

# Extended Abstract

---

MMORPG ARCHITECTUUR VOOR STARFALL ONLINE

Finn Alberts, Aaron Dang, Laurent Dassen en  
Peter Derks

Begeleider: Polo Swelsen | Opdrachtgever: Miguel van de Laar  
ZUYD HOGESCHOOL | HBO ICT



## Abstract

Massive Multiplayer Online Role-Playing Games (MMORPG) zijn spellen met grote technische uitdagingen, zoals performance, schaalbaarheid en elasticiteit. Binnen dit project is een software architectuur ontworpen en vastgelegd middels Kruchten 4 + 1 voor een MMORPG. Hierbij is in de architectuur vooral gewerkt aan een schaalbare, elastische en high-performance architectuur, omdat dit de factoren zijn die in een risicoanalyse naar voren zijn gekomen. Middels een proof of concept is aangetoond dat de ontwikkelde architectuur, waarbij gebruik wordt gemaakt van event-driven communicatie tussen client en server, werkt. Daarnaast is te zien dat de nodige informatie voor het schaalbaar maken van de applicatie beschikbaar is, al is de schaalbaarheid niet in het proof of concept verwerkt.

# Introductie

## Aanleiding

De afgelopen jaren zijn Massive Multiplayer Online games (MMO's) steeds populairder geworden. De verwachting is dat tussen 2021 en 2026 de markt met nog eens 21,25 miljard USD zal toenemen (Technavio, 2022). Een grote uitdaging van deze MMO's is het "massive" aspect. Niet alleen is het een uitdaging om de grote aantallen spelers aan te kunnen, wat er miljoenen kunnen zijn, maar ook het schaalbaar maken van de capaciteit is hierbij een uitdaging. Hierbij speelt goede performance een grote rol. Een spel met veel lag (ofwel een hoge latency) verliest namelijk snel aan populariteit. Daarnaast worden voorwerpen in MMO's regelmatig echt geld waard (Esports Extra, sd). Het tegengaan van hackers en valsspelers wordt daarmee extra belangrijk. Om al deze uitdagingen te overkomen is een goede architectuur nodig, die met al deze aspecten rekening houdt.

## Doelstelling

Het doel van dit project is het komen tot een software architectuur voor een MMORPG (Massive Multiplayer Online Role-Playing Game), waarbij rekening wordt gehouden met de specifieke uitdagingen voor dit soort spellen.

## Beschrijving van de MMORPG

De MMORPG waarvoor de architectuur wordt ontworpen heet Starfall Online. Deze MMORPG is een top-down 2D pixelart open world MMORPG die plaats vindt in een middeleeuwse fantasiewereld Astaria. De kern van het spel is dat de wereld wordt bedreigd door vallende sterren die uit de lucht vallen. Wanneer een vallende ster landt, wordt een locatie 'corrupt' en zijn de monsters in deze locatie extra sterk. De taak van de speler is om Astaria te beschermen tegen deze vallende sterren. Een volledig overzicht van wat mogelijk is in het spel is te zien in bijlage A.

## Theoretisch kader

### MMORPG

MMORPG staat voor Massive Multiplayer Online Role-playing Game. Een MMORPG is een genre van rollenspel waarbij via het internet een groot aantal spelers met elkaar kunnen interacteren. De meeste MMORPGs spelen zich af in een fantasiewereld waar de spelers controle hebben over een eigen karakter waarmee met de wereld kan worden geïnteracteed. Deze karakters hebben vaak een rijk achtergrondverhaal waardoor de speler een specifieke rol in beslag neemt. (Wikipedia-community, 2022)

### Software architecture

Software architectuur verwijst naar de fundamentele structuren van een software systeem. De architectuur van een software systeem is hetzelfde als de architectuur van een gebouw. Beide laten zien hoe de onderliggende structuur uit ziet, uit welke elementen deze bestaat en hoe deze elementen samenwerken. (Wikipedia-community, 2022)

### Kruchten 4 + 1

Kruchten 4+1 is een aanpak om architecturen te beschrijven van softwareintensieve systemen doormiddel van verschillende views. Elke view beschrijft het oogpunt van een verschillende stakeholder zoals eindgebruikers, ontwikkelaars, systeem ingenieurs, projectmanagers etc. Kruchten 4+1 bestaat totaal uit vijf delen namelijk de logical view, de process view, de development view, de physical view en de scenarios. (Kruchten, 1995)

- Logical view: de logical view beschrijft de functionaliteiten die het systeem moet aanbieden aan de eindgebruikers
- Process view: de process view beschrijft de niet-functionele eisen van een systeem.
- Development view: de development view beschrijft de softwaremodule-organisatie van een systeem.
- Physical view: de physical view beschrijft de fysieke hardware van het systeem.
- Scenarios: de scenarios beschrijft de hele architectuur doormiddel van use cases. De scenarios worden gebruikt om architectuur elementen te identificeren en architectuur keuzes te valideren. (Wikipedia-community, 2021)

## Methode

Binnen dit project is met een iteratieve werkwijze gewerkt, welke is afgeleid van agile (Scrum Company, sd) werken. Hierbij is wekelijks met de opdrachtgever samengekomen om de voortgang te bespreken en feedback te verkrijgen. Op deze manier kon tijdig worden bijgestuurd.

Om structuur te behouden is gebruikt gemaakt van een kanban bord (Rehkopf, sd), bestaande uit vier kolommen: backlog, to do, doing en done. De backlog bevatte alle nog uit te voeren taken. De to do-kolom gaf aan welke taken voor die week op de planning stonden. De doing kolom gaf de taken aan waaraan op dat moment gewerkt werd en de done kolom gaf de reeds afgeronde taken weer.

