

Architecturale Beoordeling en Strategische Analyse van het S1mpleTrader V2 Platform

Executive Summary

Dit rapport presenteert een diepgaande analyse van het S1mpleTrader V2-platform, gericht op zowel de softwarearchitectuur als de toegepaste kwantitatieve analysemethoden. De beoordeling bevestigt dat de kernarchitectuur van S1mpleTrader V2 robuust, modern en uitzonderlijk goed ontworpen is. De fundamentele principes van "Plugin First", "Contract-Driven" ontwikkeling en een strikte "Separation of Concerns" vormen een best-in-class basis voor het creëren van modulaire en testbare handelsstrategieën. De implementatie van een Assembly Team voor het dynamisch bouwen van componenten en een strikt DTO-gebaseerde datastroom getuigen van een geavanceerd architectonisch inzicht dat de stabiliteit en voorspelbaarheid van het systeem waarborgt.

Ondanks deze solide basis identificeert de analyse drie fundamentele "blinde vlekken" die de analytische diepgang van het platform momenteel beperken en de evolutie naar een institutioneel kaliber systeem in de weg staan. Deze blinde vlekken zijn geen implementatiefouten, maar conceptuele beperkingen die inherent zijn aan de huidige focus van het systeem:

1. **De OHLCV-Data Limiet:** Het platform is fundamenteel ontworpen rond de analyse van geaggregeerde OHLCV-candlestickdata. Dit model verliest per definitie de cruciale informatie over de onderliggende markt-microstructuur – de daadwerkelijke stroom van koop- en verkooporders die de prijsvorming stuurt. Het systeem analyseert het *resultaat* van de markt, niet het *proces*.
2. **Het Enkelvoudige Tijdsperspectief:** De huidige architectuur verwerkt data binnen één enkel, vooraf gedefinieerd tijdframe per strategie-uitvoering. Dit verhindert de toepassing van Multi-Timeframe Analyse (MTFA), een essentiële techniek waarbij de trend op een hoger tijdframe wordt gebruikt om de context te bepalen voor entries op een lager tijdframe.
3. **Een Ontluikende Benadering van Executie en Risicobeheer:** De huidige focus ligt

sterk op signaalgeneratie. De executie- en risicobeheerfasen, hoewel aanwezig, zijn nog rudimentair. Geavanceerde concepten zoals execution algorithms (bijv. VWAP), dynamische position sizing op portefeuilleniveau en correlatie-analyse tussen strategieën ontbreken.

Dit rapport concludeert met een reeks strategische aanbevelingen om deze blinde vlekken aan te pakken. Deze aanbevelingen omvatten de evolutie van de data-architectuur naar een multi-granulariteitsmodel dat zowel tick- als OHLCV-data kan verwerken, de uitbreiding van het plugin-ecosysteem met nieuwe interfaces voor orderstroom- en multi-timeframe-analyse, en de architectuur van een volwaardig, proactief risicobeheersysteem op portefeuilleniveau. Het volgen van deze roadmap zal S1mpleTrader V2 transformeren van een geavanceerd backtesting-framework naar een institutioneel-grade platform dat in staat is tot diepgaande en voorspellende marktanalyse.

Sectie 1: Beoordeling van het Kernarchitecturale Raamwerk

1.1. Het "Plugin First" Paradigma: Een Model van Uitbreidbaarheid

De architectuur van S1mpleTrader V2 is gebouwd op een "Plugin First" filosofie, een kernprincipe dat modulariteit, onafhankelijke testbaarheid en systeembrede uitbreidbaarheid bevordert.¹ Deze benadering wordt technisch gerealiseerd door de componenten binnen het

backend/assembly package, die gezamenlijk fungeren als het Assembly Team. De PluginRegistry is verantwoordelijk voor het ontdekken van alle beschikbare plugins door de plugins/ directory te scannen en de plugin_manifest.yaml bestanden te valideren. Dit manifest fungeert als een "ID-kaart" voor elke plugin, waarin metadata zoals type, versie en afhankelijkheden worden gedeclareerd. Dit mechanisme voor dynamische ontdekking is krachtig, omdat het de kernapplicatie volledig ontkoppelt van de specifieke implementaties van de strategielogica.¹

De WorkerBuilder is de fabriek die, op basis van de informatie uit de PluginRegistry en de gebruikersconfiguratie, een specifieke plugin-instantie creëert. Dit proces omvat het dynamisch laden van de worker-klasse en het Pydantic-schemamodel, het valideren van de door de gebruiker opgegeven parameters tegen dit schema, en het injecteren van afhankelijkheden zoals een specifieke logger.¹ Deze aanpak centraliseert de complexe logica

van objectcreatie en -validatie, waardoor de rest van het systeem met gegarandeerd correct geconfigureerde en gevalideerde componenten kan werken.

Een bijzonder sterk punt in dit ontwerp is de DependencyValidator. Deze component analyseert de dependencies en provides velden in de manifesten van de structural_context plugins om de integriteit van de data-pijplijn te valideren *voordat* de executie begint.¹ Door te controleren of elke plugin de datakolommen ontvangt die hij verwacht, worden runtime-fouten door een incorrecte configuratie effectief voorkomen. Dit toont een volwassen architectonische benadering die robuustheid boven alles stelt. Ten slotte scheidt de

EngineBuilder op correcte wijze de assemblage van de statische data-verrijkingspijplijn (de STRUCTURAL_CONTEXT fase) van de assemblage van de actieve, event-driven workers van de StrategyEngine.¹ Dit versterkt de scheiding der verantwoordelijkheden tussen de eenmalige voorbereiding van data en de continue, bar-voor-bar verwerking van signalen.

