2025

Verantwoording Document

INNOVATION TEAM
LECTORAAT ORGANISATIES IN DIGITALE TRANSITIE|

INHOUDSOPGAVE

INLEIDING

Dit document dient ter verantwoording van de keuzes die we in dit project hebben gemaakt, waaronder ontwerpbeslissingen, gebruikte technologieën en architecturale overwegingen. Verdere details van deze keuzes zijn, waar van toepassing, te vinden op de README-pagina in de GitHub-repository; dit wordt dan expliciet aangegeven.

BACK-END

SIMULATIE

Er zijn voor de simulatie een aantal beslissingen gemaakt om een goede balans tussen accuraatheid en complexiteit te behalen, welke hieronder zijn toegelicht:

ABM- FRAMEWORK

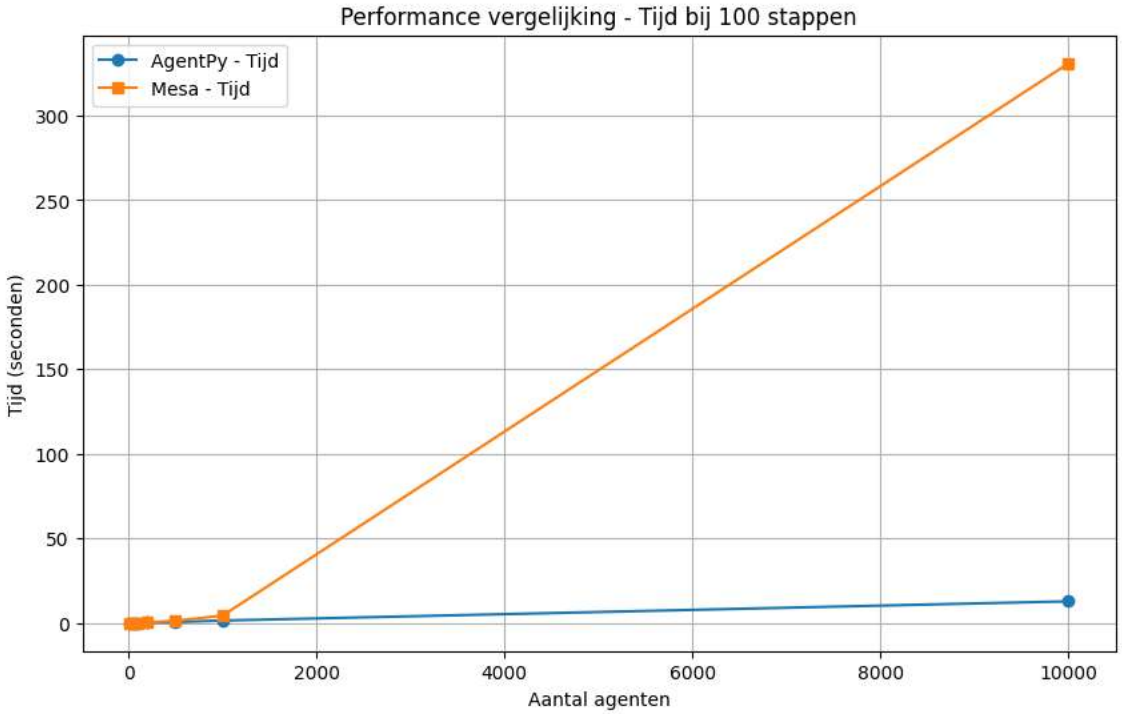
TESTS NOTEBOOK

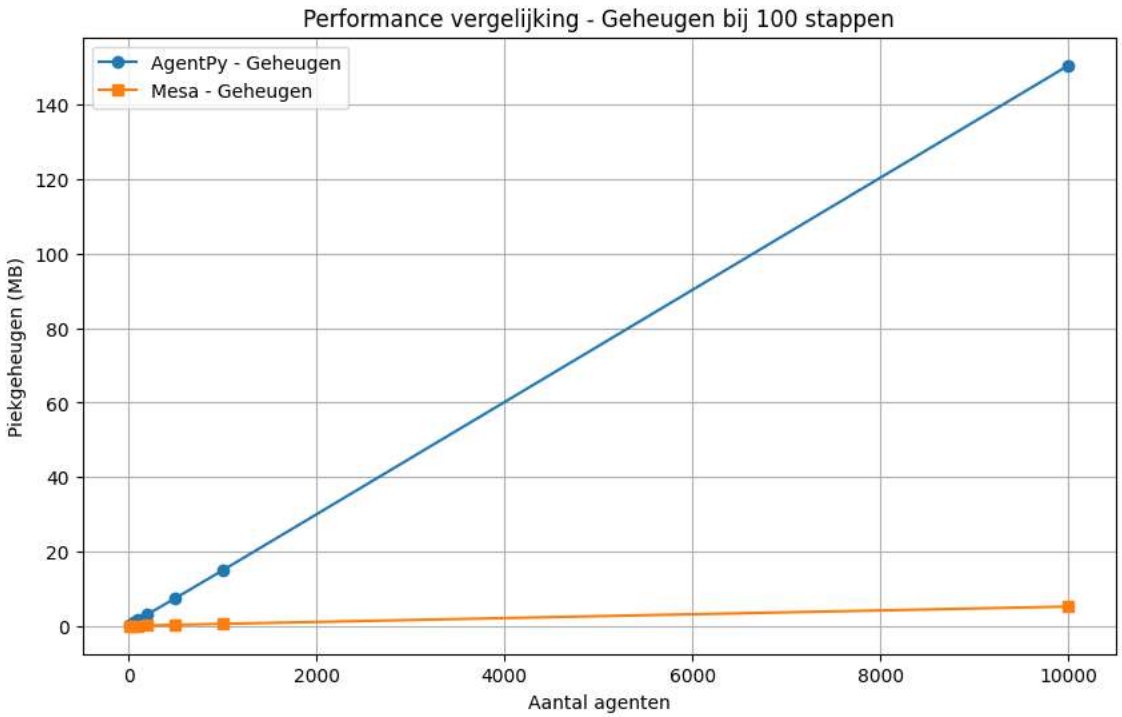
Om de verschillende frameworks objectief te kunnen vergelijken, is een Jupyter Notebook ontwikkeld waarin een reeks tests zijn uitgevoerd. Al snel werd duidelijk dat BPTK-Py en GAMA minder geschikt zijn voor de doeleinden van dit project. GAMA vereist het gebruik van een eigen programmeertaal, wat de integratie bemoeilijkt. BPTK-Py bleek lastig te installeren in de werkomgeving en is bovendien meer gericht op het modelleren van businesscampagnes dan op dynamische agent-based simulaties.

Hierdoor bleven twee frameworks over: Mesa en AgentPy. In het Jupyter Notebook is met beide frameworks een identieke simulatie opgezet: het bekende Money Model, waarin agenten willekeurig geld uitwisselen met elkaar. Dit model werd gebruikt om de prestaties van beide frameworks te vergelijken.

De simulaties zijn uitgevoerd met 100 tijdstappen en verschillende aantallen agenten (10, 50, 100, 200, 500, 1000 en 10.000). Daarbij is specifiek gekeken naar simulatiesnelheid en geheugengebruik, met als doel een schaalbare oplossing te vinden die in de toekomst gehost kan worden. De resultaten zijn weergegeven in twee overzichtelijke grafieken.