Voor dit project is een system requirements specificatie (SRS) opgesteld met daarin de eisen voor de architectuur. Deze zijn opgesteld en gevalideerd op basis van gesprekken met de opdrachtgever. Tevens zijn deze eisen met de MoSCoW-methode (Wikipedia-community, 2022) geprioriteerd. Binnen dit SRS lag de nadruk op de niet-functionele eisen, daar deze van groter belang zijn voor het ontwerpen van een software architectuur.

Op basis van de opgestelde eisen is een overzicht gemaakt van de nodige componenten voor de architectuur. Voor deze componenten is vervolgens een risicoanalyse uitgevoerd om te zien waar binnen de architectuur de grootste uitdagingen zitten. Tevens zijn direct maatregelen uitgewerkt om deze risico's te minimaliseren.

Rekening houdende met de gevonden risico's is een software architectuur ontworpen. Voor het vastleggen van deze architectuur is gebruik gemaakt van Kruchten 4 + 1 (Jayawardene, 2021). Binnen Kruchten 4 + 1 staan scenario's centraal. Hieromheen bestaan een viertal views: de development view, de process view, de logical view en de physical view. Voor het weergeven van deze scenario's en views is gebruik gemaakt van UML-diagrammen (Object Modeling Group, sd). Specifiek zijn dit een use-casediagram voor de scenario's, een conceptueel klasse-diagram voor de logical view, een componentendiagram voor de development view, een sequentiediagram voor de process view en een deploymentdiagram voor de physical view.

Om aan te kunnen tonen of de ontwikkelde architectuur ook in de praktijk werkt, is een proof of concept ontwikkeld in Unity (Unity, sd) samen met Mirror Networking<sup>1</sup>. Dit proof of concept toont niet de werking van de gehele architectuur, maar wel van de grootste risico's/uitdagingen van de architectuur, die tijdens de risicoanalyse naar voren zijn gekomen of waarbij de grootste onzekerheid is. Voor dit proof of concept is een testrapport opgesteld, waarbinnen de verwachte resultaten en het daadwerkelijke resultaat worden beschreven.

---

<sup>1</sup> Mirror Networking is een open source netwerkoplossing voor Unity  
Extended Abstract  
MMORPG Architectuur

## Resultaten

### Beschrijving artefact

Het artefact van dit project bestaat uit twee onderdelen: een architectuurbeschrijving en een proof of concept. De architectuurbeschrijving laat zien hoe de architectuur in elkaar zit met behulp van Kruchten 4+1 (Jayawardene, 2021). Deze architectuur is gemaakt aan de hand van bij de opdrachtgever opgehaalde en gevalideerde eisen. Om te laten zien dat deze architectuur werkt, is een proof of concept ontwikkeld, waarin de communicatie tussen client en server wordt getoond. Daarnaast toont het proof of concept aan dat alle informatie, die nodig is voor het kunnen op- en afschalen van de capaciteit, centraal beschikbaar is. Hiermee toont het proof of concept dus ook, dat schaalbaarheid mogelijk is.

### Eisen

Aan het begin van dit project zijn eisen opgehaald. Deze eisen maken duidelijk waaraan de architectuur hoort te voldoen en welke risico's zich eventueel kunnen voordoen. In Tabel 1 zijn de meest belangrijke eisen weergegeven. Een volledig overzicht van eisen is te vinden in bijlage A en bijlage B.

*Tabel 1 Belangrijkste eisen*

Nummer	Eis	Prioriteit
<b>N3</b>	De capaciteit van de MMORPG kan worden op- en afgeschaald afhankelijk van de behoeften.	Must
<b>N10</b>	De spelwereld is consistent voor alle spelers.	Must
<b>N18</b>	Gegevens van spelers zijn consistent.	Must

### Overzicht van componenten

De componenten die nodig zijn voor het realiseren van de architectuur zijn als volgt:

- Account management: functionaliteit voor inloggen en registreren
- Character management: alle informatie voor karakters, zoals statistieken, inventaris, enzovoorts.
- Game world: de spelwereld
- Game client: de lokaal draaiende versie van het spel bij de gebruiker
- Load and distribute management: verbinden van de Game client met de Game world en het opschalen van de Game world
- Game store: de winkel voor het kopen van cosmetics met echte valuta (zoals euro of dollar)
- Market: marktplaats waarop spelers onderling spullen kunnen verhandelen
- Online payment: component voor het verwerken van betalingen met echte valuta (zoals euro of dollar)
- Social systems: component voor alle sociale interacties, zoals chat, party's en guilds
- Coordinator: server-side component voor het centraal doorsluizen van data (zoals karakterinformatie)

### Risicoanalyse

In bijlage C is te zien dat er een aantal grotere risico's zijn (rood gemarkeerd). Deze risico's laten zien dat deze componenten een groot risico vormen bij bepaalde ilities en dat de kans hierop hoog is. Het component Game world heeft risico's bij performance, scalability en elasticity. Dit is dan ook het

component met de meeste risico's. Daarnaast heeft de Game client ook een risico bij performance en security. Tot slot vormt elasticity een risico bij Social systems en Character management.