1.2. Contract-Gedreven Datastroom: Waarborgen van Systemische Integriteit

Het "Contract-Driven" principe is een andere hoeksteen van de robuustheid van S1mpleTrader V2. Het rigoureuze gebruik van Pydantic BaseModel voor alle Data Transfer Objects (DTO's), zoals gedefinieerd in de backend/dtos directory, dwingt een strikt contract af voor alle data die tussen componenten en lagen wordt uitgewisseld.¹ Dit voorkomt een hele klasse van runtime-fouten die kunnen ontstaan door incorrecte datatypes of ontbrekende velden, en maakt de datastroom door het systeem heen volledig voorspelbaar en traceerbaar.

De DTO-keten vormt de ruggengraat van de signaalverwerkingspijplijn. Een handelsidee begint zijn leven als een Signal DTO, dat de meest basale informatie bevat: wat, waar en wanneer.¹ Vervolgens wordt dit object sequentieel verrijkt en genest door de verschillende fasen van de workflow, resulterend in een progressie van

Signal naar EntrySignal, RiskDefinedSignal, TradePlan, en uiteindelijk RoutedTradePlan. Elk DTO vertegenwoordigt een specifiek stadium van de besluitvorming en voegt een nieuwe laag van strategische informatie toe. Deze geneste structuur creëert een duidelijke, onveranderlijke (immutable) en volledig traceerbare data-afstamming (data lineage) voor elk handelsidee. De unieke correlation_id die vanaf het begin wordt toegewezen en door de hele keten wordt gepropageerd, is essentieel voor debugging en analyse achteraf.¹

Een elegant detail in dit ontwerp is de DirectiveFlattener. Deze utility-klasse is verantwoordelijk voor het transformeren van het diep geneste, interne RoutedTradePlan DTO naar een platte, eenvoudige ExecutionDirective.¹ Dit ontkoppelt op effectieve wijze de complexe, gelaagde

interne staat van de strategie-engine van het simpele, universele contract dat de

ExecutionHandler laag verwacht. Dit is een uitstekend voorbeeld van het Interface Segregation Principle, waarbij de uitvoerende laag niet wordt belast met de complexiteit van de interne datastructuren van de besluitvormingslaag.

1.3. Scheiding van Zorgen: Een Schone en Onderhoudbare Structuur

De applicatie hanteert een klassieke, drielaagse architectuur (Frontend → Service → Backend) die een strikte eenrichtingsafhankelijkheid afdwingt.¹ Deze scheiding isoleert de verantwoordelijkheden effectief: de backend fungeert als een onafhankelijke, herbruikbare "engine" met de kernlogica; de service-laag orkestreert deze kerncomponenten tot complete business workflows; en de frontend-laag verzorgt de gebruikersinteractie. Het ontwerp van de backend als een onafhankelijke library is een bijzonder sterk punt, omdat het de kernfunctionaliteit beschikbaar maakt voor verschillende contexten (CLI, Web API, of zelfs integratie in andere systemen) zonder aanpassingen.

De conceptuele scheiding tussen de StrategyOperator (de "wat"), ExecutionEnvironment (de "waar"), Assembly Team (de "hoe"), en Portfolio (de "financiële staat") is helder gedefinieerd en correct geïmplementeerd.¹ De

Portfolio klasse, zoals blijkt uit de code, gedraagt zich inderdaad als een "dom grootboek".¹ Het beheert kapitaal, houdt actieve en gesloten transacties bij, en verwerkt de sluiting van posities op basis van SL/TP-niveaus. Cruciaal is dat het geen enkele strategische beslissingslogica bevat; het is een pure state-manager die opdrachten uitvoert. Dit voorkomt een verstrengeling van logica en is een fundamentele voorwaarde voor een schaalbaar en onderhoudbaar systeem.

Deze architectuur legt de basis voor een multi-strategie, multi-asset portefeuillesysteem. De documentatie beschrijft een conceptuele PortfolioSupervisor die meerdere StrategyOperator-instanties beheert.¹ De huidige

StrategyOperator is ontworpen om één enkele strategieconfiguratie (AppConfig) uit te voeren. Het bestaan van een PortfolioSupervisor-concept impliceert dat het systeem is ontworpen om niet slechts één strategie te draaien, maar een *portfolio van strategieën*. Dit is een aanzienlijke architectonische kracht. Het betekent dat het systeem kan evolueren van een single-strategy backtester naar een volwaardig fondsbeheerplatform zonder een fundamentele herontwerp. Het Portfolio-object, dat al meerdere gelijktijdige transacties van verschillende activa kan beheren, is de sleutelcomponent die deze schaalbaarheid mogelijk maakt. Dit ontwerp ondersteunt inherent de horizontale schaalbaarheid van strategieën en biedt een duidelijk pad

naar de implementatie van geavanceerde logica op portefeuilleniveau, wat direct aansluit bij de later te bespreken blinde vlekken.

Sectie 2: De Quant Workflow: Een Fundament voor Prijsactie-Analyse

2.1. Deconstructie van de 9-Fasen Trechter

De kern van de strategie-executie in S1mpleTrader V2 is een systematische, sequentiële trechter die een abstract marktidee transformeert in een concreet, uitvoerbaar handelsplan. Hoewel de overzichtsdocumentatie spreekt van een "6-fasen workflow", onthult de implementatie in `backend/core/enums.py` een meer granulaire, 9-fasen trechter.¹ Deze 9 fasen bieden een nauwkeuriger en logischer beeld van het volledige proces, waarbij de "Trade Constructie" fase uit het 6-fasen model wordt opgesplitst in zijn logische subcomponenten.

