

Introducing Prediction Feeds

Trent McConaghy
Ocean Protocol





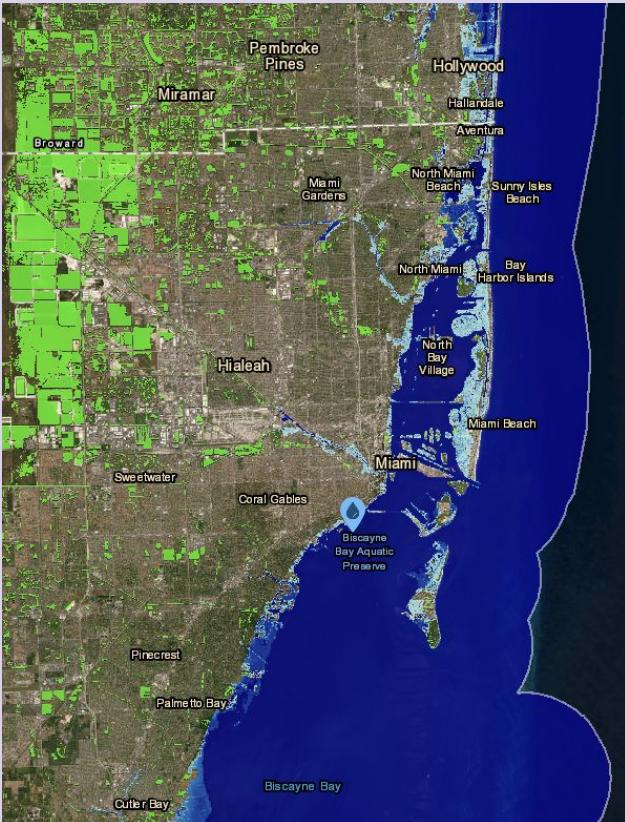


Miami vs sea level rise [NOAA]