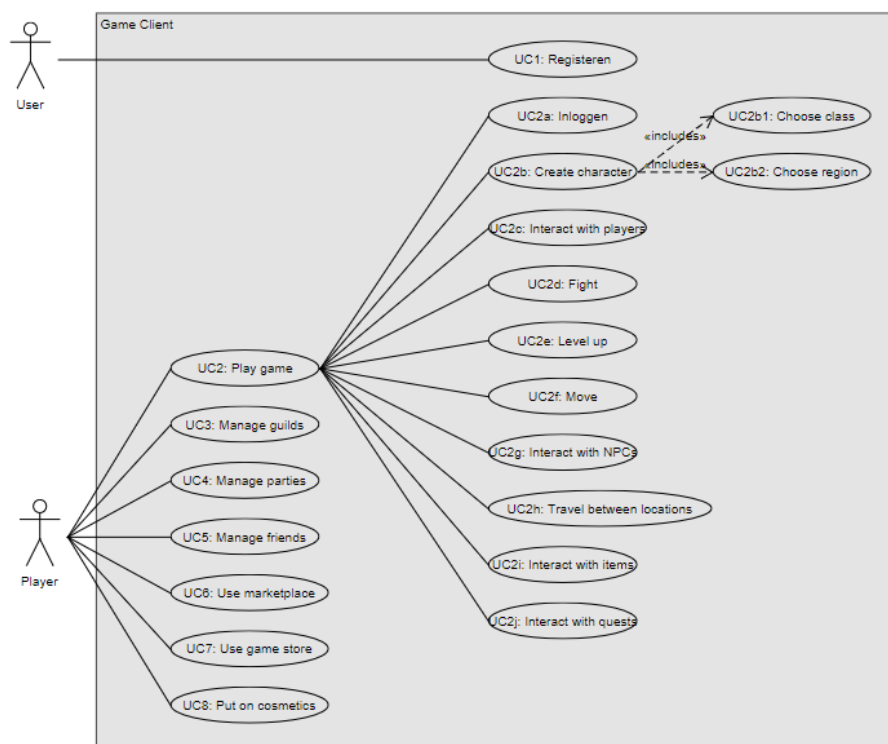
Om de risico's binnen de Game world te minimaliseren worden acties in beginsel binnen de Game client uitgevoerd en alleen gevalideerd in de Game world. Daarnaast wordt de communicatie tussen de Game client en Game world op een event-driven manier uitgevoerd. Het uitgangspunt hierbij is tevens dat in-game locaties altijd op dezelfde server draaien, om communicatie tussen servers zoveel mogelijk te voorkomen. Om de elasticity van Game world, Character management en Social systems te waarborgen, worden deze componenten schaalbaar gebouwd. Dat wil zeggen dat er meerdere instanties van deze componenten kunnen draaien.

Om de performance van de Game client te waarborgen is voor een pixelart artstyle gekozen, die grafisch minder intensief is. Daarnaast vormt de security van de Game client een risico, omdat dit component lokaal bij de gebruiker draait en dus gemanipuleerd zou kunnen worden. Er wordt daarom nooit kwetsbaar informatie door de Game client heen gevoerd en informatie vanuit de Game client wordt in beginsel niet als 'de waarheid' gezien.

### Kruchten 4 + 1

Uit de kruchten 4+1 methode zijn een zestal diagrammen opgesteld: een use-case diagram, een componentendiagram, een deploymentdiagram, een conceptueel klassendiagram en twee sequentiediagrammen.

### Use-case diagram

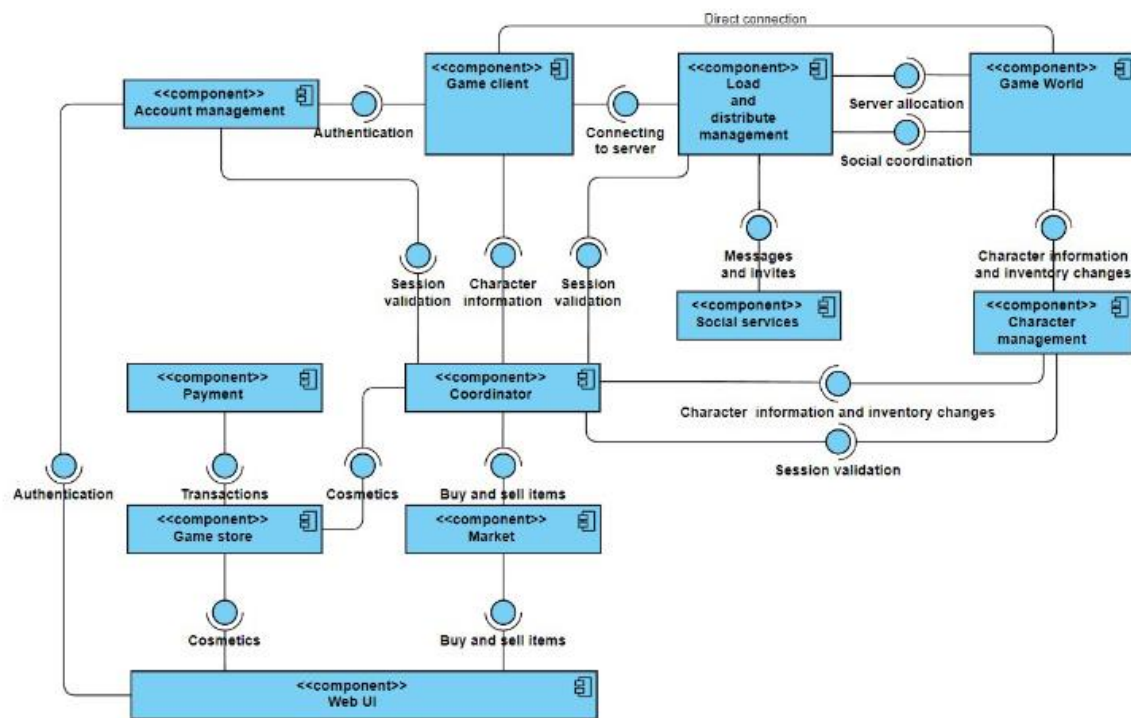


Figuur 1 Use-case diagram

Het use-case diagram in Figuur 1 laat de verschillende use cases zien die een gebruiker en speler uit kunnen voeren. Dit diagram laat ook de verschillende scenario's zien die zich kunnen voordoen in de MMORPG. In het diagram worden alleen de acties weergegeven die een gebruiker zou moeten

kunnen uitvoeren. Echter is er bij online spellen altijd sprake van valsspelers. Met deze valsspelers moet ook rekening gehouden worden.