De workflow kan worden onderverdeeld in twee hoofddelen:

Deel 1: Data Voorbereiding (Fasen 1-2)

Deze fasen worden uitgevoerd door de `ContextBuilder` en zijn gericht op het verrijken van de ruwe marktdata.

- **Fase 1: REGIME_FILTER:** Deze fase classificeert de algemene marktomstandigheden (bv. trending, ranging) zonder data te filteren. Een `adx_trend_filter` plugin zou hier bijvoorbeeld een kolom regime toevoegen.¹
- **Fase 2: STRUCTURAL_CONTEXT:** Deze fase maakt de markt "leesbaar" door micro-context toe te voegen, zoals marktstructuur, trends of technische indicatoren. De `ema_detector` plugin is een voorbeeld dat een EMA-kolom toevoegt aan de `DataFrame`.¹

Deel 2: Signaalverwerking (Fasen 3-9)

Deze fasen worden beheerd door de `StrategyEngine` in een event-driven lus, waarbij elke databar (tick) wordt verwerkt.

- **Fase 3: SIGNAL_GENERATOR:** Identificeert een specifieke, actiegerichte trigger. Een `fvg_entry_detector` zou hier een `Signal DTO` genereren wanneer aan de voorwaarden is voldaan.¹
- **Fase 4: SIGNAL_REFINER:** Valideert het signaal op basis van secundaire criteria. Een `volume_spike_refiner` keurt een signaal bijvoorbeeld alleen goed als er voldoende volume

is.¹

- **Fase 5: ENTRY_PLANNER:** Bepaalt de precieze entry-prijs, waardoor een Signal wordt omgezet in een EntrySignal DTO.
- **Fase 6: EXIT_PLANNER:** Definieert het initiële risicomanagement door stop-loss en take-profit prijzen toe te voegen, resulterend in een RiskDefinedSignal DTO.
- **Fase 7: SIZE_PLANNER:** Berekent de definitieve positiegrootte op basis van het risico en het beschikbare kapitaal, wat leidt tot een TradePlan DTO.
- **Fase 8: ORDER_ROUTER:** Vertaalt het strategische plan naar concrete, technische executie-instructies (bv. market vs. limit order), en creëert een RoutedTradePlan DTO.
- **Fase 9: CRITICAL_EVENT_DETECTOR:** Voert een finale, systeem-brede risicocheck uit, los van de individuele trade. Een max_drawdown_overlay kan hier een CriticalEvent genereren.¹

Deze gestructureerde trechter zorgt ervoor dat elke beslissing op een logische en traceerbare manier wordt opgebouwd, waarbij elke fase voortbouwt op de output van de vorige.

2.2. Het OHLCV-Datamodel: De Wereld als een Reeks Staven

Het analytische universum van S1mpleTrader V2 begint en eindigt met OHLCV-data (Open, High, Low, Close, Volume). De DataLoader component in backend/data/loader.py is verantwoordelijk voor het inlezen van deze data uit CSV-bestanden, het converteren van de timestamp-kolom naar een datetime index, en het uitvoeren van basis-opschoning zoals het verwijderen van rijen met ontbrekende waarden.¹ Dit vormt de

initial_df.

Vanaf dit punt is het gehele systeem gebouwd op de aanname van deze bar-gebaseerde datastructuur. De DependencyValidator, bijvoorbeeld, controleert of de initiële set van beschikbare kolommen {'open', 'high', 'low', 'close', 'volume'} bevat.¹ De

ContextBuilder neemt deze OHLCV DataFrame als input en verrijkt deze met extra kolommen.¹ Alle initiële plugins, zoals de

ema_detector, zijn ontworpen om op deze OHLCV-structuur te opereren.

Deze keuze voor OHLCV als het fundamentele data-artefact heeft een diepgaande implicatie: het systeem is inherent gebaseerd op geaggregeerde data. Een enkele 15-minuten OHLCV-staaf is een samenvatting van potentieel duizenden individuele transacties (ticks) die binnen die periode hebben plaatsgevonden. Het reduceert een complexe, dynamische interactie tussen kopers en verkopers tot slechts vijf datapunten. Twee 15-minuten staven

kunnen identieke OHLCV-waarden hebben, maar de onderliggende dynamiek kan radicaal verschillen. De ene kan het resultaat zijn van een langzame, gestage prijsstijging, terwijl de andere het gevolg kan zijn van een volatiele strijd met een scherpe daling gevolgd door een snel herstel. Het huidige systeem, door zijn ontwerp, kan geen onderscheid maken tussen deze twee scenario's; het ziet alleen het geaggregeerde eindresultaat. Deze fundamentele ontwerpkeuze betekent dat het systeem blind is voor het *proces* van prijsvorming en alleen kan reageren op de *uitkomst* ervan. Dit vormt de basis van de belangrijkste blinde vlek van het platform.

