

Market Agent Architecture v2.0 - Полный технологический стек

1. Структура проекта с технологиями и ссылками

```

market_agent/
├── config/
│   ├── settings.py      # [Pydantic](https://docs.pydantic.dev/) для валидации настроек
│   ├── strategies/      # [YAML](https://pyyaml.org/) для конфигураций
│   │   ├── day_trading.yaml
│   │   ├── swing_trading.yaml
│   │   └── long_term.yaml
│   └── tracker_profiles.json # JSON профили инвесторов
├── core/                # Ядро системы
│   ├── __init__.py
│   ├── event_bus.py     # [asyncio](https://docs.python.org/3/library/asyncio.html), [aio-pika]
│                       (https://aio-pika.readthedocs.io/)
│   ├── cache_manager.py # [Redis](https://redis.io/), [redis-py](https://redis-py.readthedocs.io/)
│   ├── rate_limiter.py  # [aiolimiter](https://aiolimiter.readthedocs.io/), Token Bucket Algorithm
│   └── async_executor.py # [asyncio](https://docs.python.org/3/library/asyncio.html),
│                       [concurrent.futures](https://docs.python.org/3/library/concurrent.futures.html)
├── infrastructure/      # Инфраструктурный слой
│   ├── __init__.py
│   ├── circuit_breaker.py # [tenacity](https://tenacity.readthedocs.io/), [pybreaker]
│                       (https://pypi.org/project/pybreaker/)
│   ├── health_check.py  # [aiohttp](https://docs.aiohttp.org/), [fastapi-health]
│                       (https://pypi.org/project/fastapi-health/)
│   ├── queue_manager.py # [RabbitMQ](https://www.rabbitmq.com/), [Redis Pub/Sub]
│                       (https://redis.io/docs/manual/pubsub/)
│   └── monitoring.py     # [Prometheus](https://prometheus.io/), [prometheus-client]
│                       (https://github.com/prometheus/client_python)
├── input_collection/    # Сбор данных
│   ├── __init__.py
│   ├── async_data_collector.py # [aiohttp](https://docs.aiohttp.org/), [httpx](https://www.python-httpx.org/)
│   └── market_data/     # Рыночные данные