## Componentendiagram

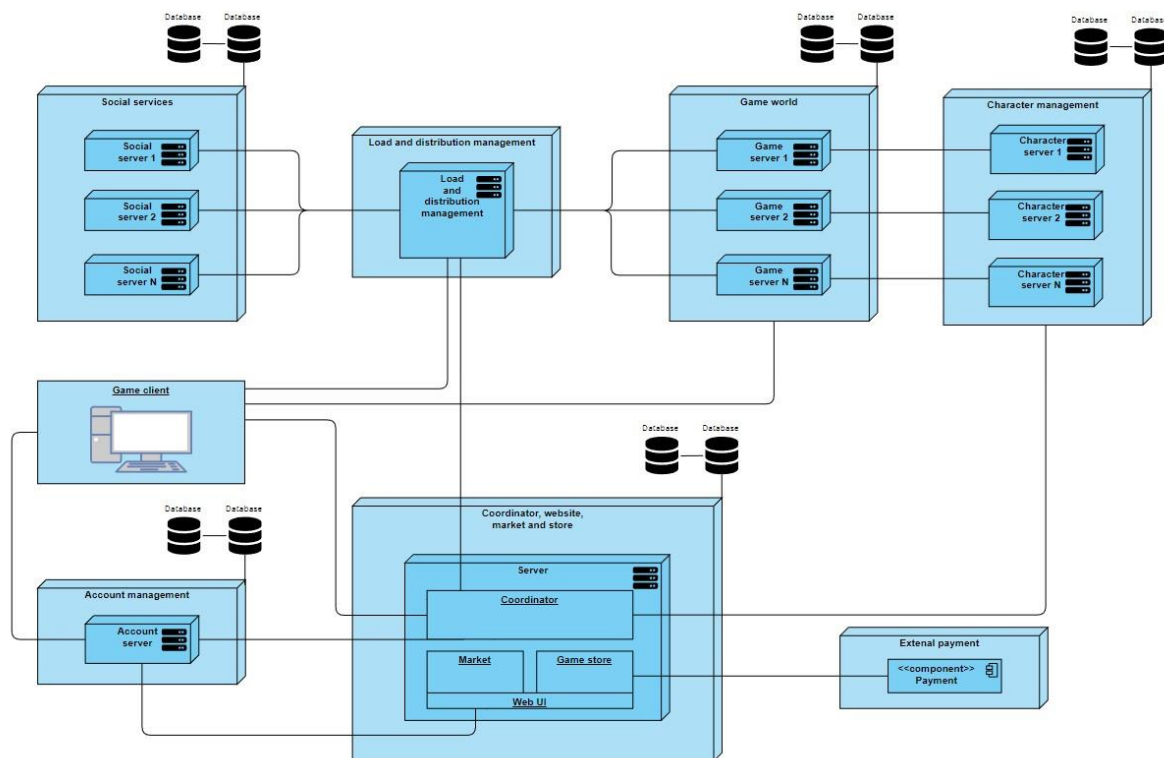


Figuur 2 Componentendiagram

In Figuur 2 is een componentendiagram te zien. Hier zijn de componenten samengevoegd tot een architectuur. Hierdoor worden de connecties en samenwerkingen tussen de componenten duidelijk. In de diagram is te zien hoe het Load and distribute management component de direct connection tussen de Game client en de Game world opzet. Ook is te zien dat Account management een session validation gebruikt om verscheidene componenten indirect van authenticatie te voorzien.



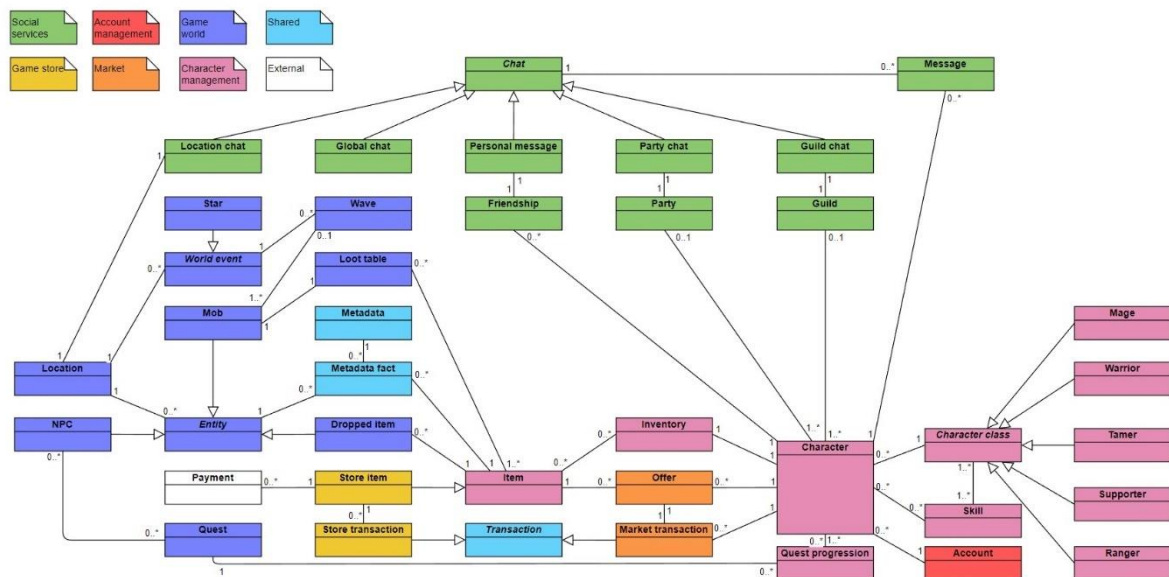
## Deploymentdiagram



Figuur 3 Deploymentdiagram

Het deploymentdiagram in Figuur 3 geeft weer hoe de fysieke systemen van de architectuur eruit zullen zien. In dit diagram is te zien dat Social services, Game world en Character management bestaan uit een variabel aantal servers. Hierdoor kunnen deze systemen snel dynamisch op- en afgeschaald worden naar de behoeften van dat specifieke moment. De Coordinator, Web UI, Market en Game store worden op dezelfde server gedraaid. De game store en market zijn toegankelijk middels een web-UI. Account management wordt op een aparte server gezet, omdat dit systeem gevoelige persoonsgegevens bevat. Tot slot wordt elke database redundant opgeslagen om de kans op data verlies en downtime zo veel mogelijk te beperken.