2.3. Sterktes en Inherente Beperkingen van Candlestick-Centrische Analyse

De focus op candlestick-data is een logisch en krachtig uitgangspunt. Candlestick patronen zoals de Engulfing, Doji, Hammer, en Morning/Evening Star zijn een hoeksteen van de technische analyse. Ze bieden een visueel intuïtieve en effectieve manier om de psychologie van de markt te interpreteren en potentiële keerpunten of voortzettingen van trends te identificeren.² De

`fvg_entry_detector` plugin is een voorbeeld van een strategie die waarschijnlijk op dergelijke prijsactiepatronen is gebaseerd.¹

De kracht van deze patronen is echter beschrijvend, niet inherent voorspellend. Hun betrouwbaarheid neemt exponentieel toe wanneer ze worden bevestigd door aanvullende factoren, met name volume.⁶ Een breakout op hoog volume is significant betrouwbaarder dan een op laag volume. Het S1mpleTrader V2-platform erkent dit principe, zoals blijkt uit de aanwezigheid van een

`volume_spike_refiner` plugin, die bedoeld is om signalen te valideren op basis van volume-activiteit.¹

Hier stuiten we echter op de inherente beperking van het OHLCV-model. Het volume-cijfer op een candlestick-staaf is een aggregaat; het vertegenwoordigt het totale verhandelde volume in die periode. Het vertelt ons *niet* of dit volume voornamelijk bestond uit agressieve kopers of agressieve verkopers. Een hoog volume op een bullish candle kan betekenen dat er veel kopers waren, maar het kan ook betekenen dat er een enorme hoeveelheid verkoopdruk was die ternauwernood werd geabsorbeerd door de kopers. Deze ambiguïteit is een kritieke beperking. Zonder de mogelijkheid om de balans tussen koop- en verkoopdruk *binnen* de kaars te analyseren, mist het systeem de context die nodig is om de ware kracht en intentie achter een prijsbeweging te beoordelen. Dit gebrek aan inzicht in de interne dynamiek van de

kaars vormt de overgang naar de diepere blinde vlekken van het systeem.

Sectie 3: Blinde Vlek Analyse: De Wereld Binnenin de Kaars Ontdekken

3.1. De Informatiekloof: Van OHLCV naar Markt-Microstructuur

De kern van de blinde vlekken van het S1mpleTrader V2-platform ligt in de informatiekloof tussen de analyse van geaggregeerde OHLCV-data en de analyse van de markt-microstructuur. Markt-microstructuur is de studie van hoe de specifieke mechanismen van een beurs en het gedrag van de marktdeelnemers de prijsvorming, liquiditeit en transactiekosten beïnvloeden.⁷ Waar OHLCV-data het

resultaat van de veiling in een bepaalde periode toont, onthullen microstructuurdata (zoals tick-data en orderboekdata) de veiling *terwijl deze plaatsvindt*. Dit is het fundamentele verschil tussen reactieve analyse (reageren op afgeronde patronen) en proactieve of zelfs voorspellende analyse (anticiperen op basis van de intenties van de deelnemers).

Door uitsluitend op OHLCV-data te vertrouwen, mist het systeem het vermogen om cruciale vragen te beantwoorden: Wie heeft de controle, de kopers of de verkopers? Waar bevinden zich grote clusters van liquiditeit (limietorders) die als steun of weerstand kunnen fungeren? Is het hoge volume in een kaars het gevolg van agressieve kopers die de prijs opdrijven, of van passieve verkopers die een grote order absorberen? Het antwoord op deze vragen ligt verborgen in de data "binnenin" de kaars, data die momenteel buiten het bereik van het platform valt.

3.2. Blinde Vlek 1: De Afwezigheid van Orderstroom-Intelligentie

Orderstroom (Order Flow) trading is de analyse van de daadwerkelijke stroom van uitgevoerde koop- en verkooporders in real-time.¹⁰ In plaats van alleen naar het totale volume te kijken, analyseert het de

Volume Delta: het netto verschil tussen het volume dat is geïnitieerd door kopers (market buy

orders) en het volume dat is geïnitieerd door verkopers (market sell orders) op elk prijsniveau.¹³ Dit biedt een directe meting van de agressie in de markt. Het huidige systeem mist de tools en datagranulariteit om deze analyse uit te voeren.

Sleuteltechnieken die ontbreken:

- **Volume Profile & Market Profile:** Deze technieken organiseren het volume horizontaal per prijsniveau, in plaats van verticaal per tijdsperiode. Dit creëert een histogram dat onthult op welke prijsniveaus de meeste (High Volume Nodes - HVNs) en de minste (Low Volume Nodes - LVNs) transacties hebben plaatsgevonden. HVNs representeren "eerlijke" prijzen of institutionele "value areas" en fungeren vaak als krachtige magneten voor de prijs, terwijl LVNs gebieden van "oneerlijke" prijzen representeren waar de prijs snel doorheen beweegt.¹⁴ De Point of Control (POC) is het prijsniveau met het hoogste volume en is vaak het belangrijkste niveau om in de gaten te houden.
- **Footprint Charts:** Dit is een geavanceerde grafiektype dat binnen elke candlestick laat zien hoeveel koop- versus verkoopvolume op elk prijsniveau is uitgevoerd.¹⁰ Hiermee kunnen traders cruciale patronen identificeren zoals *absorption* (wanneer een grote hoeveelheid agressieve orders wordt geabsorbeerd door passieve orders zonder dat de prijs significant beweegt) en *exhaustion* (wanneer een trend momentum verliest omdat de dominante partij geen orders meer plaatst).¹¹

De volgende tabel illustreert concreet de analytische meerwaarde van orderstroom-context ten opzichte van pure prijsactie-analyse, en benadrukt daarmee direct de blinde vlek van het huidige systeem.