Figuur 1: Mesa VS Agent-py tijd per hoeveelheid agents





Figuur 2: Mesa vs AgentPy Geheugen gebruik

CONCLUSIE

Na het vergelijken van de verschillende frameworks is uiteindelijk gekozen voor Mesa als basis voor dit project. Zowel Mesa als AgentPy zijn lichtgewicht, open-source en goed geïntegreerd met het Python-ecosysteem, wat ze geschikt maakt voor het opzetten van agent-based simulaties en het verzamelen van data.

Hoewel AgentPy in de praktijk sneller bleek bij het uitvoeren van simulaties met grote aantallen agenten, heeft de uiteindelijke keuze voor Mesa te maken met de toekomstbestendigheid van het project. AgentPy wordt al geruime tijd niet meer actief onderhouden, wat risico's met zich meebrengt op het gebied van compatibiliteit en ondersteuning bij verdere ontwikkeling.

Mesa biedt daarentegen een actieve community, regelmatige updates en een modulair ontwerp met ingebouwde visualisatie-ondersteuning. Ondanks dat Mesa iets trager is in uitvoering, biedt het meer zekerheid op de lange termijn en betere ondersteuning voor toekomstige uitbreidingen van de simulatie.

Op basis van deze overwegingen is Mesa gekozen als het meest geschikte framework voor dit project.

ENVIRONMENT

In de environment zijn een aantal elementen vastgesteld, die hieronder worden toegelicht:

- Energy price: De energieprijs is standaard gebaseerd op het Nederlands gemiddelde.

STREET

- Een wijk bevat meerdere straten. De hoeveelheid straten is afhankelijk van het ingestelde hoeveelheid huishoudens. Het totaal aantal huishoudens zal altijd gelijk blijven aan het vooraf ingestelde aantal, maar het aantal huizen per straat kan verschillen. Het aantal huizen per straat is gecentreerd rond het gemiddelde aantal in Nederland, 32, maar er kunnen uitschieters tussen zitten zoals straten met 60 huizen. Dit om de wijk realistischer na te bootsen.

