KU LEUVEN

FACULTEIT INDUSTRIELE
INGENIEURSWETENSCHAPPEN

TECHNOLOGIECAMPUS GENT

Titel masterproef

Ondertitel (facultatief)

Pieter-Jan ROBRECHT

Promotor(en): Annemie Vorstermans

Co-promotor(en): Wim Vancroonenburg
Carl Eeckhout

Masterproef ingediend tot het behalen van de graad van master of Science in de industriële wetenschappen: master of Science in de industriële wetenschappen ICT Advanced Communicatie Technologies

Academiejaar 2016 - 2017

Dankwoord

Dank aan mezelf
Dank aan een ander

Abstract

Televic Rail maakt gebruik van een Python framework voor het uittesten van de hardware die zij produceren. Het installeren van dit framework op een nieuwe computer of testtoren is een uitgebreide klus. Er moeten verschillende hardware drivers geïnstalleerd worden en er zijn meerdere bibliotheken nodig om het framework correct te laten functioneren. Een meer efficiënte werkwijze voor het updaten en installeren van het framework is wenselijk. Opzet van deze scriptie is dit te onderzoeken en vervolgens uit te werken. Het aantal drivers en bibliotheken zal enkel toenemen. Bovendien moet de applicatie ook schaalbaar zijn naar de toekomst toe. Het probleem werd uitgesplitst in drie logische componenten: de packager, de deployment server en de deployment environment. Deze structuur wordt doorheen de thesis aangehouden en vormt de basis voor het ontwerp van de installer/updater. In een eerste fase wordt een prototype ontworpen. Dit prototype bevat elke component en is tegelijk op een zo efficiënt mogelijke manier gebouwd. Er werden test uitgevoerd op dit prototype. Met behulp van de uitslagen is het ontwerp aangepast. Fase twee van de thesis bouwt verder op deze resultaten. Het ontwerp is waar nodig aangepast. Uiteindelijk is een demo applicatie ontwikkeld.

Trefwoorden: installer, updater, Python, Docker

Inhoudsopgave

Lijst van figuren

Hoofdstuk 1

Situering

Televic Rail heeft een Python test framework ontworpen waarmee zij in staat zijn om verschillende hardware componenten te controleren op fouten. Dit framework wordt dan ook zeer intensief gebruikt tijdens het productieproces. Het framework werd initieel ontworpen om enkel te werken op testtorens, maar werd later aangepast zodanig dat het onafhankelijk van de testtoren gebruikt kan worden.

1.1 Het probleem

Aangezien het Python framework gebruikt van verschillende niet-Python bibliotheken en verschillende drivers voor de hardware, is het installeren van dit framework op een nieuw systeem een ganse klus. Updates uitvoeren geeft ook verschillende problemen. Het doel van deze thesis is dan ook het vinden van een langdurige oplossing van dit probleem. Na een kleine analyse van het probleem, werd het al snel duidelijk dat er op verschillende onderliggende problemen een oplossing moet gevonden worden.

Het framework bestaat uit verschillende componenten en drivers. Al deze componenten en drivers hebben hun eigen installatie wijze en moeten in een bepaalde volgorde geïnstalleerd worden. Hiernaast hebben sommige doelsystemen een specifiek configuratiebestand nodig om goed te kunnen functioneren. Tijdens deze thesis zal er dan ook onderzocht worden hoe alle verschillende componenten kunnen worden gepackaged tot één groot geheel dan kan worden verzonden naar de verscheidene doelsystemen.

De installatie van het framework gebeurt best in een speciale deployment omgeving. Mocht er een fout optreden tijdens de installatie, of tijdens een update, van het framework, dan moet een vorige, nog werkende, versie van het framework herstelt worden. Hierdoor voorkomen we verschillende problemen, zoals het stilleggen van de productie. Er moet dus onderzocht worden of het mogelijk is om een omgeving te creëren waarin een update/installatie kan plaatsen vinden en, mocht dit nodig zijn, de aanpassingen kunnen ongedaan worden gemaakt.