Tabel 1: Contrasterende Analyse van Prijsactie versus Orderstroom-Confirmatie

Candlestick Signaal (Huidige Systeemperspectief)	Orderstroom Context (De Blinde Vlek)	Verfijnde Interpretatie
Bullish Hammer op een Steunniveau ³	Hoog volume met een aanzienlijke negatieve delta aan de onderkant van de wick. Dit duidt op veel agressieve verkopers die werden geabsorbeerd door passieve limiet-kooporders.	Hoge Zekerheid Long: Verkopers hebben agressief geprobeerd de prijs omlaag te drukken, maar stuitten op een muur van koopinteresse. De hammer vertegenwoordigt een mislukte veiling naar beneden. De kans op een opwaartse beweging is

		groot.
Bullish Hammer op een Steunniveau ³	Laag volume met een positieve of neutrale delta aan de onderkant van de wick. Dit duidt op een gebrek aan verkoopdruk, niet op de aanwezigheid van sterke koopdruk.	Lage Zekerheid Long: De hammer is gevormd omdat de verkopers simpelweg afwezig waren, niet omdat sterke kopers de markt overnamen. Dit patroon is kwetsbaar en kan gemakkelijk falen als er alsnog verkoopdruk ontstaat.
Breakout Boven een Weerstandsniveau	De breakout vindt plaats met een afnemende volume delta en beweegt rechtstreeks in een High Volume Node (HVN) uit een eerdere Volume Profile analyse.	Hoge Waarschijnlijkheid van een Fakeout: De breakout mist overtuiging van kopers (afnemende agressie) en loopt vast in een historisch gebied waar veel volume is verhandeld, wat duidt op sterke weerstand en aanbod.
Bearish Engulfing Candle na een Uptrend ²	De engulfing candle toont een grote koop-onbalans (veel meer agressieve kopers dan verkopers) aan de onderkant van de candle, die er desondanks niet in slaagde de prijs terug omhoog te duwen.	Exhaustion (Uitputtings) Signaal: Kopers hebben een laatste, agressieve poging gedaan om de trend voort te zetten, maar werden volledig overweldigd door passief aanbod. Dit signaleert uitputting van de koopkracht en een hoge waarschijnlijkheid dat de verkoopdruk zal aanhouden.

3.3. Blinde Vlek 2: Het Enkelvoudige Tijdspectief

Een robuuste handelsstrategie vereist context, en de meest cruciale context is de trend op een hoger tijdframe. Multi-Timeframe Analyse (MTFA) is een fundamentele "top-down" benadering waarbij een handelaar eerst de primaire trend op een langetermijngrafiek (bv. dagelijks) vaststelt, en vervolgens alleen zoekt naar instapmogelijkheden in de richting van die trend op een kortere termijngrafiek (bv. per uur).²⁰ Dit filtert een aanzienlijke hoeveelheid ruis en transacties met een lage waarschijnlijkheid weg.

De huidige architectuur van S1mpleTrader V2 is hier niet op ingericht. De StrategyOperator en ContextBuilder zijn ontworpen om één enkele DataFrame voor één specifiek tijdframe te verwerken, zoals gedefinieerd in het run_schema.yaml.¹ Er is geen native mechanisme voor een plugin die op de 1-uurs data draait om de staat van een indicator (bijvoorbeeld een 200-daags voortschrijdend gemiddelde) op de dagelijkse grafiek op te vragen.

Een naïeve implementatie van MTFA kan bovendien leiden tot een verraderlijke valkuil: de "look-ahead bias".²³ Stel dat een 1-uurs strategie de richting van de dagelijkse EMA wil weten. Om 10:00 uur 's ochtends sluit de 1-uurs kaars en wordt de strategie uitgevoerd. Als het systeem op dat moment simpelweg de

huidige, nog vormende waarde van de dagelijkse EMA opvraagt (die data tot 10:00 uur bevat), gebruikt het informatie uit de toekomst. De dagelijkse kaars sluit pas over vele uren. Een correcte, bias-vrije implementatie moet ervoor zorgen dat de strategie om 10:00 uur alleen toegang heeft tot de waarde van de *vorige, voltooide* dagelijkse EMA. Dit is een subtiele maar kritieke architectonische uitdaging. Het systeem heeft een robuuste manier nodig om historische, voltooide staven van meerdere tijdlijnen tegelijkertijd te beheren en beschikbaar te stellen, zonder toekomstige informatie te lekken. Het huidige model met één enkele DataFrame is hiervoor ontoereikend.

3.4. Blinde Vlek 3: Van Signaalgeneratie naar Optimale Executie en Risico-Overlay

De huidige workflow legt een sterke nadruk op het genereren van een signaal (Fasen 3-4) en het construeren van een basis handelsplan (Fasen 5-7). De laatste fasen, executie en risicobeheer, zijn echter nog onderontwikkeld en vertegenwoordigen een significante blinde vlek.

Executie als een Strategie: De OrderRouter plugins lijken momenteel een TradePlan simpelweg te vertalen naar een basisordertype zoals 'market' of 'limit'.¹ Voor institutionele

handelaren is de

manier waarop een order wordt uitgevoerd echter een strategie op zich, gericht op het minimaliseren van market impact en het verkrijgen van een betere prijs. Een belangrijk concept hier is de Volume-Weighted Average Price (VWAP). Grote partijen gebruiken VWAP als een benchmark om grote orders gedurende de dag in kleinere stukjes uit te voeren, met als doel een gemiddelde uitvoeringsprijs te realiseren die dicht bij de VWAP van die dag ligt.²⁴ Een

ExecutionAlgorithm plugin die een VWAP-strategie implementeert, zou een TradePlan niet omzetten in één enkele order, maar in een reeks van getimed, kleinere orders.