HOUSEHOLD

Ook in een huishouden zijn enkele beslissingen gemaakt:

- Solarpanel amount: Het aantal zonnepanelen dat een huishouden kan hebben, varieert tussen de 6, 8 of 10 zonnepanelen. Dit kan gebaseerd worden op dakoppervlak van echte woningen.
- Energy generation: Hoeveel energie de zonnepanelen opbrengen. Gebaseerd op gemiddelde opbrengst uit Nederland.
- In een huishouden wordt een lijst met residents aangemaakt en gevuld. De residents worden eerlijk verdeeld over de huishoudens. Dit kan ook gebaseerd zijn op gemiddelde huishouden uit Nederland.

RESIDENT

Ook in residents zijn bepaalde beslissingen gemaakt:

- Salary: Elke resident heeft een salaris, standaard gebaseerd op het mediaan inkomen van Nederland, met een lognormale verdeling. Dit is de meest accurate manier om een standaard inkomen van Nederland te simuleren.

- Attitude: Elke resident heeft een attitude, die bepaalt hoe welwillend een resident is voor verduurzaming. Deze waarde wordt willekeurig bepaald tussen 0 en 1.
- Modifiers: Er zijn enkele modifiers, voor attitude, environment en behavioral influence. Deze modifiers bepalen hoe belangrijk een resident een bepaald onderdeel vinden. Deze waarden worden willekeurig bepaald tussen 0 en 2.
- Alle bovenstaande waarden worden samen in een formule berekend tot een bepaalde waarde. Wanneer deze waarde boven een bepaalde threshold komt, in dit geval 0.5, zal de resident besluiten te verduurzamen. De hoogst mogelijke te behalen score is 6 (1 punt voor elke influence, maximaal x2 voor elke modifier) Deze score wordt vervolgens genormaliseerd om tot een waarde tussen de 0 en 1 te komen.
- Subjective norm: De subjective norm heeft betrekking tot de 'druk' die de omgeving van een inwoner op de inwoner kan uitoefenen. Er zijn hiervoor 3 niveaus: wijk, straat en direct. Op een wijkniveau, heeft elk huishouden in de wijk invloed op de subjective norm van de inwoner. Dit betekent dat wanneer een huis 4 straten verderop zonnepanelen aanlegt, de subjective norm van de inwoner omhoog gaat. Op straatniveau, hebben alleen de huishoudens in dezelfde straat invloed op de subjective norm van een gegeven inwoner. Bij een direct niveau hebben alleen de linker- en rechterbuur van een inwoner invloed. Er kan vrij gekozen worden tussen de 3 niveaus.

SOLAR PANELS

- ROI: De terugverdientijd van zonnepanelen wordt berekend aan de hand van de kosten van zonnepanelen, de huidige energie prijs en de opbrengst van de zonnepanelen. Deze terugverdientijd heeft invloed op de Behavioral influence van een resident en is ook anders voor elke resident.
- Solarpanel price: De prijs van zonnepanelen is standaard gebaseerd op de meest recente prijs per zonnepaneel in Nederland. Ieder jaar is er een kans dat de prijs van de zonnepanelen stijgt, in lijn met inflatie.

HEAT PUMP

MISCELLANEOUS

API

Omdat wij als team een keuzen moeten maken voor een bepaald framework wat past bij onze gekozen criteria namelijk: Documentatie, Support (gebaseerd op community en officiële), bekendheid (gebaseerd op onze ervaringen en algemeen), leeftijd (hoe oud is het framework). Daarnaast moet het framework vanuit de opdrachtgever open source zijn en hebben wij besloten dat voor handigheid met de simulatie het framework in

Python gemaakt moet zijn. Uit lijstjes zoeken op het internet en overleggen met het team zijn de volgende 4 geselecteerd:

Flask, bekend bij ons team en staat in de meeste tops.
FastAPI, bekend framework in de rest community.
Django, uit de een van de top lijsten gekozen.
Pyramid, uit een van de top lijsten gekozen.

De frameworks die gekozen zijn, zijn gebaseerd op een “majority vote”. Omdat er een limiet is aan hoeveel tijd wij hiervoor hebben afgesproken is er besloten om 4 frameworks te vergelelijken op deze punten.

Uit het onderzoek is de volgende tabel gekomen:

Framework	Flask	FastAPI	Django	Pyramid
Documentatie	Zeer goed	Goed	Zeer goed	Goed
Support (community / official)	Groot	Groeiend	Groot	Klein
Bekendheid (persoonlijk)	Bekend	Bekend	Onbekend	Onbekend
Bekendheid (algemeen)	Zeer bekend	Groeiend	Zeer bekend	Minder bekend
Leeftijd	Sinds 2010	2018	2005	2008

Gebaseerd op deze tabel is er besloten dat **Flask** de API framework is die wij gaan gebruiken om onze frontend met de simulatie te koppelen.