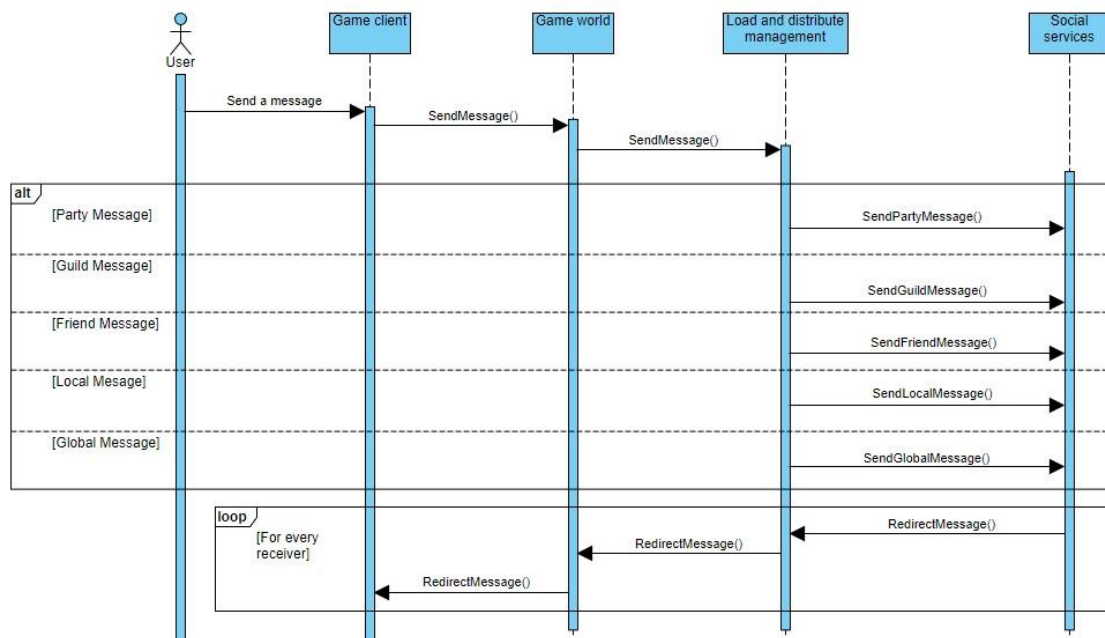
## Klassendiagram



Figuur 4 Klassendiagram

In Figuur 4 zijn de verschillende componenten van het componentendiagram uitgewerkt tot klassen. Ieder component en zijn bijbehorende klassen is voorzien van dezelfde kleur. Dit maakt duidelijk hoe de verschillende componenten met elkaar samenwerken.

## Sequentiediagram voor het versturen van chatberichten

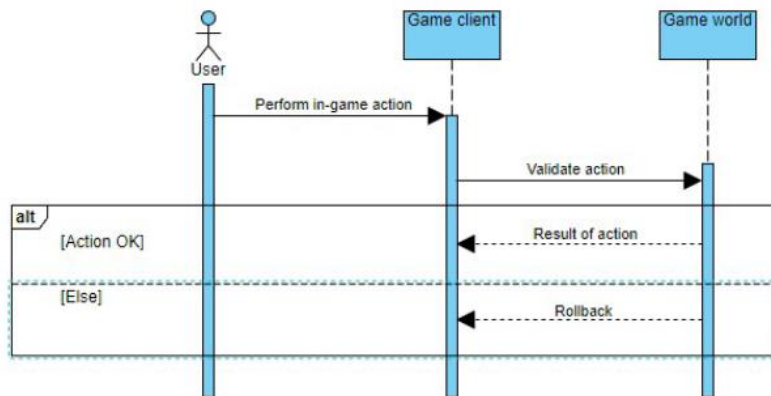


Figuur 5 Sequentiediagram voor het versturen van chatberichten

In Figuur 5 is het sequentiediagram te zien voor het versturen van chatberichten. Een chatbericht gaat door een aantal componenten heen, voordat het bericht bij de ontvanger(s) belandt. Als de gebruiker een chatbericht verstuurd, wordt dit via de Game world naar de Load and distribute management gestuurd. Deze roept vervolgens de correctie functie voor het type chatbericht aan op

Social services. Binnen Social services worden de ontvangers bepaald. Voor iedere ontvanger wordt het bericht vervolgens teruggecommuniceerd. Hiervoor wordt binnen Load and distribute management gezocht naar de serverlocatie van een gebruiker. Vervolgens wordt het bericht via de Game world naar de Game client gecommuniceerd.

Sequentiediagram voor het uitvoeren van acties in de spelwereld



Figuur 6 Sequentiediagram voor het uitvoeren van acties in de spelwereld

In figuur 4.6 is het sequentiediagram te zien voor het uitvoeren van acties in de spelwereld. In de basis wordt een actie client-side uitgevoerd en heeft de server alleen een controlerende taak. Dat wil zeggen dat de server controleert of een actie uitgevoerd kon/mocht worden. Wanneer een actie niet uitgevoerd kon/mocht worden zal de actie van de gebruiker worden teruggedraaid.