Risicobeheer op Portefeulleniveau vs. Handelsniveau: De critical_event_detector fase, met de max_drawdown_overlay plugin, is een goede eerste stap, maar functioneert als een reactieve, systeem-brede "noodstop".¹ Professioneel risicobeheer is proactief en portfolio-centrisch, en gaat veel verder dan een stop-loss op een individuele transactie. De volgende concepten ontbreken in de huidige opzet:

- **Correlatie-Analyse:** Het diversifiëren over meerdere strategieën of activa is alleen effectief als hun rendementen niet sterk met elkaar gecorreleerd zijn. Een professioneel systeem moet de correlatiematrix tussen actieve strategieën kunnen analyseren en de blootstelling aanpassen als strategieën te veel in tandem bewegen, wat het risico verhoogt.²⁸
- **Dynamische Position Sizing:** De huidige SizePlanner gebruikt waarschijnlijk een vast risicopcentage. Geavanceerde methoden zoals het **Kelly Criterion** of **volatiliteit-geschaalde position sizing** passen de grootte van de transactie dynamisch aan op basis van de historische winstkans van de strategie en de huidige marktvolatiliteit. In volatiele markten worden posities verkleind om het risico constant te houden.²⁸
- **Portfolio Heat / Drawdown Controls:** In plaats van een harde noodstop, zou een geavanceerd systeem de positiegroottes van alle actieve strategieën dynamisch en proportioneel verkleinen naarmate de totale portefeuille een vooraf ingestelde maximale drawdown-limiet nadert. Dit wordt ook wel het beheren van de "portfolio heat" genoemd.²⁹

Sectie 4: Strategische Aanbevelingen voor een Institutioneel-Grade Evolutie

4.1. Evolutie van de Data-Laag voor Multi-Granulariteitsanalyse

Om de geïdentificeerde blinde vlekken met betrekking tot markt-microstructuur aan te pakken, is een fundamentele evolutie van de data-laag noodzakelijk. De aanbeveling is om de architectuur te ontwikkelen naar een dual-granulariteitsmodel dat zowel geaggregeerde OHLCV-staven als ruwe tick-data gelijktijdig kan opnemen, opslaan en verwerken. Dit stelt het systeem in staat om de bestaande, op prijsactie gebaseerde plugins te blijven ondersteunen, terwijl het de deur opent voor geavanceerde orderstroom-analyse.

Voor de live data-omgeving wordt een architectonisch patroon voorgesteld dat is geïnspireerd op de **Kappa Architectuur**, een model dat is ontworpen voor het verwerken van real-time datastromen.³¹ In deze opzet zou een onveranderlijk logboek (een

immutable log), zoals Apache Kafka, fungeren als de centrale ruggengraat voor data-inname. Alle ruwe tick-data van de beurs wordt naar dit logboek geschreven. Vanuit dit ene logboek kunnen meerdere, onafhankelijke stream-processoren parallel opereren:

1. **Een Orderstroom-Processor:** Deze processor consumeert de tick-stroom in real-time om on-the-fly orderstroom-metrieken te berekenen, zoals volume delta, en om footprint-data te genereren.
2. **Een OHLCV-Aggregator:** Een tweede processor consumeert dezelfde tick-stroom en aggregeert deze live naar de benodigde OHLCV-staven (bv. 15-minuten, 1-uur). Deze geaggregeerde data kan vervolgens worden gebruikt door de bestaande plugin-pijplijn.

Dit Kappa-geïnspireerde model heeft als voordeel dat het beide data-granulariteiten laat co-existeren op basis van één enkele, betrouwbare bron van waarheid (het tick-log). Het elimineert de noodzaak voor aparte batch- en real-time verwerkingslagen en biedt een schaalbare en fouttolerante oplossing.

Voor de backtesting-omgeving zou de DataLoader moeten worden uitgebreid om naast CSV-bestanden met OHLCV-data ook bestanden met historische tick-data te kunnen lezen.³⁶ Een pre-processing stap tijdens het opstarten van een backtest zou dan de benodigde OHLCV-staven kunnen genereren uit de tick-data. Vervolgens zouden beide datasets (de ruwe ticks en de geaggregeerde staven) beschikbaar worden gesteld aan de

StrategyOperator, zodat plugins van beide typen analyse kunnen uitvoeren.

4.2. Uitbreiding van het Plugin-Ecosysteem voor Diepere Analyse

De "Plugin First" architectuur is de grootste kracht van het systeem en moet worden benut om

de nieuwe analytische mogelijkheden te integreren. Dit vereist de introductie van nieuwe plugin-types en -interfaces die zijn ontworpen om met de nieuwe databronnen en concepten te werken. De volgende tabel schetst een blauwdruk voor de uitbreiding van het plugin-ecosysteem.

Tabel 2: Voorgestelde Nieuwe Plugin-Types en Interfaces

Nieuw Plugin Type	Doel	Positie in Workflow	Input	Output
TickContextWorker	Verwerkt ruwe tick-data om orderstroom-metrieken te genereren.	Pre-Fase 1 (Data Voorbereiding)	Ruwe Tick Data	Een DataFrame met extra kolommen zoals delta, bid_volume, ask_volume, imbalance.
ProfileContextWorker	Genereert Volume Profile of Market Profile data voor de gehele dataset of een rollend venster.	Fase 2 (Structurele Context)	OHLCV + Volume Data	Een Profile-object (met POC, VA, etc.) dat wordt geregistreerd in de TradingContext.
MultiTimeframeProvider	Een service (geen plugin) die wordt geïnjecteerd in de TradingContext.	Beschikbaar in alle fasen	N/A	Een dictionary van DataFrames, bv. {'1H': df_1h, '1D': df_1d}, met ingebouwde bescherming tegen look-ahead bias.
ExecutionAlg	Vervangt de	Fase 8	TradePlan DTO	Een stroom

orithm	simpele OrderRouter. Implementeert complexe executielogica.	(Executie)		van meerdere, kleinere "child orders" over een bepaalde tijdsperiode, met als doel een benchmark (bv. VWAP) te benaderen.
PortfolioRisk Model	Een plugin op portefeulleniveau voor dynamische aanpassingen.	Fase 9 (Portfolio Overlay)	De volledige Portfolio-staat, een lijst van voorgestelde TradePlans.	Een lijst van <i>aangepaste</i> TradePlans (bv. met gereduceerde omvang) op basis van correlatie of drawdown.