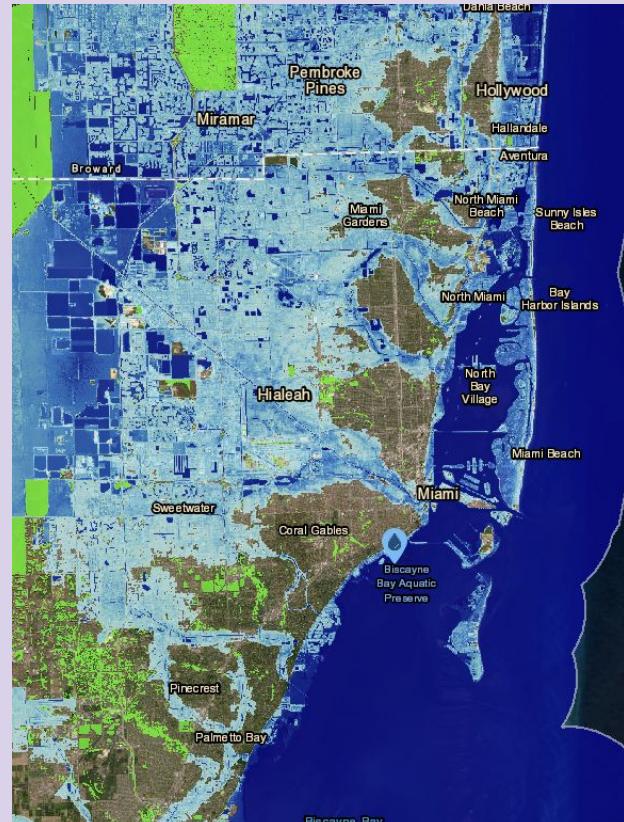
Today



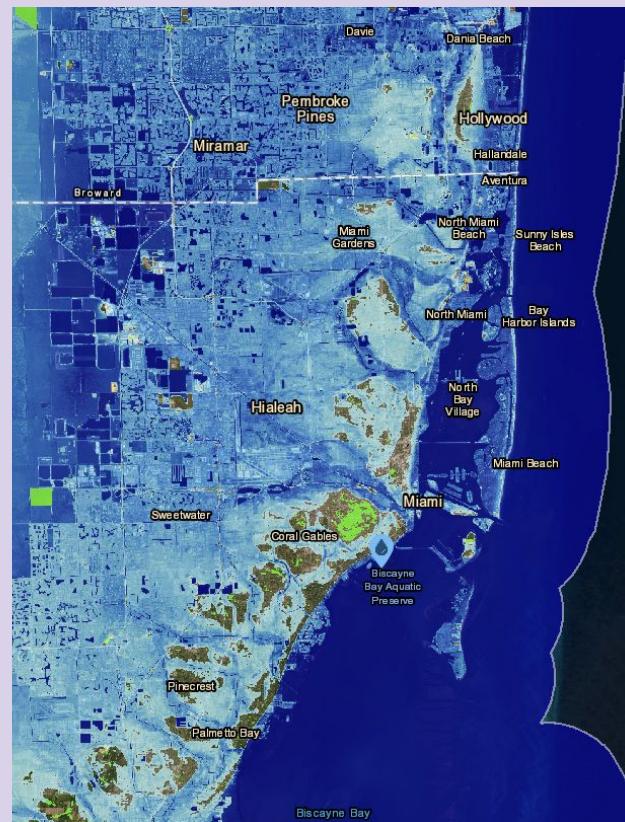
0.9m



2.1m

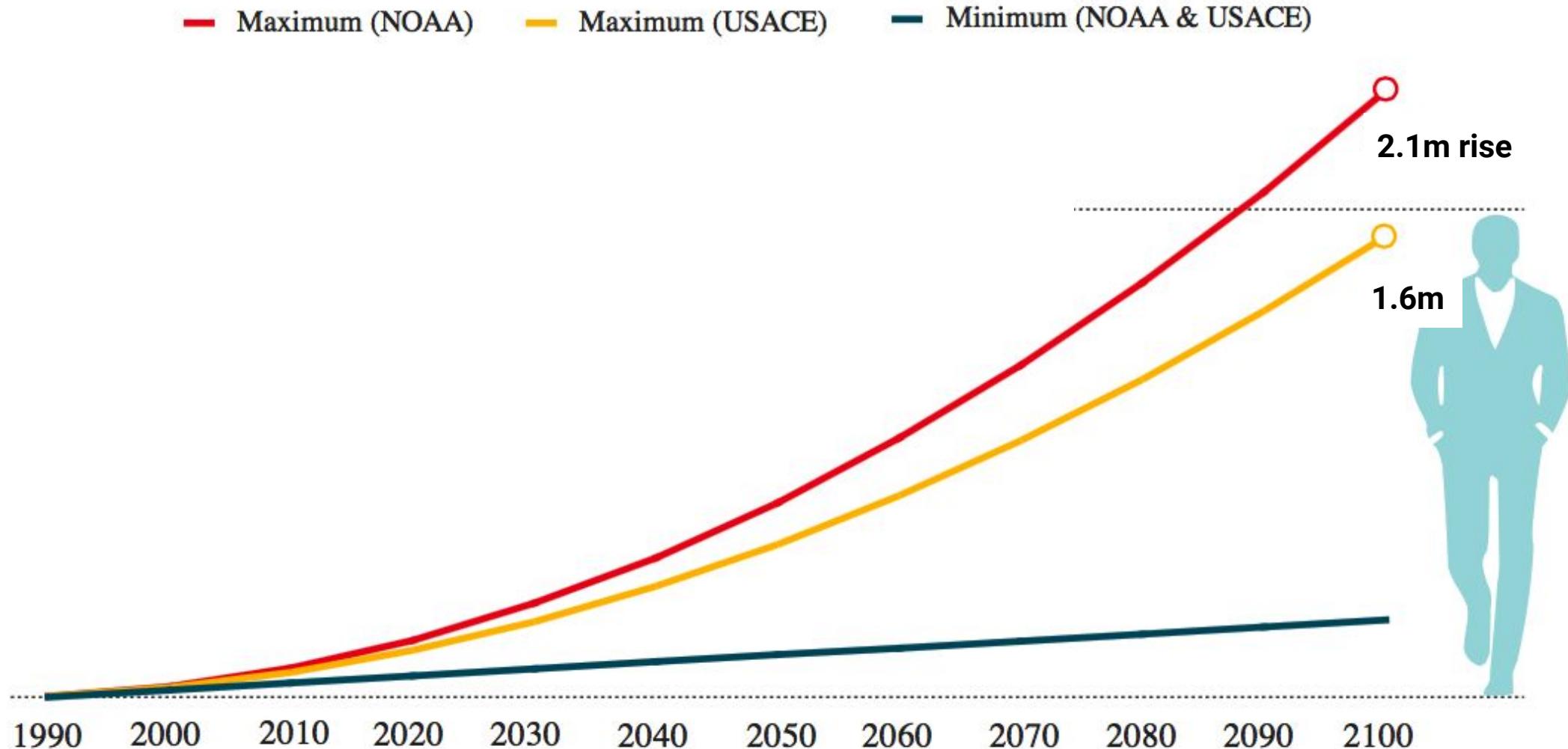


3m

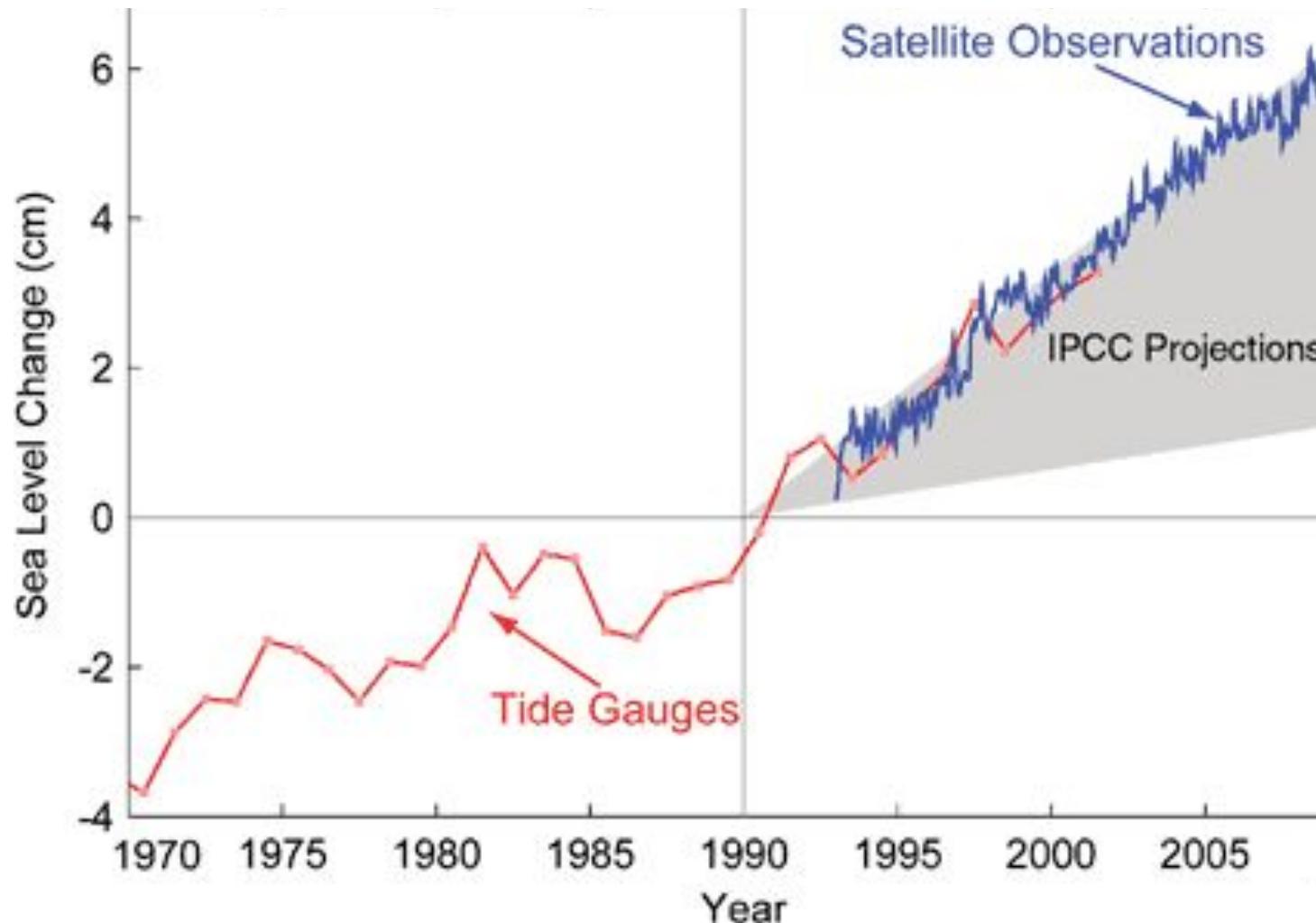


Miami sea level prediction feeds

It *will* be underwater 😱; only question is *when*



Previous sea level prediction feeds were too optimistic

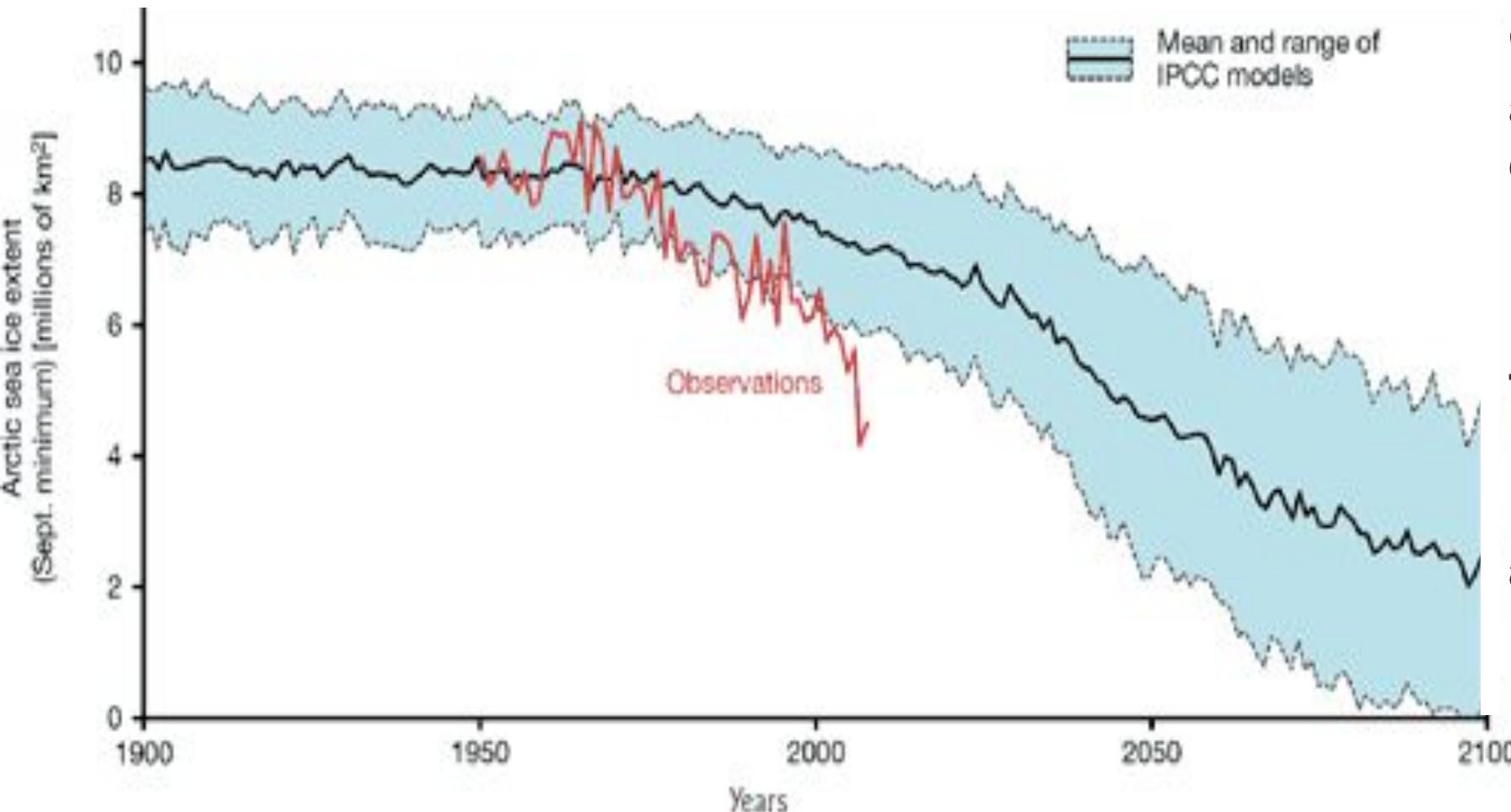


"underestimated the rate of sea level rise because the rapid melting of the **ice sheets** and **glaciers** was not incorporated"

There's a lot on the line!

How do we get more accurate predictions?

Previous sea ice prediction feeds were too optimistic



"Melting of Arctic sea-ice has accelerated far beyond the expectations of climate models ...40% greater..."

There's a lot on the line!

How do we get more accurate predictions?

There's great value for accurate prediction feeds

In climate, real estate, agriculture, logistics, crypto markets