Voor acties met een random-factor (RNG-acties) is het proces iets anders. Hierbij wordt de berekening server-side uitgevoerd om valsspelen tegen te gaan. Acties waar vaak RNG bij voorkomt zijn bijvoorbeeld acties in een gevecht. Zo is er tijdens het aanvallen van een andere speler een kans dat de speler een 'kritieke klap' uitdeelt, die extra schade toebrengt. Hierbij moet worden voorkomen dat de speler de kans op deze 'kritieke klap' client-side verhoogt.

### Proof of concept

Het proof of concept is een Unity game waarin het event-driven systeem van de architectuur wordt aangetoond. Hierin worden acties van de gebruikers als events aangeroepen en worden deze op de server berekend en eventueel gecorrigeerd. Willekeurige acties, zoals het toepassen van schade binnen een bepaalde range of het afhandelen van critical hits, worden op de server afgehandeld zodat de client hier geen invloed op heeft. Het proof of concept bevat ook een component Load and distribute manager die het aantal online spelers op elke locatie bijhoudt. Hiermee wordt aangetoond dat de nodige data beschikbaar is om de MMORPG dynamisch op en af te schalen. Ook wordt in het proof of concept aangetoond dat de ontworpen client-server communicatie geschikt is voor de MMORPG. De source-code van het proof of concept is beschikbaar via GitHub (Alberts, Derks, Dassen, & Dang, 2022).

## Discussie

Binnen de architectuur zijn een aantal uitdagingen. In het proof of concept is echter alleen de werking van de communicatie tussen de Game world en de Game client gedemonstreerd, samen met het schaalbaarheidsaspect van de Game world. Er is hiervoor gekozen, omdat deze aspecten deel zijn van de meest belangrijke aspecten volgens de risicoanalyse. Daarnaast zijn deze aspecten het meest zichtbaar voor de eindgebruiker, wanneer deze niet goed (genoeg) werken.

Er is gekozen voor een Coordinator-component, zonder dat dit component een functie op zichzelf heeft. Het wordt uitsluitend gebruikt voor het doorsluizen van data. Er zou beargumenteerd kunnen worden dat dit dus een overbodig component is en er in plaats hiervan een directe verbinding tussen de componenten kan worden opgezet. De keuze voor de Coordinator is echter gemaakt om het systeem overzichtelijker te houden en te voorkomen dat een big ball of mud<sup>2</sup> ontstaat.

Een ander discussiepunt is de keuze voor de gemaakte sequentiediagrammen in de process view. Er is gekozen voor een diagram voor het uitvoeren van acties in de spelwereld en een diagram voor het versturen van chatberichten, terwijl er nog vele andere diagrammen gemaakt hadden kunnen worden. De keuze voor deze twee diagrammen is echter gemaakt, omdat deze op technisch vlak een grotere uitdaging zijn om te realiseren, rekening houdende met de niet-functionele eisen. Daarmee zijn deze onderdelen van de architectuur complexer en is het van belang dat de werking hiervan duidelijk is. Andere aspecten van de architectuur, zoals de Game store en Market, zijn op architectuurniveau minder complex. Daarom is het minder van belang om hiervoor concrete process views te maken.

---

<sup>2</sup> Een big ball of mud ontstaat wanneer er geen goede architectuur is en alles met alles wordt verbonden, zonder enige structuur.

## Conclusie

Aan de hand van de architectuurbeschrijving en het proof of concept is een architectuur neergezet voor een MMORPG. De architectuurbeschrijving is vastgelegd met Kruchten 4 + 1 en laat een viertal views zien die duidelijk weergeven hoe de architectuur op verschillende niveaus uit ziet. Het klassendiagram laat de logical view zien waar duidelijk is welke klassen aangemaakt horen te worden in de MMORPG. De process view wordt weergegeven via een tweetal sequentiediagrammen die tonen hoe de communicatie via chatberichten uit ziet en hoe clientacties worden verwerkt door de server. Voor de development view is een componentdiagram uitgewerkt die laat zien welke systemen aan de pas komen om deze architectuur te realiseren. Tot slot laat de physical view de daadwerkelijke hardware zien die nodig is om het componentdiagram te ondersteunen.

Om de werking van de architectuur aan te kunnen tonen is een proof of concept gerealiseerd met behulp van Unity. Binnen het prototype wordt een stuk van het systeem aangetoond dat gebaseerd is op de event-driven architectuur. Dit laat zien dat het mogelijk is om de server de client te laten controleren via verificatie en eventuele rollbacks. Daarnaast laat het proof of concept ook zien dat het realiseren van het load en distribute management component mogelijk is en dat deze in de toekomst in staat is om de Game world servers op en af te laten schalen. Tot slot laat dit proof of concept zien dat RNG-acties (zoals kritieke aanvallen) server-side worden afgehandeld om zo het valsspelen van spelers via hun eigen client te voorkomen.

Dit project houdt de deur open voor vervolgprojecten of zelfs realisatieprojecten die kunnen aantonen dat deze architectuur in zijn volledigheid werkt voor een MMORPG. Eventuele vervolgprojecten kunnen gaan over het verder uitwerken van de architectuur.

