

# Viserion V1 Technical Description

## Table of Contents

Viserion V1 Technical Description.....	1
Technical Market Analysis.....	2
Assumption 1:.....	2
Deflationary Assets are bullish markets.....	2
Assumption 1.1:.....	3
Trading Call Options with Viserion Martingale on Growing Markets is worth it.....	3
Recovering losses.....	4
Assumption 2.0:.....	4
Recovering losses suggests profitably before the first scale reaches it's peak.....	4
Assumption 2.1:.....	4
Shutting down the system at a favorable scalestate.....	4
Technical execution.....	5
Assumption 3.0:.....	5
Finding an Exchange with sufficient payout and reasonable fees, and a robust api.....	5
Assumption 3.1:.....	5
Running the Trading system on a reliable Internet connection.....	5

# Introduction

At the beginning of 2022 I **discovered a risk management system** that would allow trading systems to recover over losses, if the win-rate is above 50%, or there is no losing streak that is long enough to **exhaust the available balance**.

An more effective risk score can be archived in **using multiple independent scales** instead of one single scale.

## Terminology:

Term	Description
Costs (C)	Required capital to operate a V1 system
Peak (P)	Maximum capacity for one system before failing
Scale (S)	One RiskManagement column that works independent of other scales in a V1 system
PeakSingle (PS)	Peak on one single Scale explicitly
ScaleCount (SC)	Number of Scales in a system

## Cost Calculation:

### Single Martingale Scale:

$$C = ( 2^{(P + 1)} - 1 )$$

### Multiple Martingale Scales:

$$C = ( 2^{(PS + 1)} - 1 ) * SC$$

# Technical Market Analysis

## Assumption 1:

### Deflationary Assets are bullish markets.

When analysing candlestick patterns of the gold market from 2010 until 2024 a clear bullish trend shows in outweighing the number of days wher the gold price went down with the days where the gold market had a rising price.

Year	Bullish Candles	Bearish Candles	Bullish Candles %	Bearish Candles %	Max Consecutive Bearish Candles	Total Candles
2010	143	109	56.74	43.25	7	252.0
2011	142	110	56.34	43.65	6	252.0
2012	126	126	50	50	7	252.0
2013	112	140	44.44	55.55	8	252.0
2014	124	128	49.2	50.79	6	252.0
2015	107	145	42.46	57.53	15	252.0
2016	127	125	50.39	49.60	8	252.0
2017	143	112	56.07	43.92	7	255.0
2018	125	133	48.44	51.55	7	258.0
2019	129	129	50	50	9	258.0
2020	151	108	58.30	41.69	7	259.0
2021	128	130	49.61	50.38	9	258.0
2022	135	123	52.32	47.67	8	258.0
2023	133	124	51.75	48.24	9	257.0
2024	145	112	56.42	43.57	6	257.0

## Assumption 1.1:

**Trading Call Options with Viserion Martingale on Growing Markets is worth it.**

When simply Buying a Put Option at 1\$ every day, estimated profits are for 2010: \$34

> 143 - 109

34

With single applied Martingale profits: \$143

Required capital(for 8 levels): \$511

>  $2^{(8+1)} - 1$

511

1
2
4
8
16
32
64
128
256

With 2 scales Martingale profits: \$143

Required capital(for 2\*4 levels): \$62

>  $(2^{(4+1)} - 1) * 2$

62

1	1
2	2
4	4
8	9
16	16

## Recovering losses

### Assumption 2.0:

**Recovering losses suggests profitably before the first scale reaches it's peak.**

Backtest 4 Years, 2020 – 2024, Gold Price Hourly: **TODO NOT DONE YET**

Won Trades	501 orso
Scalestates:	0, 0, 0, 10, 7, 4, 5, 5, 0, 0 orso
Unfinished Profit	Sum of all capital in the nodes not at idle or so
Balance	Account balance orso

### Assumption 2.1:

#### **Shuting down the system at a faverable scalestate.**

When the system is stopped the risk of reaching the peak increases as profitable scales are activated. Which could lead the last running scale to peak. When shuting down a system starting to deactivate unprofitable scales is recommended, at **3\*scale count** before the wished trade end. Research to test the value 3 is not done yet.

# Technical execution

## Assumption 3.0:

**Finding an Exchange with sufficient payout and reasonable fees, and a robust api.**

## Assumption 3.1:

**Running the Trading system on a reliable Internet connection.**

The consequence of losing connectivity to the api is the risk of not continuing to make profits, there is no time correlated risk of losing capital when the system is not active.

All relevant configuration is saved in a statefile.json, as the History of the program is saved in a history.csv file.

Every iteration is executed in the following order:

- Fetching Price-data
  - A Request to the broker is initialized to check connectivity
  - Market-data is collected to calculate prices for the next trade
- Node indexes are updated
  - The current node-index becomes the last node-index if more then 24 hours have passed and there is no open position
  - Current node-index is read from to file to calculate risk for the next position
  - Last node-index is read to determine the last outcome and the new scale-state
- New position is opened
  - A Request with the order-information is sent to the broker
  - Required data is saved in the statefile for the current node-index to determine a win or loss at the next iteration
- Write outcome to history
  - If all the processes are successful, the result is written to the historyfile
  - If some steps fail, they are written to a log-file

The Server can be executed every 10 Minutes if the errorhandling is well buildt, the statefile would alow to read the last tradedate back, to check if the next position should be opened. This setup can be archived with a cron task. When the entire application is buildt with python the react dashboard can analyse the statefile, and or the historyfile.