There's a lot on the line!

How do we get more accurate predictions?

The background of the image is a vibrant, abstract composition dominated by shades of purple and blue. It features several bright, glowing streaks that curve and flow across the frame, creating a sense of motion and depth. These streaks are particularly prominent in the upper left and lower right areas. In the lower left foreground, there's a partial view of a car's front end, showing its headlight and a bit of the grille, which also reflects the surrounding light.

Towards accurate prediction feeds

Accurate predictions are hard. No accountability when wrong





**For accurate prediction feeds, make predictions accountable
Correct → \$\$, wrong → lose \$\$**

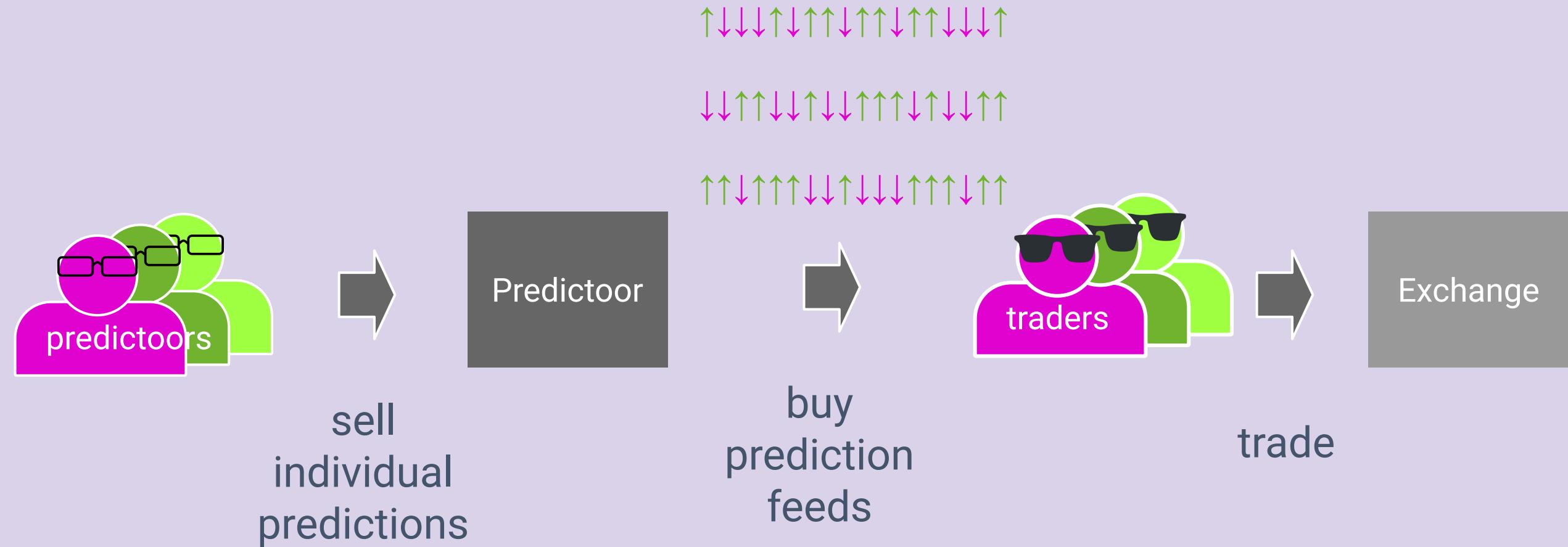
The goal

- 10,000+ prediction feeds
- globally-distributed
- anyone can predict and earn \$
- accountable --> accuracy
- non-custodial
- privacy-preserving