## Bijlage A – Overzicht van functionele eisen

Nummer	Eis	Prioriteit	Eigenaar
<b>Functionele Eise</b>			
<b>F1</b>	De gebruiker kan een account registreren.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>F2</b>	De gebruiker kan inloggen met een geregistreerd account.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>F3</b>	Na het inloggen kan de gebruiker een keuze maken uit een regio.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>F4</b>	De gebruiker kan een karakter aanmaken gekoppeld aan de gekozen regio.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>F5</b>	De gebruiker kan een aangemaakt karakter binnen de spel wereld inladen.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>F6</b>	De gebruiker kan karakters aanmaken.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>F7</b>	Bij het aanmaken van een karakter kiest de gebruiker een klasse.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>F8</b>	De karakters kunnen rondlopen in de spelwereld.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>F9</b>	De wereld bestaat uit locaties. Deze locaties bestaan weer uit verschillende componenten.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>F10</b>	De karakters kunnen tussen de locaties reizen.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>F11</b>	De locaties kunnen entiteiten bevatten (alle niet statische objecten binnen de wereld).	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>F12</b>	Er worden gegevens bijgehouden van de karakters van de gebruikers.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>F13</b>	De karakters kunnen spullen dragen.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>F14</b>	De karakters kunnen spullen (ver)kopen op een marktplaats.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>F15</b>	De karakters kunnen spullen onderling verhandelen.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>F16</b>	De karakters kunnen spullen (ver)kopen bij NPC's.	Should	O.d. projectgroep G.d. Miguel van de Laar
<b>F17</b>	De karakters kunnen spullen gebruiken voor het creëren van nieuwe spullen.	Should	O.d. projectgroep G.d. Miguel van de Laar
<b>F18</b>	De karakters kunnen vechten tegen vijanden.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>F19</b>	De karakters kunnen skills gebruiken.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>F20</b>	Vijanden kunnen spullen achterlaten wanneer verslagen.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>F21</b>	De karakters kunnen levelen door het verkrijgen van ervaringspunten.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>F22</b>	De karakters kunnen worden uitgerust met spullen.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>F23</b>	De karakters kunnen berichten naar elkaar sturen via een lokale chat.	Must	O.d. projectgroep G.d. Miguel van de Laar

<b>F24</b>	De karakters kunnen vriendenverzoeken versturen naar andere karakters.	Should	O.d. projectgroep G.d. Miguel van de Laar
<b>F25</b>	De karakters kunnen vrienden verwijderen.	Should	O.d. projectgroep G.d. Miguel van de Laar
<b>F26</b>	De karakters kunnen berichten naar elkaar sturen via een vrienden chat.	Should	O.d. projectgroep G.d. Miguel van de Laar
<b>F27</b>	De karakters kunnen een party vormen met andere karakters.	Should	O.d. projectgroep G.d. Miguel van de Laar
<b>F28</b>	De karakters kunnen berichten binnen hun party versturen via een party chat.	Should	O.d. projectgroep G.d. Miguel van de Laar
<b>F29</b>	De karakters kunnen Guilds vormen met andere karakters.	Should	O.d. projectgroep G.d. Miguel van de Laar
<b>F30</b>	De karakters kunnen berichten binnen hun Guild versturen via een Guild chat.	Should	O.d. projectgroep G.d. Miguel van de Laar
<b>F31</b>	De gebruikers kunnen quests (opdrachten met beloningen) uitvoeren.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>F32</b>	De karakters kunnen elkaar uitdagen voor een 1 op 1 gevecht.	Should	O.d. projectgroep G.d. Miguel van de Laar
<b>F33</b>	De karakters kunnen uit 5 klassen kiezen: Krijger, tovenaars, ranger, supporter en temmer	Should	O.d. projectgroep G.d. Miguel van de Laar
<b>F34</b>	Plaatsen kunnen getroffen worden door een vallende ster.	Should	O.d. projectgroep G.d. Miguel van de Laar
<b>F35</b>	Een plaats getroffen door een vallende ster wordt overgenomen door sterkere monsters.	Should	O.d. projectgroep G.d. Miguel van de Laar
<b>F36</b>	De karakters kunnen vechten tegen meerdere golven van monsters op een plaats getroffen door een vallende ster.	Should	O.d. projectgroep G.d. Miguel van de Laar
<b>F36a</b>	De laatste golf bevat een eindbaas.	Should	O.d. projectgroep G.d. Miguel van de Laar
<b>F36b</b>	Als de eindbaas verslagen is, verdwijnt de vallende ster.	Should	O.d. projectgroep G.d. Miguel van de Laar
<b>F37</b>	De karakters kunnen deelnemen aan een wereldbaas gevecht.	Should	O.d. projectgroep G.d. Miguel van de Laar
<b>F38</b>	De spelers kunnen via de game store cosmetics kopen met echt geld.	Should	O.d. projectgroep G.d. Miguel van de Laar
<b>F39</b>	Plaatsen in de wereld hebben een spelers limiet	Must	O.d. projectgroep G.d. Miguel van de Laar

## Bijlage B – Overzicht van niet-functionele eisen

Nummer	Eis	Prioriteit	Eigenaar
<b>Niet-functionele Eisen</b>			
<b>N1</b>	95% van de spelers heeft een maximale ping van 100ms.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>N2</b>	De MMORPG heeft een ongeplande downtime van maximaal 0,01%.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>N3</b>	De capaciteit van de MMORPG kan worden op- en afgeschaald afhankelijk van de behoeften.	Must	O.d. Miguel van de Laar G.d. Miguel van de Laar
<b>N4</b>	Data wordt volgens de regionale privacywetgeving (zoals AVG) verwerkt.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>N5</b>	Dataverzoeken moeten volgens de regionale wet- en regelgeving kunnen worden verwerkt.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>N6</b>	Spelerdata wordt gescheiden per regio.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>N7</b>	De MMORPG is ondersteund op Windows, MacOS, Xbox One, Xbox Series, Playstation 4, Playstation 5, Nintendo Switch.	Could	O.d. Miguel van de Laar G.d. Miguel van de Laar
<b>N8</b>	Componenten van een locatie in de spelwereld in de MMORPG zijn herbruikbaar voor locaties.	Should	O.d. projectgroep G.d. Miguel van de Laar
<b>N9</b>	Locaties in de spelwereld zijn modulair.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>N10</b>	De spelwereld is consistent voor alle spelers.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>N11</b>	De spelwereld is wijzigbaar.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>N12</b>	De data van de economie is consistent.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>N13</b>	Acties binnen de economie worden geauthentiseerd.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>N14</b>	Entiteiten in de spelwereld reageren binnen maximaal 100ms op acties van andere entiteiten.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>N15</b>	Entiteiten zijn consistent voor alle spelers.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>N16</b>	Entiteiten zijn herbruikbaar voor nieuwe entiteiten.	Should	O.d. projectgroep G.d. Miguel van de Laar
<b>N17</b>	Entiteiten zijn wijzigbaar.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>N18</b>	Gegevens van spelers zijn consistent.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>N19</b>	Gegevens van spelers zijn herstelbaar met een maximum verlies van 2 uur aan gegevens.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>N20</b>	Sociale interacties worden geauthentiseerd.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>N21</b>	Er wordt vertrouwelijk omgegaan met data van sociale interacties.	Must	O.d. projectgroep G.d. Miguel van de Laar



