt via een deployment server verzonden naar alle verschillende environments. Het beste zou zijn, dat het een algemene deployment is. Hierdoor moet er maar op 1 knop gedrukt worden en zullen alle omgevingen de nieuwe versie ontvangen en kunnen installeren. Het laatste element van de schakeling is de environment. De environment moet de executabel kunnen uitvoeren, vervolgens moet er gecontroleerd worden of de nieuwe versie werkt. Als dit het geval is, dan mag de nieuwe omgeving de oude vervangen (deze houden we best nog achter de hand zodat we altijd iets hebben om naar terug te keren) -> een GUI waarin we de gewenste versie van het framework kunnen oproepen? Na de deployment laat de environment dan aan de server weten of de installatie al dan niet gelukt is. Die info is dan zichtbaar op de server.

In wat volgt, ga ik uitleggen wat het plan is voor het maken van mijn prototype. Achteraf zal dan gekeken worden of dit goed verlopen is en waar er eventueel aanpassingen moeten gebeuren.

# Packager

Uit het blokschema wordt al snel duidelijk wat er allemaal nodig is voor de packager. Het hoofddoel van de packager is uiteraard het bundelen van alle verschillende software componenten nodig voor het framework draaiende te krijgen. Naast het framework moeten ook de verschillende drivers gebundeld worden. Mocht het mogelijk zijn, dan moet er ook de mogelijkheid bestaan om terug te keren naar een vorige versie als bijvoorbeeld een hardware apparaat wordt aangesloten.

Tijdens het bouwen van het prototype gaan we proberen het volgende te realiseren:  
We gaan alle nodige software voor het framework in 1 package steken. Hiernaast gaan we ook een driver in een package steken. We gaan dit allemaal in 1 executabel steken, en we gaan controleren of deze werkt.   
We gaan dit allemaal maken met het qt installer framework. We gaan een online installer maken zodanig dat we kunnen kiezen welke packages we allemaal gaan installeren.

# Environment

Het doel van de environment is het draaien van de executabel. De omgeving moet voorzien worden van alle nodige drivers en zal ervoor zorgen dat het goed draait. Dit onderdeel zou ik willen realiseren met Docker. Er moet ergens eens overkoepelend geheel zijn waarin het mogelijk is om bij een nieuwe release een nieuwe omgeving op te zetten. Als de nieuwe release goed draait dan mag deze de oude vervangen. Als dit niet het geval is, dan moet de oude omgeving gebruikt worden.

Tijdens het bouwen van het prototype is het de bedoeling om het volgende te realiseren:  
Ik wil Docker draaiende hebben op een computer. Vervolgens ben ik van plan om vanuit de server een nieuwe versie van het framework te lanceren. Deze versie moet worden uitgepakt in de nieuwe omgeving en er moet gekeken worden of deze goed geïnstalleerd is. Vervolgens moet er dan een OK/NOK verstuurd worden en de juiste versie moet dan actief zijn.

# Server

De deployment server is de laatste component van het geheel. Het doel van de server is een overzicht bieden van alle verschillende hosts met alle versies van het framework, de mogelijkheid om een nieuwe packager te pushen en een overzicht met mogelijke fouten.

Het doel voor het maken van de prototype is het volgende:  
Ik wil een simpele server draaiende hebben met enkele versies van het framework op. Dit kan een eenvoudige HTTP server zijn draaiende mbv PHP. Verder wil ik ook dat hierop alle verschillende logs toekomen van de geïnstalleerde versies van het framework.  
Het is misschien wel mogelijk om gebruik te maken van een van de verschillende omgevingen, maar aangezien deze oplossingen niet gratis zijn, zal gekeken moeten worden of het dat wel waard is.