```

- | | | — polygon_client.py # [Polygon.io API](https://polygon.io/) (\$199/месяц)
- | | | — alpaca_client.py # [Alpaca Markets API](https://alpaca.markets/) (\$99/месяц)
- | | | — twelve_data_client.py # [Twelve Data API](https://twelvedata.com/) (\$79-329/месяц)
- | | — social_data/ # Социальные сигналы
- | | | — bullaware_client.py # [BullAware API](https://bullaware.com/) (custom pricing)
- | | | — stocktwits_client.py # [StockTwits API](https://api.stocktwits.com/developers)
- | | — fundamental_data/ # Фундаментальные данные
- | | | — fmp_client.py # [Financial Modeling Prep](https://financialmodelingprep.com/) (\$69/месяц)
- | | | — sec_api_client.py # [SEC-API.io](https://sec-api.io/) (\$89/месяц)
- | | — news_sentiment/ # Новости и sentimento
- | | | — finnhub_client.py # [Finnhub API](https://finnhub.io/) (бесплатно - \$1900/месяц)
- | | | — custom_llm_analyzer.py # [NewsAPI](https://newsapi.org/) + LLM
- | | — processing_analysis/ # Обработка и анализ
- | | | — __init__.py
- | | | — llm_orchestrator.py # [Claude API](https://www.anthropic.com/api), [OpenAI API](https://openai.com/api/)
- | | | — ensemble_scorer.py # [scikit-learn](https://scikit-learn.org/), [XGBoost](https://xgboost.readthedocs.io/)
- | | | — feature_engineering.py # [pandas](https://pandas.pydata.org/), [numpy](https://numpy.org/)
- | | | — correlation_engine.py # [scipy](https://scipy.org/), [statsmodels](https://www.statsmodels.org/)
- | | | — pre_filter.py # Custom logic с [numba](https://numba.pydata.org/) для скорости
- | | | — risk_manager.py # [pyfolio](https://github.com/quantopian/pyfolio), [empyrical](https://github.com/quantopian/empyrical)
- | | | — sentiment_pattern_learner.py # [transformers](https://huggingface.co/transformers/)
- | | | — signal_predictor.py # [TensorFlow](https://www.tensorflow.org/), [PyTorch](https://pytorch.org/)
- | | | — strategy_engine.py # [ta-lib](https://ta-lib.org/), [pandas-ta](https://github.com/twopirllc/pandas-ta)
- | | | — tie_breaker.py # Custom logic
- | | — decision_execution/ # Исполнение решений
- | | | — __init__.py
- | | | — order_manager.py # [asyncio](https://docs.python.org/3/library/asyncio.html)

```
| | └─ execution_algos/      # Алгоритмы исполнения
| |   | └─ twap.py         # Time-Weighted Average Price
| |   | └─ vwap.py         # Volume-Weighted Average Price
| |   | └─ iceberg.py      # Iceberg orders
| | └─ ibkr_connector.py   # [ib_async](https://github.com/ib-api-reloaded/ib_async) (ex ib_insync)
|
| └─ monitoring_feedback/  # Мониторинг и обратная связь
|   | └─ __init__.py
|   | └─ real_time_monitor.py # [Grafana](https://grafana.com/), [Dash](https://plotly.com/dash/)
|   | └─ alert_manager.py   # [Slack SDK](https://slack.dev/python-slack-sdk/), [Telegram Bot API]
|                               (https://python-telegram-bot.org/)
|   | └─ backtest_engine.py  # [Backtrader](https://www.backtrader.com/), [Zipline](https://www.zipline.io/)
|   | └─ performance_monitor.py # [MLflow](https://mlflow.org/), [Weights & Biases](https://wandb.ai/)
|   | └─ portfolio_memory.py # [SQLAlchemy](https://www.sqlalchemy.org/), [PostgreSQL]
|                               (https://www.postgresql.org/)
|   | └─ reporter.py        # [Jinja2](https://jinja.palletsprojects.com/), [WeasyPrint](https://weasyprint.org/)
|
| └─ security/            # Безопасность
|   | └─ __init__.py
|   | └─ audit_log.py       # [structlog](https://www.structlog.org/), [python-json-logger]
|                               (https://pypi.org/project/python-json-logger/)
|   | └─ secrets_manager.py  # [HashiCorp Vault](https://www.vaultproject.io/), [AWS Secrets Manager]
|                               (https://aws.amazon.com/secrets-manager/)
|
| └─ tests/               # Тестирование
|   | └─ unit/            # [pytest](https://docs.pytest.org/), [pytest-asyncio](https://pytest-
|                               asyncio.readthedocs.io/)
|   |   | └─ integration/  # [testcontainers](https://testcontainers-python.readthedocs.io/)
|   |   | └─ backtesting/  # [pytest-benchmark](https://pytest-benchmark.readthedocs.io/)
|   |
|   └─ scripts/           # Вспомогательные скрипты
|     | └─ data_scout.py    # Scheduled c [APScheduler](https://apscheduler.readthedocs.io/)
|     | └─ cache_warmer.py  # Pre-load cache утилита
```

```
| └─ health_checker.py    # System health проверка
|
└─ docker/               # Контейнеризация
    └─ Dockerfile         # [Docker](https://www.docker.com/)
    └─ docker-compose.yml # [Docker Compose](https://docs.docker.com/compose/)
    └─ k8s/               # [Kubernetes](https://kubernetes.io/) манифесты
```

2. Подробное описание технологий по слоям

2.1 Configuration Layer

Технологии:

- Pydantic - валидация настроек с type hints
- python-dotenv - загрузка .env файлов
- PyYAML - парсинг YAML конфигураций
- Dynaconf - динамическое управление настройками

2.2 Core Layer (Ядро системы)

Технологии:

- Redis - in-memory кэширование и pub/sub
- RabbitMQ - надёжная доставка сообщений
- Apache Kafka - для high-throughput streaming (опционально)
- asyncio - асинхронное программирование
- aiohttp - асинхронные HTTP запросы
- httpx - современная альтернатива requests

2.3 Infrastructure Layer

Технологии:

- [Prometheus](#) + [Grafana](#) - метрики и визуализация
- [ELK Stack](#) - централизованное логирование
- [Jaeger](#) - distributed tracing
- [tenacity](#) - retry логика с backoff
- [circuitbreaker](#) - защита от каскадных сбоев

2.4 Input Collection Layer

API провайдеры и их стоимость:

Market Data (Рыночные данные):

- [Polygon.io](#) - \$199/месяц для real-time, лучший для US stocks
- [Alpaca Markets](#) - \$99/месяц, включает trading API
- [Twelve Data](#) - \$79-329/месяц, глобальное покрытие
- [Alpha Vantage](#) - бесплатно-\$250/месяц
- [Finnhub](#) - бесплатно-\$1900/месяц

Social Trading:

- [BullAware API](#) - custom pricing, eToro данные
- [StockTwits API](#) - бесплатно
- [TipRanks API](#) - enterprise pricing
- [Sentifi API](#) - enterprise pricing

Fundamental Data:

- [Financial Modeling Prep](#) - \$69-499/месяц

- [EOD Historical Data](#) - €99/месяц
- [Quandl](#) - \$99-599/месяц
- [IEX Cloud](#) - закрылся в 2024!