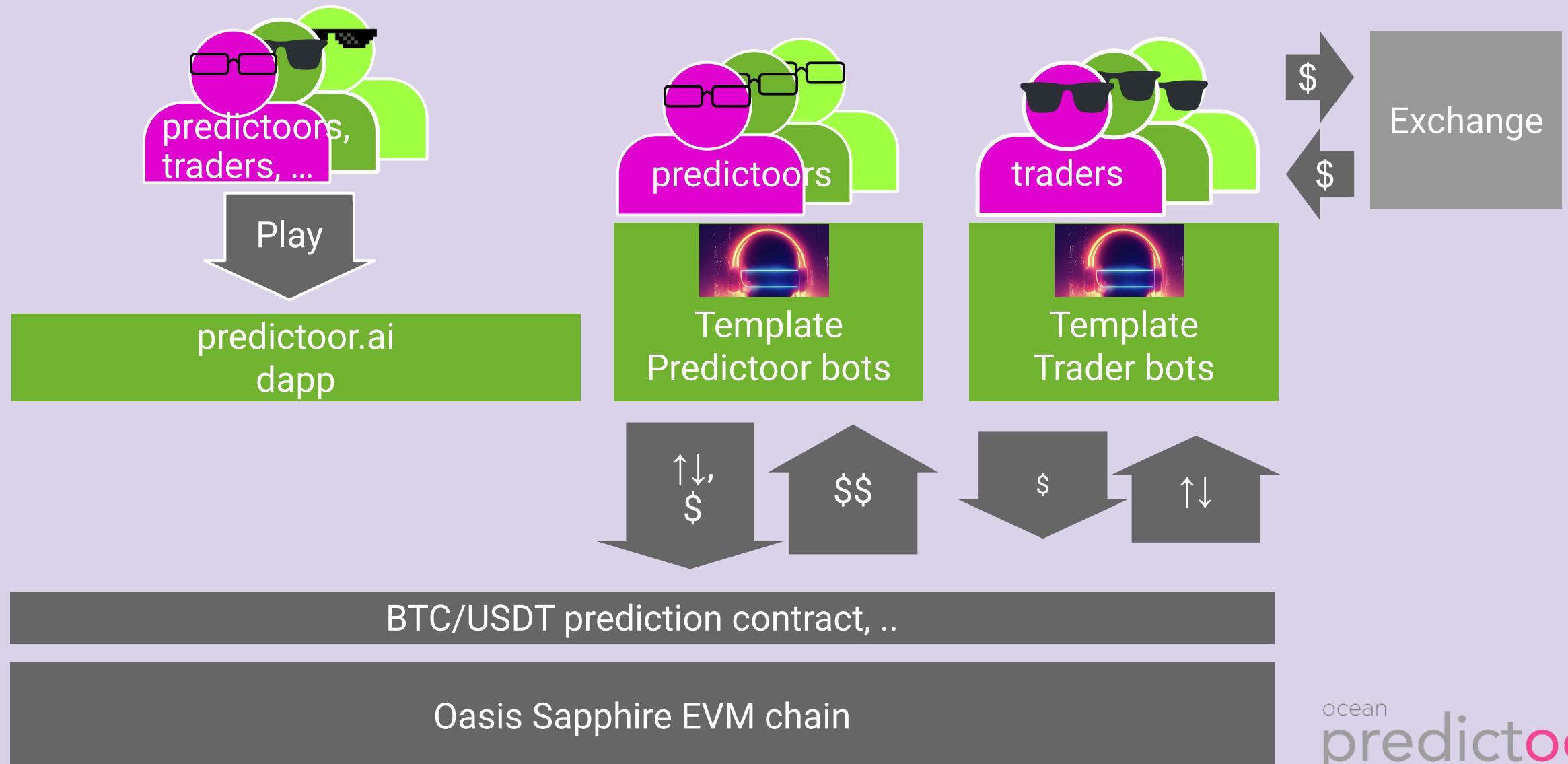


ocean
predictoor

Predictoor Overview



Predictoor Structure



predictoor.ai & testnet is live

ocean
predictoor

Oasis Sapphire Testnet [0xf25B...54C2](#) [ipy SDK](#) [About](#)

Asset (5m)	11:15	11:20	11:23	11:25 Predictions	Accuracy 24h	Subscription
BTC-TUSD BINANCE	\$26250.91 Pred ↑ 0.00%	\$26237.44 Pred ↓ 0.00%	\$26247.81	+ 1 stated Pred ?	49%	3 / 24h FREE
DOGE-USDT BINANCE	\$0.06361 Pred ↑ 0.00%	\$0.06356 Pred ↓ 0.00%	\$0.06358 Pred ↓ 0.00%	+ 0.01 stated Pred ↓ 0.00%	51.8%	3 / 24h Th 11min left
BTC-USDT BINANCE	\$26213.62 Pred ↓ 0.00%	\$26198.19 Pred ↓ 0.00%	\$26213.51 Pred ↓ 0.00%	-	49.8%	3 / 24h BUY
ETH-USDT BINANCE	\$1645.09 Pred ↓ 0.00%	\$1644.29 Pred ↓ 0.00%	\$1644.87 Pred ↓ 0.00%	-	42.2%	3 / 24h BUY
ADA-USDT BINANCE	\$0.2578 Pred ↓ 0.00%	\$0.2575 Pred ↓ 0.00%	\$0.2576 Pred ↓ 0.00%	-	55.3%	3 / 24h BUY
BNB-USDT BINANCE	\$216.9 Pred ↓ 0.00%	\$216.7 Pred ↓ 0.00%	\$216.8 Pred ↓ 0.00%	-	50.5%	3 / 24h BUY
TRX-USDT BINANCE	\$0.07902 Pred ↓ 0.00%	\$0.07899 Pred ↓ 0.00%	\$0.07897 Pred ↓ 0.00%	-	50%	3 / 24h BUY
SOL-USDT BINANCE	\$19.77 Pred ↓ 0.00%	\$19.76 Pred ↓ 0.00%	\$19.79 Pred ↓ 0.00%	-	53%	3 / 24h BUY
XRP-USDT BINANCE	\$0.5037 Pred ↓ 0.00%	\$0.5036 Pred ↓ 0.00%	\$0.5039 Pred ↓ 0.00%	-	48.4%	3 / 24h BUY
DOT-USDT BINANCE	\$4.284 Pred ↓ 0.00%	\$4.274 Pred ↓ 0.00%	\$4.278 Pred ↓ 0.00%	-	49.6%	3 / 24h BUY
LTC-USDT BINANCE	\$63.24 Pred ↓ 0.00%	\$63.21 Pred ↓ 0.00%	\$63.27 Pred ↓ 0.00%	-	49.7%	3 / 24h BUY

BALANCE = 1,958

Prediction feeds for the top 10 tokens (non-stablecoins)