Naast het updaten/installeren zijn er nog problemen. Het framework wordt gebruikt op verschillende sites en niet ieder site zal dezelfde versie van het framework gebruiken. Bij een rollout van een nieuwe versie is het mogelijk dat de update slaagt in de ene site maar niet op de andere. Het bijhouden van de verschillende, al dan niet geslaagde, deployments moeten centraal zichtbaar zijn. Om die redenen zal er onderzocht moeten worden om een overkoepelende manager te voorzien. Met deze manager kan er een overzicht gecreëerd worden waarmee zichtbaar wordt welke deploy-

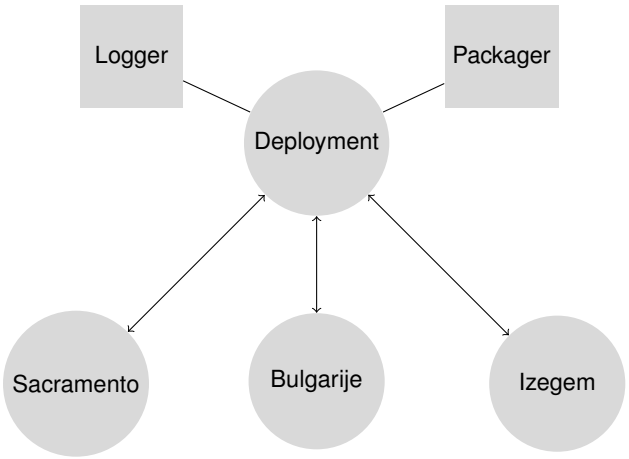
ments geslaagd zijn, hoeveel updates gelukt zijn, de versie van het framework dat gebruikt wordt op één bepaalde site, ... Dit onderdeel moet zeer schaalbaar zijn zodanig dat naar de toekomst toe, het zeer eenvoudig is om nieuwe systemen toe te voegen.

1.2 Werkwijze

In Sectie 1.1 heb ik besproken welke verschillende problemen opgelost moeten worden. In wat volgt ga ik bespreken hoe ik dit ga aanpakken en op welke wijze ik te werk ga gaan.

Tijdens de bespreking van de problemen werd het al snel duidelijk dat het werk op te delen valt in drie grote componenten. De oplossing die opgebouwd wordt tijdens deze thesis zal dan ook uit deze drie componenten bestaan en alle verschillende stappen worden uit gelegd aan de hand van deze drie onderdelen. Het eerste onderdeel is de packager. Deze component zal instaan voor het inpakken van de verschillende drivers en zal ook de software bevatten die het installatieproces in goede banen zal leiden. Naast de packager hebben we dan de deployment server. De server zal functioneren als manager in het grote geheel met als doel het weergeven van alle verschillende deployments en verschillende gegevens, zoals bijvoorbeeld de versie van het framework. Het laatste onderdeel dat voorzien zou moeten worden is dan de deployment omgeving. De omgeving draait op de verschillende systemen en zal dienen als een sandbox voor het installeren/updaten van het framework.

Verder gaat deze thesis nog onderverdeelt worden in twee grote delen. In het eerste deel gaat er een prototype worden gemaakt met de bovengenoemde elementen. Als het prototype gemaakt is, zal het systeem getest worden en zal er gecontroleerd worden of alle nodige componenten goed geïmplementeerd werden. Na deze stap zal het ontwerp aangepast worden en zal de einde demo gemaakt worden. De laatste fase van deze thesis bestaat dan uit het testen van deze demo waarna het dan mogelijk is om een conclusie te trekken uit het geleverde werk. Op Figuur 1.1 zien we de algemene structuur die de applicatie zal krijgen.