SEC Filings:

- [SEC-API.io](#) - \$89-499/месяц
- [Quiver Quantitative](#) - \$30-150/месяц
- [EDGAR Online](#) - enterprise

2.5 Processing & Analysis Layer

ML/AI библиотеки:

- [scikit-learn](#) - классический ML
- [XGBoost](#) - gradient boosting
- [LightGBM](#) - быстрый gradient boosting
- [TensorFlow](#) / [PyTorch](#) - deep learning
- [Hugging Face Transformers](#) - NLP модели

LLM провайдеры:

- [Claude API \(Anthropic\)](#) - \$15/million tokens
- [OpenAI API](#) - \$10-60/million tokens
- [Google Gemini API](#) - \$7-35/million tokens

Технический анализ:

- [TA-Lib](#) - классическая библиотека индикаторов
- [pandas-ta](#) - современная альтернатива

- [tulipy](#) - быстрая C библиотека

2.6 Decision & Execution Layer

Брокерские API:

- [Interactive Brokers API](#) - официальный API
- [ib_async](#) - Python wrapper для IBKR
- [Alpaca Trading API](#) - commission-free trading
- [TD Ameritrade API](#) - полнофункциональный API

2.7 Monitoring & Feedback Layer

Инструменты мониторинга:

- [Prometheus](#) - сбор метрик
- [Grafana](#) - визуализация
- [Datadog](#) - enterprise monitoring
- [New Relic](#) - APM решение

Backtesting frameworks:

- [Backtrader](#) - популярный Python framework
- [Zipline](#) - от Quantopian
- [PyAlgoTrade](#) - простой framework
- [bt](#) - flexible backtesting

2.8 Security Layer

Инструменты безопасности:

- [HashiCorp Vault](#) - управление секретами

- [AWS Secrets Manager](#) - облачное решение
- [Azure Key Vault](#) - Microsoft решение
- [CyberArk](#) - enterprise security

3. Базы данных и хранилища

Time-Series данные:

- [InfluxDB](#) - специализированная TSDB
- [TimescaleDB](#) - PostgreSQL расширение
- [Apache Cassandra](#) - distributed NoSQL
- [QuestDB](#) - высокопроизводительная TSDB

Transactional данные:

- [PostgreSQL](#) - основная БД
- [MongoDB](#) - для неструктурированных данных
- [Redis](#) - кэширование и сессии

Data Lake/Warehouse:

- [Apache Parquet](#) - колоночный формат
- [Delta Lake](#) - ACID транзакции для data lakes
- [Apache Arrow](#) - in-memory формат

4. Deployment & DevOps

Container & Orchestration:

- [Docker](#) - контейнеризация
- [Kubernetes](#) - оркестрация

- Helm - package manager для K8s
- Terraform - Infrastructure as Code

CI/CD:

- GitHub Actions - CI/CD pipeline
- GitLab CI - альтернатива
- Jenkins - классический выбор
- ArgoCD - GitOps для K8s

Cloud Providers:

- AWS - полный набор сервисов
- Google Cloud - хорошая интеграция с AI/ML
- Azure - enterprise выбор
- DigitalOcean - простой и недорогой

5. Примерная стоимость инфраструктуры

MVP версия (личное использование):

- Market Data: Polygon.io Starter - \$199/месяц
- Fundamental: Financial Modeling Prep - \$69/месяц
- LLM: OpenAI/Claude API - ~\$50-100/месяц
- Hosting: DigitalOcean - \$50-100/месяц
- **Итого: ~\$400-500/месяц**

Production версия (профессиональная):

- Market Data: Polygon.io Business - \$999/месяц

- All data sources premium - ~\$2000/месяц
- LLM APIs - \$500-1000/месяц
- Cloud Infrastructure - \$1000-2000/месяц
- **Итого: ~\$5000-7000/месяц**

Enterprise версия:

- Direct market data feeds - \$10,000+/месяц
- Dedicated infrastructure - \$5,000+/месяц
- Premium APIs & services - \$10,000+/месяц
- **Итого: \$25,000+/месяц**