For USDT pairs on Binance

5m ahead predictions

Predictoor bot templates



Predictoor

main → pdr-backend / pdr_backend / predictoor / approach1 / predictoor_agent1.py

trizin Reduce random stake amount (#134) · 5 days ago · History

Code Blame 149 lines (120 loc) · 5.08 KB

```
1 import sys
2 import random
3 import time
4 from typing import Dict, List, Tuple
5
6 from enforce_typing import enforce_types
7
8 from pdr_backend.models.feed import Feed
9 from pdr_backend.predictoor.approach1.predictoor_config1 import PredictoorConfig1
10
11
12 @enforce_types
13 class PredictoorAgent1:
14     """
15     What it does
16     - Fetches Predictoor contracts from subgraph, and filters them
17     - Monitors each contract for epoch changes.
18     - When a value can be predicted, call predict.py::predict_function()
19     """
20
21
22     def __init__(self, config: PredictoorConfig1):
23         self.config = config
24
25         self.feeds: Dict[str, Feed] = self.config.get_feeds() # [addr] : Feed
26
27         if not self.feeds:
28             print("No feeds found. Exiting")
29             sys.exit()
```



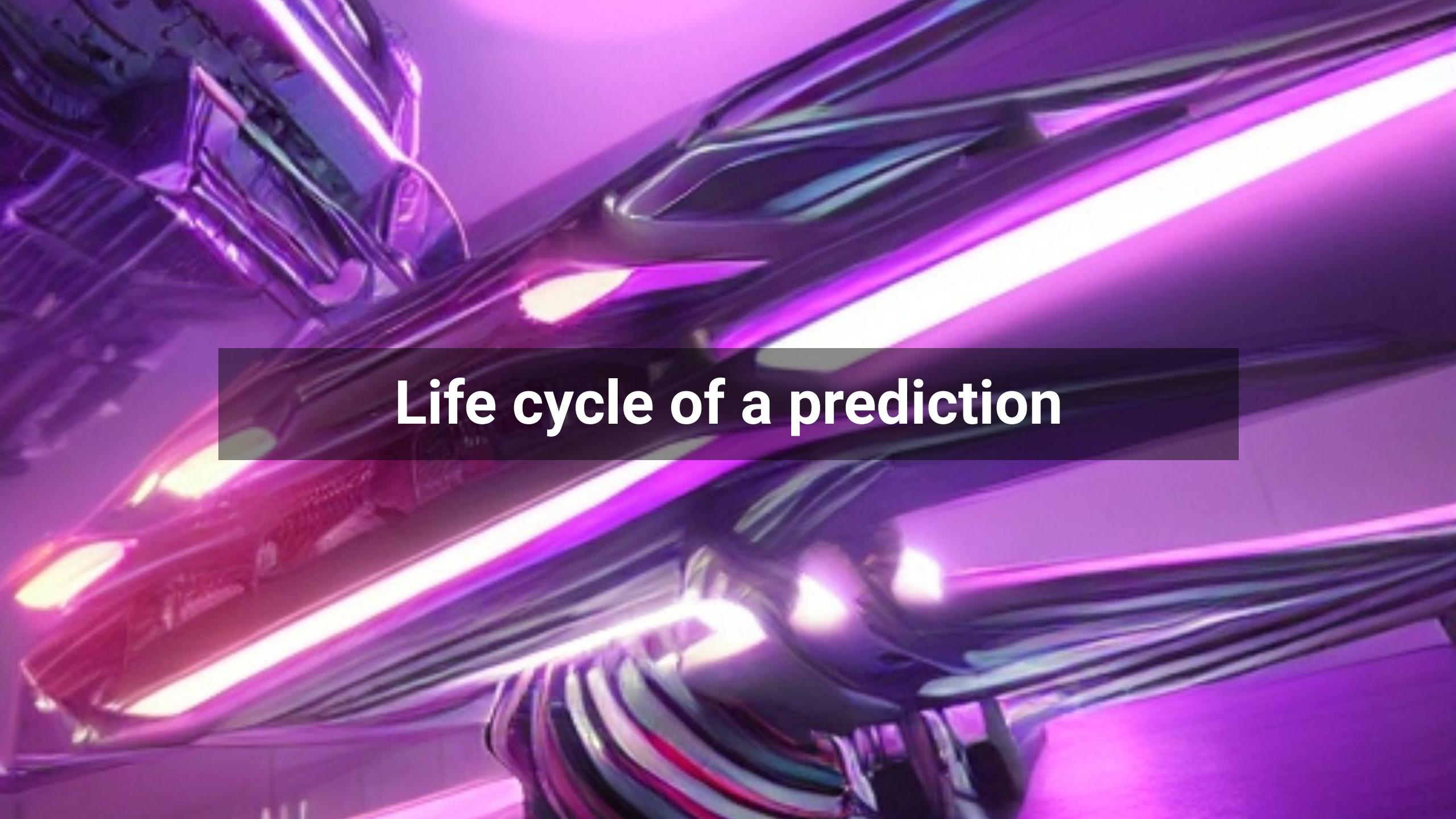
Trader

main → pdr-backend / pdr_backend / trader / trader_agent.py

trizin #157 - [trader] Move max tries to trader agent config class (#159) · yesterday · History

Code Blame 176 lines (145 loc) · 5.99 KB

```
1 import sys
2 import time
3 from typing import Any, Callable, Dict, List, Optional, Tuple
4
5 from enforce_typing import enforce_types
6
7 from pdr_backend.models.feed import Feed
8 from pdr_backend.trader.trader_config import TraderConfig
9 from pdr_backend.util.cache import Cache
10
11
12 # pylint: disable=too-many-instance-attributes
13 class TraderAgent:
14     def __init__(
15         self,
16         trader_config: TraderConfig,
17         _do_trade: Optional[Callable[[Feed, Tuple], Any]] = None,
18         cache_dir=".cache",
19     ):
20         self.config = trader_config
21         self._do_trade = _do_trade if _do_trade else do_trade
22
23         self.feeds: Dict[str, Feed] = self.config.get_feeds() # [addr] : Feed
24
25         if not self.feeds:
26             print("No feeds found. Exiting")
27             sys.exit()
```

The background of the slide features a dynamic, abstract design composed of thick, glowing streaks in shades of purple, blue, and white. These streaks curve and twist across the frame, creating a sense of motion and depth. The overall aesthetic is futuristic and high-energy.

Life cycle of a prediction

epoch t

epoch t+1

epoch t+2

Exchange



Buy / sell



traders



Buy feed, see
↑↓ pred'n & stake



Sell / buy



traders



Get trueval
"↑"



trueval
agent



Submit "↑"

Contract

Store ↑↓

Σ

Distribute \$

Predict "↑"
(and stake)



predictoors



predictoors



Predictoor & privacy

Predictoor & privacy: the challenge

Predictoor needs privacy for:

1. Submitted predictions
2. Compute aggregate predictions; and
3. Aggregated predictions - only subscribers can see

(While remaining decentralized, non-custodial, etc.)



Ocean Predictoor uses Oasis Sapphire for Privacy

We researched & prototyped many privacy technologies.