De introductie van een MultiTimeframeProvider als een centrale service die in de TradingContext wordt geplaatst, is de meest robuuste oplossing voor MTFA. In plaats van dat elke plugin zelf data moet laden, biedt de provider een gecentraliseerde en gecontroleerde toegang tot de verschillende tijdlijnen, waarbij de logica om look-ahead bias te voorkomen centraal wordt beheerd.²³ Een plugin in de 1-uurs workflow kan dan bijvoorbeeld

context.mtf_provider.get_last_closed_bar('1D') aanroepen om veilig de data van de vorige, voltooide dagelijkse kaars op te halen.

4.3. Architectuur voor een Volwaardig Risicobeheersysteem op Portefeulleniveau

Om de blinde vlek in risicobeheer aan te pakken, is een herdefiniëring van de rol van het Portfolio-object en de laatste fase van de workflow noodzakelijk. De aanbeveling is om deze componenten te transformeren van een reactieve alarmfunctie naar een proactief, optimaliserend risicobeheersingshub.

Ontwerpvoorstel:

1. **Uitbreiding van het Portfolio-object:** Het Portfolio-object moet worden uitgebreid om niet alleen zijn huidige staat bij te houden, maar ook een historische reeks van zijn eigen equity-curve en belangrijke prestatiemetrieken (zoals Sharpe ratio, maximale drawdown). Deze historische data is de input die nodig is voor dynamische berekeningen.
2. **Evolutie van de StrategyOperator:** De StrategyOperator moet in staat worden gesteld om een "portfolio-simulatie" uit te voeren, waarbij het meerdere, potentieel gecorreleerde strategieconfiguraties gelijktijdig orkestreert binnen één Portfolio-object. Dit is de technische realisatie van het PortfolioSupervisor-concept.¹
3. **Herdefiniëring van Fase 9:** De CRITICAL_EVENT_DETECTOR-fase moet worden hernoemd en geherpositioneerd naar **PORTFOLIO_OPTIMIZER**. De plugins in deze fase moeten de autoriteit krijgen om de lijst van uitgaande TradePlans te *wijzigen*, niet alleen om alarmen te genereren. Dit maakt proactief ingrijpen mogelijk:
 - Een CorrelationMatrix plugin in deze fase zou de voorgestelde nieuwe transactie kunnen analyseren in de context van de reeds openstaande posities. Als de nieuwe transactie sterk positief gecorreleerd is met een bestaande positie, kan de plugin de omvang van de nieuwe transactie reduceren om het geclusterde risico te beperken.²⁸
 - Een DynamicSizer plugin zou de recente prestaties van de portefeuille (bv. de huidige drawdown) kunnen gebruiken als input. Als de portefeuille zich in een drawdown-periode bevindt, kan de plugin de risico-allocatie voor *alle* nieuwe transacties tijdelijk verlagen om kapitaal te beschermen.²⁹

Door deze aanpassingen te implementeren, evolueert de laatste fase van de workflow van een simpele "noodrem" naar een intelligent "gaspedaal en stuur", dat de algehele blootstelling van de portefeuille dynamisch beheert op basis van prestaties en marktomstandigheden. Dit is de essentie van geavanceerd, kwantitatief risicomanagement.

Geciteerd werk

1. project_context.txt
2. Advanced Candlestick Patterns - CFI Trading, geopend op oktober 2, 2025, <https://cfi.trade/en/jo/educational-articles/what-is-technical-analysis/advanced-candlestick-patterns>
3. Candlestick Pattern Trading Strategies | TrendSpider Learning Center, geopend op oktober 2, 2025, <https://trendspider.com/learning-center/candlestick-pattern-trading-strategies/>
4. 16 Candlestick Patterns Every Trader Should Know | IG International, geopend op oktober 2, 2025, <https://www.ig.com/en/trading-strategies/16-candlestick-patterns-every-trader-should-know-180615>
5. Candlestick Patterns - Definition, How They Work, Examples - Corporate Finance Institute, geopend op oktober 2, 2025, <https://corporatefinanceinstitute.com/resources/equities/candlestick-patterns/>