<b>N22</b>	Sociale interacties worden binnen maximaal 100ms verwerkt.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>N23</b>	Quests zijn foutbestendig voor alle acties van spelers.	Should	O.d. projectgroep G.d. Miguel van de Laar
<b>N24</b>	Quests zijn wijzigbaar.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>N25</b>	Quests zijn herbruikbaar voor nieuwe quests.	Should	O.d. projectgroep G.d. Miguel van de Laar
<b>N26</b>	Gebruikers worden tijdens het inloggen geauthentiseerd.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>N27</b>	Er wordt vertrouwelijk omgegaan met logingegevens.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>N28</b>	De setting van het spel is middeleeuws en fantasie.	Must	O.d. projectgroep G.d. Miguel van de Laar
<b>N29</b>	De MMORPG heeft een 2D top-down view.	Must	O.d. projectgroep G.d. Miguel van de Laar

## Bijlage C – Risicoanalyse

	Account management	Character management	Game world	Load and distribute management	Game client	Game store	Market	Online payment	Social systems	Coordinator
<b>Performance</b>	1*1 1	2*2 4	3*3 9	3*1 3	2*3 6	1*1 1	1*2 2	1*2 2	1*2 2	1*1 1
<b>Scalability</b>	1*2 2	1*3 3	2*3 6	1*2 2	1*1 1	1*2 2	1*2 2	1*2 2	1*3 3	1*2 2
<b>Elasticity</b>	1*1 1	2*3 6	3*3 9	1*2 2	1*1 1	1*2 2	1*2 2	1*1 1	2*3 6	1*2 2
<b>Availability</b>	1*2 2	1*3 3	1*3 3	1*3 3	1*1 1	1*2 2	1*2 2	2*2 4	1*2 2	1*3 3
<b>Fault tolerance</b>	1*3 3	1*2 2	2*2 4	1*1 1	2*1 2	1*3 3	1*3 3	1*3 3	1*1 1	1*1 1
<b>Security</b>	1*3 3	1*2 2	2*2 4	1*1 1	3*2 6	2*1 2	2*2 4	1*3 3	1*2 2	1*1 1
<b>Data integrity</b>	1*3 3	2*2 4	1*2 2	1*1 1	1*1 1	1*2 2	1*3 3	1*3 3	1*2 2	1*1 1
<b>Maintainability</b>	1*1 1	2*2 4	2*2 4	1*2 2	1*2 2	1*2 2	1*1 1	1*1 1	1*1 1	2*1 2
<b>Recoverability</b>	1*3 3	1*3 3	1*1 1	1*1 1	1*1 1	1*2 2	1*3 3	1*3 3	1*1 1	1*1 1

## Verwijzingen

Esports Extra. (sd). *The 8 Most Expensive Items In MMOS That Have Been Sold For Real Money*.

Opgehaald van Esports Extra: <https://esportsextras.com/the-8-most-expensive-items-in-mmoms-that-have-been-sold-for-real-money/>

Jayawardene, P. D. (2021, september 17). *4+1 Architectural view model in Software*. Opgehaald van

Medium: <https://medium.com/javarevisited/4-1-architectural-view-model-in-software-ec407bf27258>

Kruchten, P. (1995). Architectural Blueprints—The “4+1” View. *IEEE Software*, 42-50.

Object Modeling Group. (sd). *Unified Modeling Language*. Opgehaald van Unified Modeling

Language: <https://www.uml.org/>

Rehkopf, M. (sd). *Wat is een kanban-bord?* Opgehaald van Atlassian:

<https://www.atlassian.com/nl/agile/kanban/boards>

Scrum Company. (sd). *Wat is Agile*. Opgehaald van Scrum Company:

<https://www.scrumcompany.nl/wat-is-agile/>

Technavio. (2022, juni 8). *Massive Multiplayer Online (MMO) Games Market Size to Grow by USD*

*21.25 billion*. Opgehaald van Technavio: <https://www.prnewswire.com/news-releases/massive-multiplayer-online-mmo-games-market-size-to-grow-by-usd-21-25-billion-rise-of-professional-gaming-to-boost-growth--technavio-301563082.html>

Unity. (sd). *Unity*. Opgehaald van Unity: <https://unity.com/>

Wikipedia-community. (2021, december 13). *4+1 architectural view model*. Opgehaald van

Wikipedia: [https://en.wikipedia.org/wiki/4%2B1\\_architectural\\_view\\_model](https://en.wikipedia.org/wiki/4%2B1_architectural_view_model)

Wikipedia-community. (2022, mei 24). *Massively multiplayer online role-playing game*. Opgehaald

van Wikipedia: [https://nl.wikipedia.org/wiki/Massively\\_multiplayer\\_online\\_role-playing\\_game](https://nl.wikipedia.org/wiki/Massively_multiplayer_online_role-playing_game)

Wikipedia-community. (2022, april 5). *MoSCoW-methode*. Opgehaald van Wikipedia:

<https://nl.wikipedia.org/wiki/MoSCoW-methode>

Wikipedia-community. (2022, maart 2). *Software Architecture*. Opgehaald van Wikipedia:

[https://en.wikipedia.org/wiki/Software\\_architecture](https://en.wikipedia.org/wiki/Software_architecture)