We chose Oasis Sapphire:

- It could handle the privacy needs cleanly end-to-end
- It's the only privacy-preserving EVM chain in production (!)
- Uses Intel SGX as Trusted Execution Environment (TEE)



Sapphire: Confidential & EVM Compatible



Privacy

- Contract state confidentiality
- Flexibility - public or private

Strong security/integrity guarantees

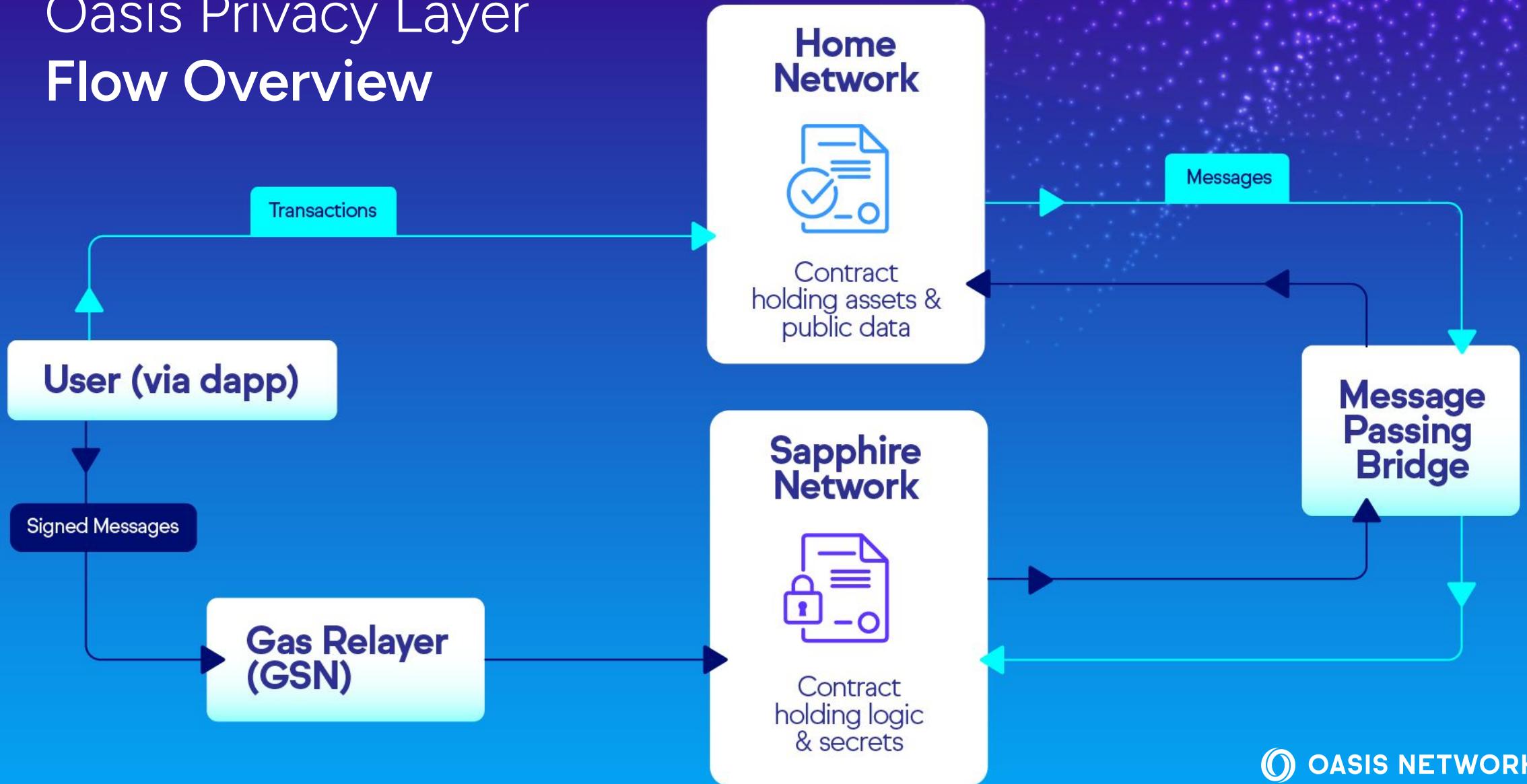
- Need to break secure enclave
- Separation of key management and compute nodes

Full EVM compatibility

- Easy to use with familiar tooling
- User-facing tools: Web3 gateway, wallet, block explorer
- Dev tools: Solidity/Vyper, Hardhat/Truffle, etc.

New built-in functions: RNG, signing, encryption

Oasis Privacy Layer Flow Overview





Use cases for on-chain privacy

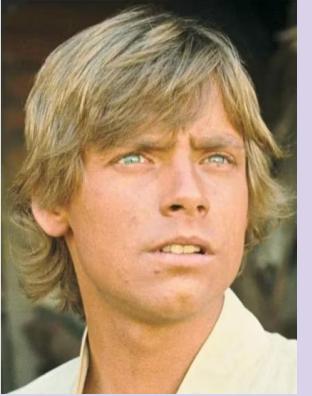
- Private data management
- DID
- Secret ballots
- Fully confidential NFT unlockables
- Sealed bid auctions
- Games with zero trust assumptions
- Private payments



How to Earn \$

How to become a predictoor

Calling all AI/ML jedis



1. Play with
predictoor.ai

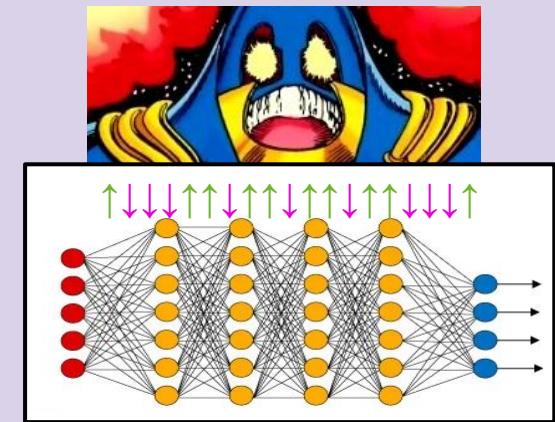
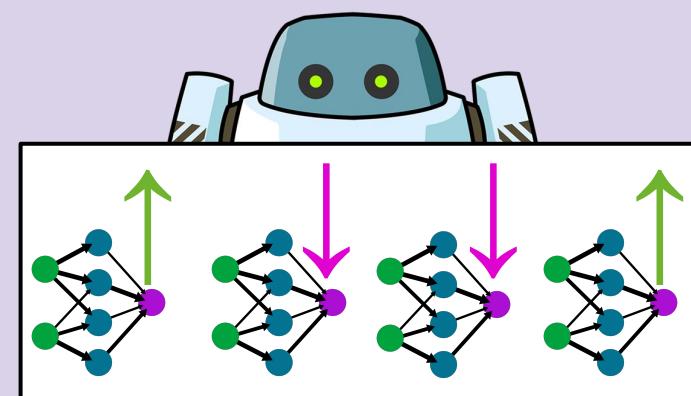
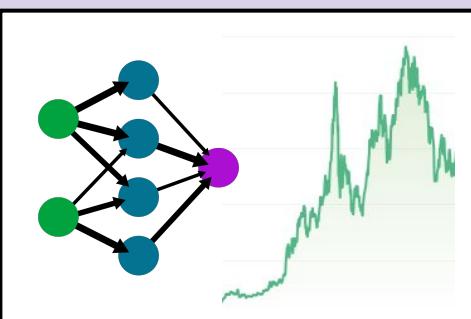
2. Do Challenge DF
One-off predictions

3. Run a predictoor bot
Continuous predictions

4. Optimize the bot
Increase \$\$

A screenshot of the predictoor.ai dashboard. It shows a table of asset balances and a line chart of price movements over time.

