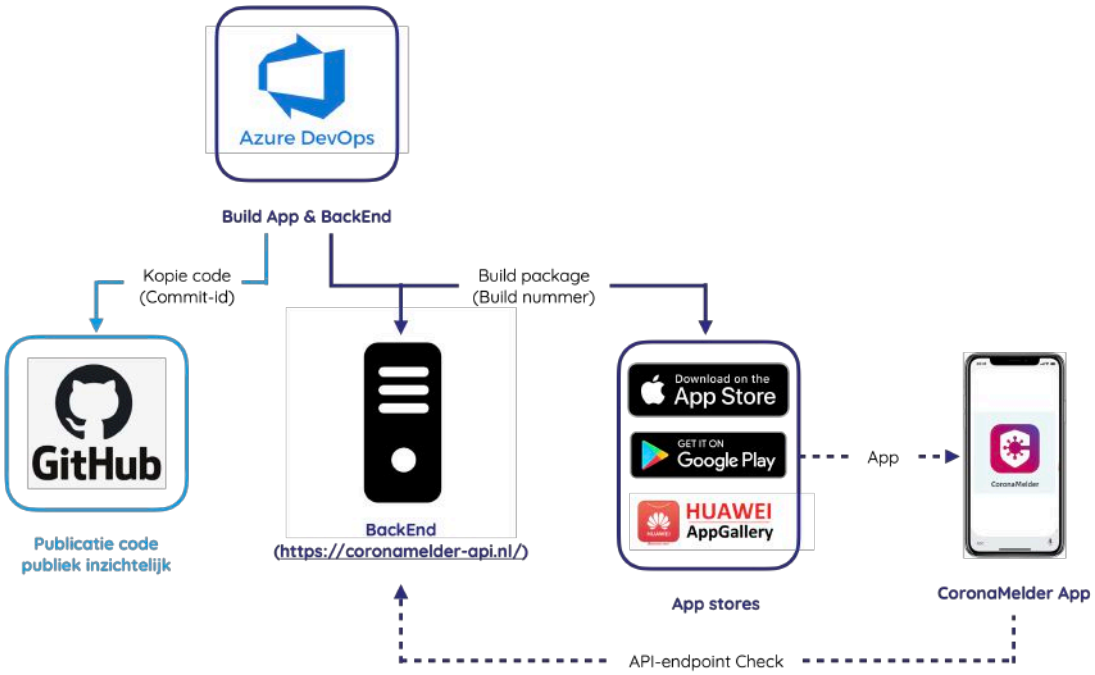
Beschrijving Build Verificatie CoronaMelder App



Doel van de Build Verificatie

De Build Verificatie controleert of de CoronaMelder App zoals deze via app stores wordt aangeboden aan het publiek, alsmede de backend waarmee de apps communiceren, ook daadwerkelijk zijn gebouwd met eenzelfde code als gepubliceerd in de publiek toegankelijke GitHub-omgeving (https://github.com/minvws).

Naast het vergelijk of de toegepaste code voor de bouw van de apps en backend overeenkomt met de in GitHub gepubliceerde code, wordt tevens geverifieerd of de apps ook communiceren met de juiste backend (https://coronamelder-api.nl/). Daarnaast wordt nog gecheckt of een gemandateerde medewerker van de Rijksoverheid de app heeft vrijgegeven aan de store c.q. de backend vrijgegeven voor productie.

Dit document is samengesteld om het publiek op hoofdlijnen inzage te geven op welke wijze de Build Verificatie wordt uitgevoerd. De lezer hoeft dan ook niet te beschikken over diepgaande ontwikkel- c.q. programmeerkennis.

Op het moment van totstandkoming van dit document wordt de CoronaMelder App gedistribueerd via de app stores van Apple, Google en Huawei.

Bouw en release van de apps en backend en toevoegen code aan GitHub

De code waarmee de apps en de backend worden gebouwd wordt ontwikkeld en onderhouden in Microsoft Azure DevOps (DevOps). Op het moment dat de code gereed is om toe te passen voor het bouwen van de app of backend, vindt de zogenaamde '**build**' plaats op basis van deze code. Binnen DevOps wordt er een commando gegeven waarmee een geautomatiseerd proces wordt afgetrapt die op basis van de code en app bouwt. Het resultaat hiervan noemen we de '**build package**'.

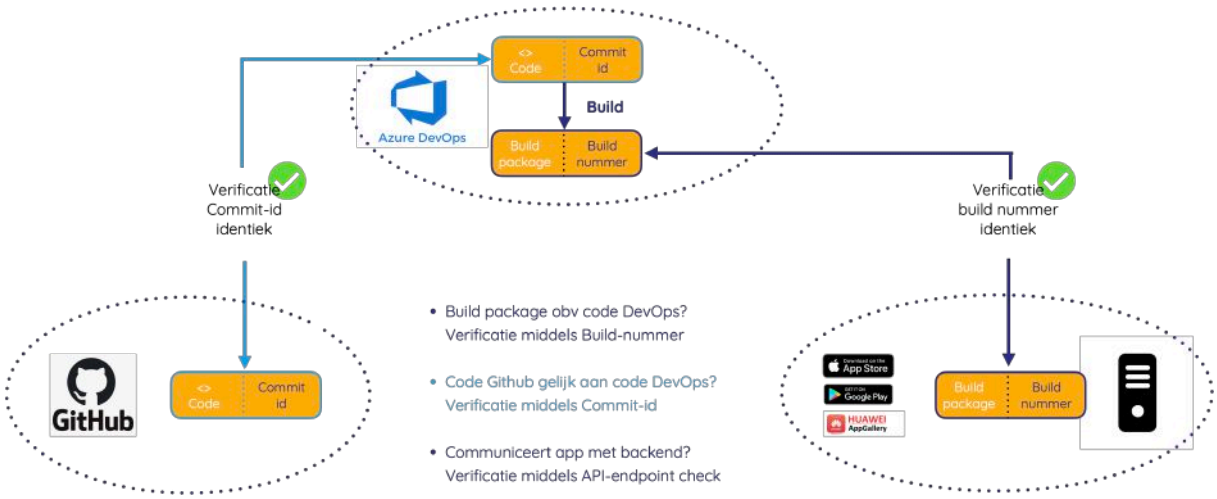
Vervolgens wordt de verkregen build package ge-upload naar de betreffende app store of de backend productie-omgeving. In de backend productie-omgeving staat de backend direct live. Betreft het de app, dan valideert de app store of de app aan de door de app store gestelde eisen voldoet om vrij te geven voor publicatie. Indien de app hieraan voldoet, dan mag de app worden vrijgegeven. Na vrijgave is de app publiekelijk beschikbaar en kan deze worden gedownload door gebruikers of indien reeds eerder gedownload, wordt de app ge-update met de vrijgegeven nieuwe versie.