FRONT-END

TALEN & FRAMEWORKS

REACT

Voor de visuele weergave van de gegevens uit de simulatie is gekozen voor de ontwikkeling van een webapplicatie. Er is onderzoek gedaan naar diverse frameworks die geschikt zijn voor deze toepassing, waarbij een aantal criteria in overweging is genomen.

Vanuit de opdrachtgever zijn met name de volgende eisen als leidend aangemerkt:

- Het framework dient open source te zijn.
- Het framework moet de mogelijkheid bieden om de applicatie online te delen.

Daarnaast zijn aanvullende criteria vastgesteld op basis van relevante kennis en inzichten uit de gevolgde opleiding. Deze criteria richten zich op aspecten die van belang zijn bij het selecteren van een geschikt framework voor de front-endontwikkeling van een webapplicatie.

Criteria:	**Reden:**
Ervaring	We geven de voorkeur aan een framework waarmee de developers al vertrouwd zijn, zodat de ontwikkeling soepeler en efficiënter verloopt.
Documentatie	Goede ondersteuning en documentatie zorgen ervoor dat toekomstige developers, zelfs met weinig ervaring in het framework, gemakkelijk aan de applicatie kunnen werken.
Schaalbaarheid	Naarmate de applicatie in de toekomst mogelijk groeit, willen we een framework dat goed schaalbaar is.
Community	Een actieve community helpt bij het oplossen van problemen, zorgt voor nieuwe updates en maakt het makkelijker om van anderen te leren.
Snelheid	We willen een framework dat snel werkt, vooral wanneer de simulatie veel data bevat. Alle data moet dan op tijd en zonder vertraging getoond worden.

Op basis van deze criteria hebben we verschillende frameworks onderzocht en met elkaar vergeleken.

Framework	**Ervaring**	**Documentatie**	**Schaalbaarheid**	**Community**	**Snelheid**	**Open-Source**	**Online Hosting**
React	Developers hebben er ervaring in	React heeft uitgebreide documentatie	Goed schaalbaar en ook flexibel door het toevoegen van tools/bibliotheken.	Heeft een grote community, ondersteund door Meta	Zeer snel	Ja	Ja
Angular	Developers hebben er geen ervaring in	Angular heeft uitgebreide documentatie	Zeer schaalbaar doordat het een volledig framework is met ingebouwde oplossingen.	Heeft een grote community, ondersteund door Google	Snel	Ja	Ja
Vue	Developers hebben er geen ervaring in	Vue heeft uitgebreide documentatie	Goed schaalbaar vanwege het toevoegen van uitbreidingen.	Heeft een groeiende community, wordt niet ondersteund door tech gigant	Zeer Snel	Ja	Ja

De uiteindelijke keuze is gevallen op React. Dit framework beschikt over een grote en actieve community, evenals uitgebreide documentatie, wat de ontwikkeling ondersteunt. React is open source, biedt voldoende schaalbaarheid en staat bekend om zijn snelle prestaties. Daarnaast maakt het framework toekomstige online hosting mogelijk. Daarmee voldoet React aan zowel de gestelde eisen van de opdrachtgever als aan de aanvullende technische en praktische criteria.

ARCHITECTUUR

De architectuur van de front-end is gebaseerd op het Model-View-Controller (MVC)-patroon, vanwege de duidelijke scheiding van verantwoordelijkheden binnen de applicatie. Deze scheiding bevordert de onderhoudbaarheid, schaalbaarheid en testbaarheid van de code. Het MVC-patroon is een aanbevolen aanpak binnen de gevolgde opleiding en wordt vaak toegepast in vergelijkbare projecten.

- Model (Services): Verantwoordelijk voor data en logica, zoals API-aanroepen en verwerking van externe bronnen. Door isolatie van deze logica wordt hergebruik vereenvoudigd en blijft de presentatie gescheiden van de dataverwerking.

- View (Components, Pages): Richt zich uitsluitend op presentatie en de gebruikersinterface. Binnen een React-context zorgt dit voor visueel georiënteerde, herbruikbare componenten zonder directe afhankelijkheid van de achterliggende logica.

- Controller: Vormt de tussenlaag tussen data en presentatie. Verwerkt gegevens uit de services en stuurt componenten aan. Deze opzet bevordert transparantie en maakt het eenvoudiger om logica aan te passen bij veranderende eisen.

Toepassing van het MVC-patroon resulteert in een modulaire front-endstructuur waarin afzonderlijke onderdelen eenvoudig door te ontwikkelen of te vervangen zijn, zonder invloed op andere lagen. Dit draagt bij aan een hogere ontwikkelsnelheid en verbeterde codekwaliteit op de lange termijn.

De toepassing en uitleg van deze architectuur voor het project staat in de README van de GitHub-repository.