Asset (\$m)	11/19	11/20	11/21	11/22	11/23
BTC-USD	\$26250.91	\$26237.44	\$26247.8	\$26247.8	\$26247.8
DODGE-USDOT	\$0.06361	\$0.06356	\$0.06356	\$0.06356	\$0.06356
BTC-USD	\$26213.62	\$26198.19	\$26198.19	\$26213.51	-
ETH-USD	\$1645.09	\$1644.29	\$1644.87	-	42.2%
ADA-USD	\$0.2578	\$0.2575	\$0.2576	-	55.3%
BNB-USD	\$216.9	\$216.7	\$216.8	-	50.5%
TRX-USD	\$0.07902	\$0.07899	\$0.07899	-	50%
SOL-USD	\$19.77	\$19.76	\$19.79	-	53%
XRP-USD	\$0.5037	\$0.5036	\$0.5039	-	48.4%
DOT-USD	\$4.284	\$4.274	\$4.278	-	49.6%
LTC-USD	\$60.24	\$60.21	\$60.27	-	49.7%



How Predictoors Earn \$

1. Feed sales

- All \$ goes to predictoors (except a % for fees)
- Distributed pro-rata by stake among correct predictoors

2. Stake reshuffling

- Incorrect predictoors have their stake slashed
- Slashed stake is distributed pro-rata among correct pdr's

3. Predictor Data Farming

- An incentive program that amounts to additional \$ for pdr's
- 37,000 OCEAN/week rewards ($\approx \$10K/week$)

4. (Bonus) Challenge DF. One-off predictions

5. (Bonus) Passive DF & Volume DF. Lock OCEAN for veOCEAN, curate

How to become a trader in Predictoor

Calling all trading wizards

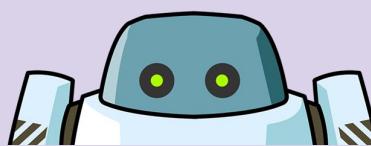


1. Play with
predictoor.ai

A screenshot of the predictoor.ai web application. The interface includes a top navigation bar with links like 'Oasis Sapphire Testnet', 'ENVIRON', 'SDK', and 'About'. Below this is a table showing asset prices (e.g., BTC-FUSD, DOGE-USDT) over time (11:15, 11:20, 11:23, 11:25). A section titled 'Accuracy' shows 40% for 'Predict 1' and 18% for 'Predict 2'. At the bottom, there's a 'BALANCE = 1.95\$' and a 'Subscription' section.

Asset (Sm)	11:15	11:20	11:23	11:25	Accuracy	Subscription
BTC-FUSD	\$26250.01	\$26237.44	\$26247.8	-	40%	= 3 / 24h
DOGE-USDT	\$0.0301	\$0.03056	\$0.03058	-	18%	= 3 / 24h
ETH-USDT	\$26213.02	\$26198.19	\$26215.51	-	40.8%	= 3 / 24h
	\$1645.09	\$1644.29	\$1644.87	-	41.2%	= 3 / 24h

2. Run a trader bot
Continuous trading



3. Optimize the bot
Increase \$\$



How Traders Earn \$

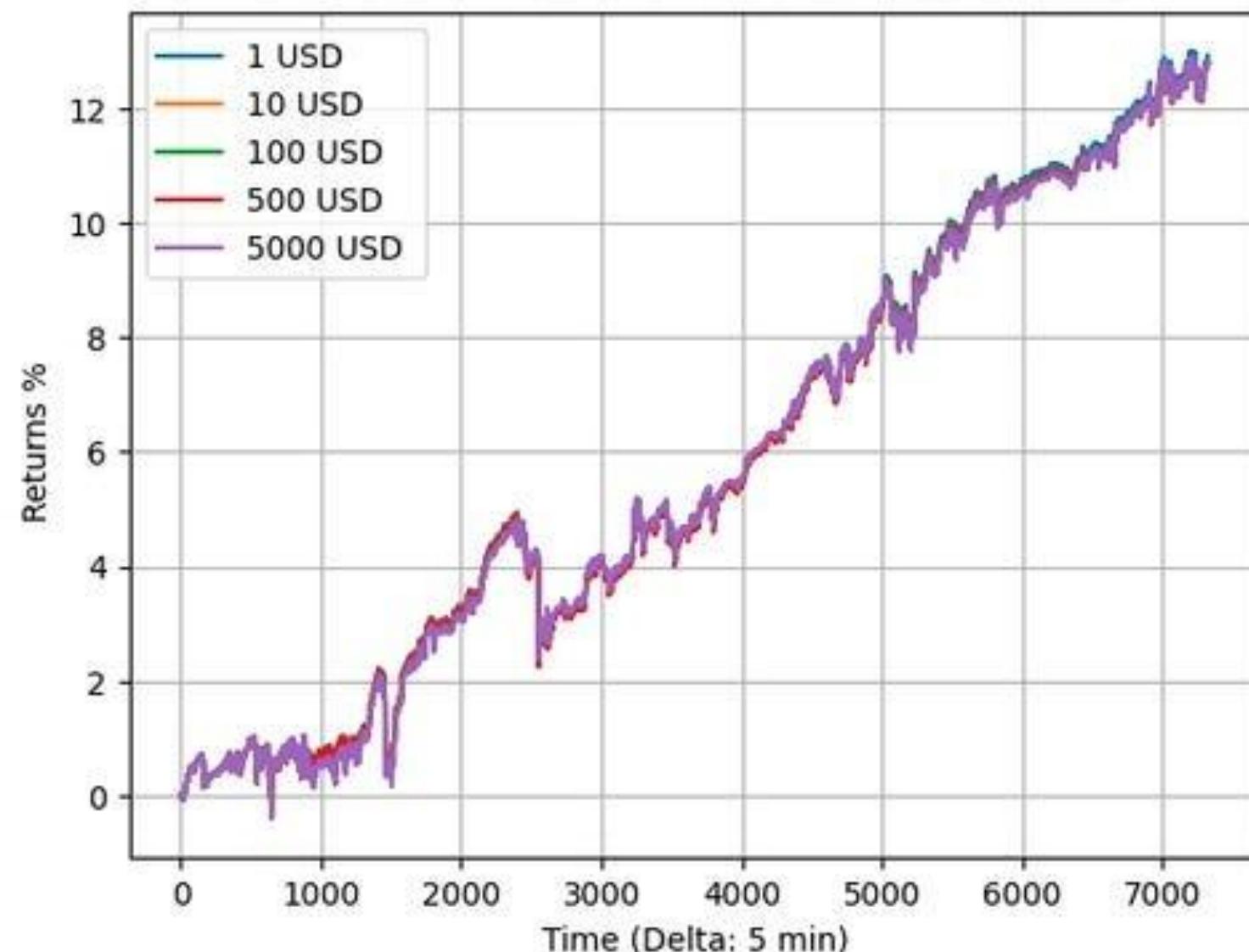
One single yet powerful main way to earn:

Trading revenue. Buy low and sell high (and vice versa).