De vrijgave van de app geschiedt in de stores van Apple en Google door middel van het geven van toestemming door een gemandateerde medewerker van VWS. In de app store van Huawei is dat niet mogelijk en indien de app de validatie doorstaat, wordt deze automatisch vrijgegeven. Echter het proces van build tot en met upload store en vrijgave app in de store van Huawei is volledig geautomatiseerd en het aftrappen van dit proces geschiedt door een gemandateerde medewerker van VWS.

Parallel of opvolgend aan bovenstaand beschreven proces wordt een kopie van de code toegepast in het build proces gepubliceerd op GitHub. Voor de iOS app (Apple store) is dit in de publieke GitHub repository: 'minvws/nl-covid19-notification-app-ios', voor de Android apps (Android en Huawei stores) is dit de publieke GitHub repository: 'minvws/nl-covid19-notification-app-android' en voor de backend is dit de publieke Github repository: 'minvws/nl-covid19-notification-app-backend' .

Tot slot wordt er tijdens de build binnen DevOps Git-metadata opgeslagen van betreffende build. Deze metadata bevat onder meer het door DevOps aan betreffende build toegewezen '**build-nummer**' en '**commit-id**' alsmede de timestamp van de build. Het build-nummer en de commit-id zijn uniek binnen het VWS applicatie domein. Het build-nummer wordt meegegeven in de build package naar de app store en de commit-id met de kopie code naar GitHub.

Proces Build Verificatie apps en backend



De uitvoer van de Build Verificatie voor de apps en de backend start na de upload van een build package in de app store voor de apps en in de backend productie-omgeving voor de backend. Als eerste stap verzamelt Escrow Alliance minimaal de volgende informatie uit de verschillende omgevingen (DevOps, GitHub en voor de apps uit betreffende app store en voor de backend van de API-endpoint statuspagina):

1. Metadata uit de app store (voor apps) en van de API-endpoint statuspagina (voor de backend) behorende bij de nieuwe build;
2. Git-metadata uit DevOps behorende bij de nieuwe build;
3. Build-logs uit DevOps behorende bij het uit stap 1 verkregen build-nummer en de gerelateerde pipeline-id (release stage);
4. Code uit DevOps behorende bij het verkregen build-nummer uit stap 1;
5. Code uit GitHub behorende bij het verkregen commit-id uit stap 2.

Op basis van de verkregen informatie voert Escrow Alliance de volgende controles uit:

1. Verificatie of het build-nummer uit app store (apps) of API-endpoint statuspagina (backend) identiek is aan het build-nummer in DevOps. Deze dienen identiek te zijn waarmee wordt aangetoond dat het build package is gemaakt op basis van de code behorende bij het build-nummer in DevOps. *

2. Verificatie of het commit-id uit GitHub identiek is aan het commit-id in DevOps. Ook deze dienen identiek te zijn waarmee aangetoond wordt dat de code gepubliceerd en publiek gemaakt in GitHub, identiek is aan de code waarmee in DevOps de build heeft plaatsgevonden.

3. Verificatie of de code verkregen uit de DevOps en de GitHub omgevingen inhoudelijk identiek aan elkaar zijn. De code dient met uitzondering van aanmaak- en/of mutatiedatum van de bestanden gelijk te zijn en toont daarmee aan dat dezelfde code voor de bouw is toegepast als publiek is gemaakt via GitHub.

4. Verificatie of de verkregen build-logs het juiste build-nummer en commit-id bevatten en een controle of de tijdlijnen opgenomen in de build-logs bij benadering overeenkomen met de te verwachte chronologische volgorde van de release pipeline activiteiten.

5. Verificatie of medewerker geautoriseerd was door VWS om de app of backend voor publicatie vrij te geven in betreffende app store respectievelijk backend productie-omgeving.

** Voor de Android apps (app store van Google en Huawei) wordt het build package in de DevOps omgeving en in de store van Google c.q. Huawei nog met elkaar vergeleken op basis van een gegenereerde hash op het build package. Deze dienen identiek te zijn. Voor iOS is dit niet mogelijk, omdat de build package niet uit de store van Apple kan worden gedownload en dat Apple een resource bundle aan het build package toevoegt, waardoor een hash-vergelijk niet meer mogelijk is.*

De Build Verificatie is succesvol afgerond indien bovengenoemde verificaties allen met een positief resultaat worden afgerond en er geen afwijkingen worden geconstateerd. De resultaten worden vastgelegd in het verificatie rapport en de bewijsstukken van de uitgevoerde verificatie opgenomen in een archiefbestand.

Het verificatierapport bevattende het resultaat en het archiefbestand bevattende de bewijslast worden gedeeld met de notaris ten behoeve van de afgifte van de verklaring.