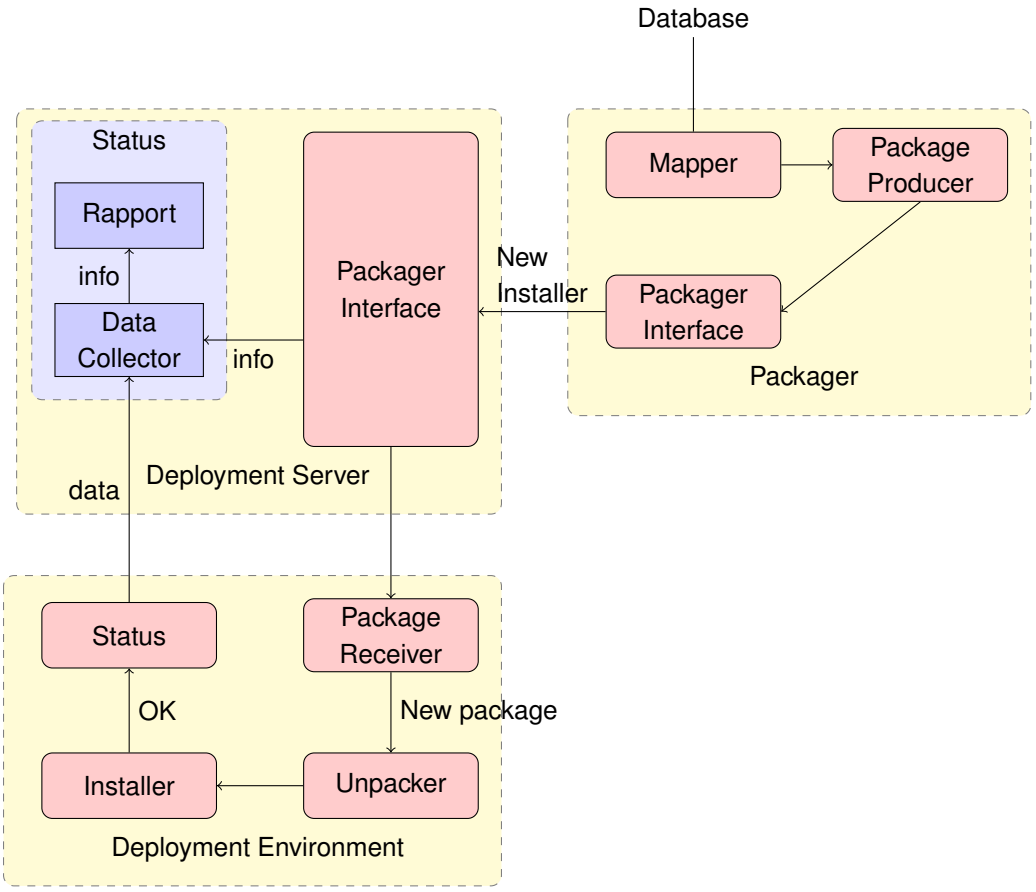


Figuur 1.1: Overzichtsdiagram met algemene structuur

Hoofdstuk 2

Bespreking

In Hoofdstuk 1 op pagina 1 werd er uitgelegd welke problemen er allemaal zich voordoen tijdens het zoeken naar oplossing voor het installeren van het Python framework. In wat volgt ga ik bespreken hoe het prototype zal gevormd worden en vervolgens hoe de demo vorm zal krijgen. Dit ga ik doen aan de hand van het blokschema dat terug te vinden is op Figuur 2.1.



Figuur 2.1: Blok diagram met alle componenten

2.1 Packager

Het eerste onderdeel van de applicatie is de packager. Zoals in Sectie 1.1 is uitgelegd, moet de packager instaan voor het samenvoegen van alle drivers en bibliotheken.

Hoofdstuk 3

Prototype

3.1 Uitwerking

3.2 Testen

Hoofdstuk 4

Eind product

4.1 Uitwerking

4.2 Testen

Hoofdstuk 5

Conclusie

Bibliografie

Castleman, K. R., Schulze, M. A., and Wu, Q. (1998). Simplified design of steerable pyramid filters. In *Proc. IEEE ISCAS*.

Granlund, G. and Knutsson, H. (1995). *Signal Processing for Computer Vision*. Kluwer Academic Publishers.

Holmes, T., Bhattacharyya, S., Cooper, J., Hanzel, D., and Krishnamurti, V. (1995). *Handbook of biological Confocal Microscopy (2nd ed.)*, chapter Light Microscopic Images reconstructed by Maximum Likelihood deconvolution, pages 389–402. Plenum Press, New York.

Hossack, W. (WWW). Digital image analysis and image processing 1. `http://www.ph.ed.ac.uk/~wjh/teaching/dia/index.html`.

Plášek, J. and Reischig, J. (1998). Transmitted-light microscopy for biology: A physicist's point of view part i. *Proceedings of the Royal Microscopical Society*, 33(2):121–127.

van der Voort, H. (1989). *Three dimensional Image Formation and Processing in Confocal Microscopy*. PhD thesis, University of Amsterdam.

Bijlage A

Een aanhangsel

sdfsffqsfsf

Bijlage B

Beschrijving van deze masterproef in de vorm van een wetenschappelijk artikel

Bijlage C

Poster

FACULTEIT INDUSTRIELE INGENIEURSWETENSCHAPPEN
TECHNOLOGIECAMPUS GENT
Gebroeders De Smetstraat 1
9000 GENT, België
tel. + 32 92 65 86 10
fax + 32 92 25 62 69
iiw.gent@kuleuven.be
www.iiw.kuleuven.be