- All \$ goes to predictoors (except a % for fees)
- Distributed pro-rata by stake among correct predictoors

Bonus: **Passive DF & Volume DF.** Lock OCEAN for veOCEAN, curate.

Sample Benchmark



BTC/USDT on Binance

Model training dates:
Jan 1, 2021 to Jun 30, 2023

Prediction dates:
Jul 1, 2023 to Jul 24, 2023
Always 5m ahead; 7000 ticks

"Baseline" trading strategy:

- if prediction is " \uparrow " then buy, if " \downarrow " then sell
- exit the position 5 min later

This example:

- accounts for order-book slippage
- ignores trading fee

Predictoor for devs & entrepreneurs

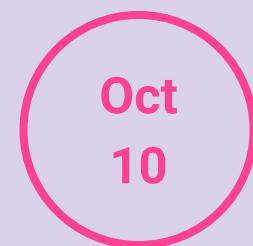
- **Create your own dapps & prediction feeds for a different vertical**
 - Code is all open-source Apache2
- **Use prediction feeds as a new DeFi primitive. "Future oracles"**
 - For derivatives on top. Relation with options, futures, etc
 - For stablecoins & loans: predictions to reduce collateralization ratio
- **And whatever you else you dream of:)**

Timeline



**Predictoor
Testnet**

4
wks



**Predictoor
Mainnet**
60min feeds

4
wks



**Predictoor
Data Farming**

Conclusion

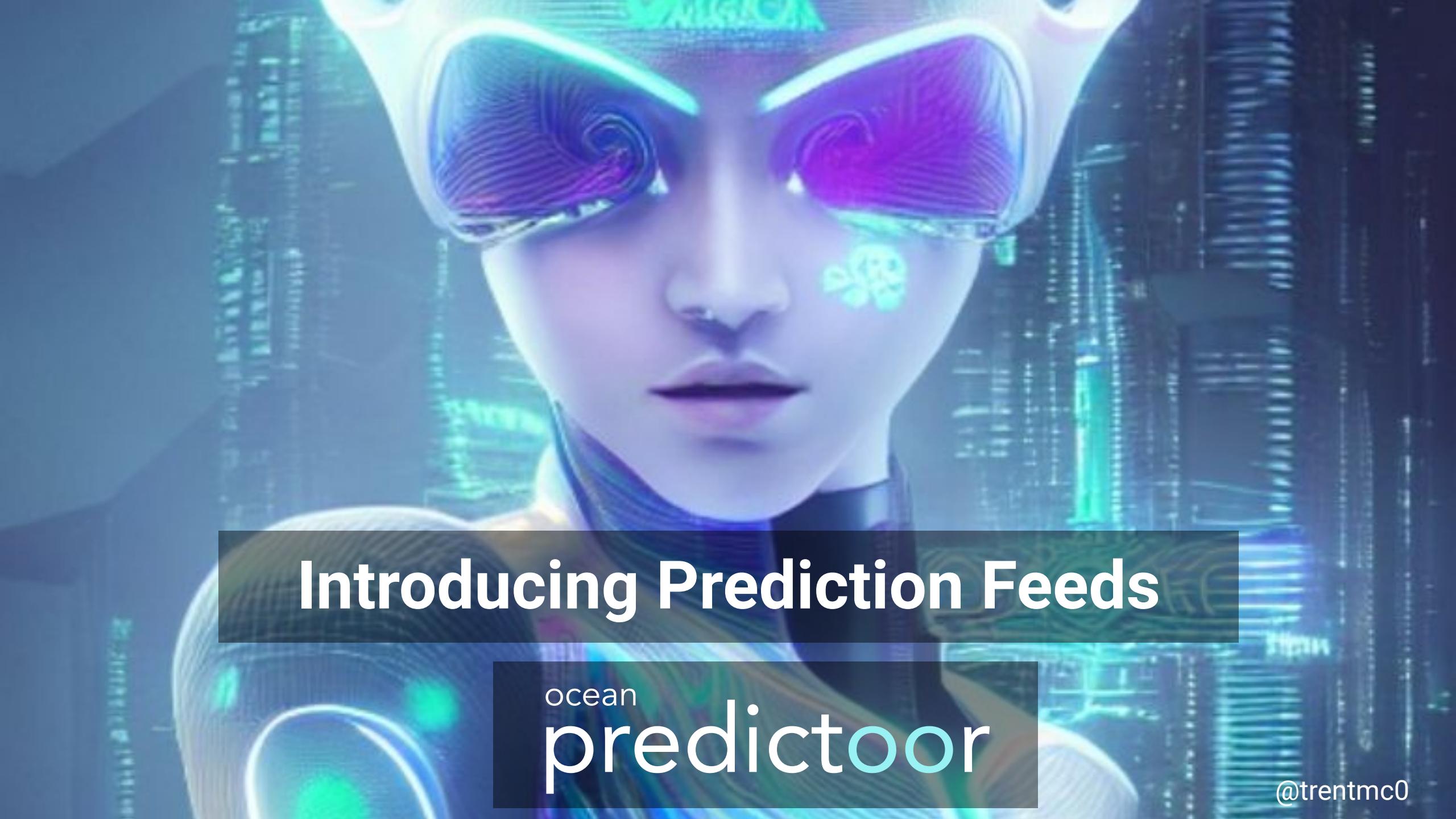
Introducing Predictoor

Accountable Accurate Prediction Feeds

- Crowd-sources predictions
- Globally distributed, non-custodial
- Built on Ocean stack
- Privacy-preserving via Oasis Sapphire
- Predictoors make \$ doing AI/ML;
traders \$ by using feeds

Try at
predictoor.ai



A woman with short blonde hair is shown from the chest up, wearing large, white, glowing sunglasses. The lenses of the sunglasses have a vibrant, swirling pattern of blue, green, and purple. She has a small, glowing green flower tattoo on her left cheek. The background is a dark, blurred image of a futuristic city at night, with tall buildings and glowing windows.

Introducing Prediction Feeds

ocean
predictoor

@trentmc0

Appendix

How to get more accurate prediction feeds

Main idea:

- Get people to put \$ where their mouth is
- Accurate prediction → make \$. Inaccurate → lose \$.
- Q: Isn't this prediction markets? They use \$ for accountability.
- No, those are for one-off predictions. E.g. "will Biden win in 2024?"
- What we want: a stream of predictions over time, ie a "prediction feed"
- Time-series not one-offs