6. Advanced Block: Candlestick Tactics Explained - LuxAlgo, geopend op oktober 2, 2025,
<https://www.luxalgo.com/blog/advanced-block-candlestick-tactics-explained/>
7. Synergizing Quantitative Finance Models and Market Microstructure Analysis for Enhanced Algorithmic Trading Strategies | Request PDF - ResearchGate, geopend op oktober 2, 2025,
https://www.researchgate.net/publication/381810459_Synergizing_Quantitative_Finance_Models_and_Market_Microstructure_Analysis_for_Enhanced_Algorithmic_Trading_Strategies
8. High Frequency Trading I: Introduction to Market Microstructure - QuantStart, geopend op oktober 2, 2025,
<https://www.quantstart.com/articles/high-frequency-trading-i-introduction-to-market-microstructure/>
9. Financial Market Microstructure and Trading Algorithms - CBS Research Portal, geopend op oktober 2, 2025,
https://research.cbs.dk/files/58451772/jens_valloe_christiansen.pdf
10. Order flow trading - Wikipedia, geopend op oktober 2, 2025,
https://en.wikipedia.org/wiki/Order_flow_trading
11. Order Flow Trading: Definition and Strategies - XS, geopend op oktober 2, 2025,
<https://www.xs.com/en/blog/order-flow-trading/>
12. Top 4 Order Flow Trading Strategies for Profit - Black Eagle Financial Group, geopend op oktober 2, 2025,
<https://blackeaglefg.com/4-order-flow-trading-strategies-to-increase-profit/>
13. Order Flow Analysis: A Complete Guide to Reading Market Dynamics Like a Pro, geopend op oktober 2, 2025, <https://tradefundrr.com/order-flow-analysis/>
14. Volume Profile & Order Flow: Tools for Deep Market Insight - Bookmap, geopend op oktober 2, 2025,
<https://bookmap.com/blog/volume-profile-order-flow-tools-for-deep-market-insight>
15. Market Profile Trading Strategies : Understanding its Power and Impact - Bookmap, geopend op oktober 2, 2025,
<https://bookmap.com/blog/market-profile-trading-understanding-its-power-and-impact>
16. Market Profile vs. Volume Profile, Which Tool Is More Effective for Traders? - stp trading, geopend op oktober 2, 2025,
<https://www.stptrading.io/blog/market-profile-vs-volume-profile/>
17. Anticipating Market Action Using Market Profile And Volume Analytics Strategies, geopend op oktober 2, 2025,
<https://www.wyckoffanalytics.com/demand/market-profile-and-volume-analytics-strategies/>
18. Volume Profile Trading Guide: Master Market Analysis Today - TradeFundrr, geopend op oktober 2, 2025,
<https://tradefundrr.com/volume-profile-trading-guide/>
19. Order Flow Chart - #Footprint | Indicator for MetaTrader 4/5 | ClusterDelta, geopend op oktober 2, 2025, <https://clusterdelta.com/footprint>

20. Multiple Time Frame Analysis Explained | Bitsgap blog, geopend op oktober 2, 2025,
<https://bitsgap.com/blog/using-multiple-time-frames-advantages-examples>
21. Multi-Timeframe Trading Strategy | Backtest Insights - QuantifiedStrategies.com, geopend op oktober 2, 2025,
<https://www.quantifiedstrategies.com/multiple-timeframe-strategy/>
22. How To Perform A Multi TimeFrame Analysis + 5 Strategies - Tradeciety, geopend op oktober 2, 2025,
<https://tradeciety.com/how-to-perform-a-multiple-time-frame-analysis>
23. How to use multiple timeframes in your algotrading strategy - Jesse Blog, geopend op oktober 2, 2025,
<https://jesse.trade/blog/tutorials/how-to-use-multiple-timeframes-in-your-algotrading-strategy>
24. Volume Based Automated Trading Strategies - Blog - TradersPost, geopend op oktober 2, 2025,
<https://blog.traderspost.io/article/volume-analysis-trading-systems>
25. Volume-Weighted Average Price (VWAP): Definition and Calculation - Investopedia, geopend op oktober 2, 2025,
<https://www.investopedia.com/terms/v/vwap.asp>
26. Competitive Algorithms for VWAP and Limit Order Trading - UPenn CIS, geopend op oktober 2, 2025, <https://www.cis.upenn.edu/~mkearns/papers/vwap.pdf>
27. VWAP Trading Strategy - Empirica, geopend op oktober 2, 2025,
<https://empirica.io/blog/vwap-algorithm/>
28. Essential Risk Management Strategies for Algo Trading - Algocrab, geopend op oktober 2, 2025,
<https://www.algocrab.com/blog/3-risk-management-algo-trading.html>
29. 7 Risk Management Strategies For Algorithmic Trading | Nurp, geopend op oktober 2, 2025,
<https://nurp.com/wisdom/7-risk-management-strategies-for-algorithmic-trading/>
30. Risk Management Strategies for Algo Trading - LuxAlgo, geopend op oktober 2, 2025,
<https://www.luxalgo.com/blog/risk-management-strategies-for-algo-trading/>
31. What is Kappa Architecture? | Dremio, geopend op oktober 2, 2025,
<https://www.dremio.com/wiki/kappa-architecture/>
32. Kappa Architecture - System Design - GeeksforGeeks, geopend op oktober 2, 2025,
<https://www.geeksforgeeks.org/system-design/kappa-architecture-system-design/>
33. Kappa Architecture 101—Deep Dive into Stream-First Design - Chaos Genius, geopend op oktober 2, 2025,
<https://www.chaosgenius.io/blog/kappa-architecture/>
34. Kappa Architecture - Hazelcast, geopend op oktober 2, 2025,
<https://hazelcast.com/foundations/software-architecture/kappa-architecture/>
35. Kappa Architecture: An Efficient Model for Real-Time Processing | by Lenon Rodrigues, geopend op oktober 2, 2025,

<https://medium.com/@lenonrodrigues/kappa-architecture-an-efficient-model-for-real-time-processing-767c623d04ad>

36. TigerData Documentation | Analyze financial tick data with TimescaleDB - Docs, geopend op oktober 2, 2025,
<https://docs.tigerdata.com/tutorials/latest/financial-tick-data/>
37. Pandas OHLC: Convert tick by tick data to OHLC data - QuantInsti Blog, geopend op oktober 2, 2025,
<https://blog.quantinsti.com/tick-tick-ohlc-data-pandas-tutorial/>
38. Multiple Time Frames: How to Use Them In Your Trading - Investopedia, geopend op oktober 2, 2025,
<https://www.investopedia.com/articles/trading/07/timeframes.asp>