Day 7: AI Agents, LLMs, and Autonomous Finance

POMDPs, Multi-Agent Game Theory, and Formal Security

Prof. Jörg Osterrieder

PhD Seminar – Digital Finance Research

2026

PhD Seminar Series: Digital Finance Research

Recap: Days 1–6

Foundations (Days 1–3)

- Stochastic models, jump-diffusions
- CFMM mathematics, AMM design
- MEV, mechanism design, fee markets

Applications (Days 4–5)

- ML microstructure, deep LOB
- Network risk, EVT, copulas
- Regulation and systemic fragility

Day 6: DeFi Derivatives

- Perp funding: $f = r - y$
- Power perps: $\Gamma_{\$} = p(p-1)V$
- DOVs, structured products, CPPI

Today's Theme

Autonomous agents as market participants:

- Formal agent models (POMDPs)
- Multi-agent game theory
- LLM oracles and security
- Account abstraction

When GPT-4 Resolves a $100M Bet

Polymarket + UMA Oracle System

- LLM-based oracles resolve prediction markets
- 89% accuracy on 1,660 resolved markets (2024–2025)
- **Failure case:** $500K market incorrectly resolved due to LLM misinterpreting a merger headline
- Staked bonds: $10K per resolution attempt

"Should we trust AI with financial decisions? If yes, under what conditions? If no, what is the alternative?"

Key Statistics

Metric	Value
Markets resolved	1,660
Accuracy	89%
Total volume	$2.3B
Avg. bond	$10K
Challenge rate	4.2%
Overturn rate	1.1%

89% accuracy on $2.3B volume $\Rightarrow$ $253M exposed to incorrect resolution risk.

Today's Roadmap

1. Formal Agent Models
2. Multi-Agent Game Theory in DeFi
3. LLM Oracles, Security, and Account Abstraction
4. Synthesis and Research Frontiers

Formal Agent Models

POMDPs · Belief States · Optimal Policies

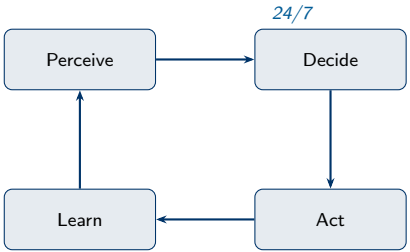
What Is an AI Agent?

Definition 1

An **AI agent** is a system that (1) *perceives* its environment through observations, (2) *decides* actions based on a learned or programmed policy, and (3) *acts* on the environment—all autonomously and continuously.

Chatbot vs. Agent

	Chatbot	Agent
Perceives env.	No	Yes
Takes actions	No	Yes
Continuous	No	Yes
Has goals	No	Yes
Learns from acts	No	Yes



The agent loop: perceive → decide → act → learn → repeat.

Formalizing Agents as POMDPs

Definition 2 (Partially Observable Markov Decision Process)

A **POMDP** is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, T, O, R, \gamma)$:

- $\mathcal{S}$: state space (true market state, protocol state, balances)
- $\mathcal{A}$: action space (trade, deposit, withdraw, rebalance)
- $\mathcal{O}$: observation space (oracle prices, on-chain data, API feeds)
- $T(s'|s, a)$: transition probability
- $O(o|s', a)$: observation probability (noisy perception)
- $R(s, a)$: reward function (returns, fees, penalties)
- $\gamma \in (0, 1)$: discount factor

Key Distinction from MDP

Agent observes $o_t \in \mathcal{O}$ rather than true state $s_t \in \mathcal{S}$. It maintains a **belief state** $b_t \in \Delta(\mathcal{S})$—a probability distribution over possible states.

Belief Update and Optimal Policy

Bayesian Belief Update

After taking action a_t and observing o_{t+1}:

$$b_{t+1}(s') = \frac{O(o_{t+1}|s', a_t) \sum_{s \in \mathcal{S}} T(s'|s, a_t)\, b_t(s)}{\sum_{s''} O(o_{t+1}|s'', a_t) \sum_{s} T(s''|s, a_t)\, b_t(s)} \tag{1}$$

Optimal Policy

The optimal policy π^* maximizes:

$$V^*(b) = \max_{a \in \mathcal{A}} \left[R(b, a) + \gamma \sum_{o \in \mathcal{O}} \mathbb{P}(o|b, a)\, V^*(\tau(b, a, o \right. \tag{2}$$

where $\tau(b, a, o)$ is the updated belief.

Computational Complexity

- Exact POMDP solving: **PSPACE-hard**
- Continuous state/action: intractable
- Practical approaches:
  - Point-based value iteration
  - Monte Carlo tree search
  - Deep RL with belief encoding

DeFi Agent as a POMDP: Concrete Instantiation

Yield Optimization Agent

State space $\mathcal{S}$:

- True lending rates across n protocols
- Protocol solvency status (healthy/at-risk)
- Gas prices across m chains
- Pending governance proposals

Observation space $\mathcal{O}$:

- Reported APY (may lag true rate)
- Oracle price feeds (may be stale/manipulated)
- Gas estimators (noisy)
- On-chain TVL (delayed)

Action space $\mathcal{A}$:

- Deposit amount d_i to protocol i
- Withdraw amount w_i from protocol i
- Bridge across chains
- Hold (do nothing)

Reward:

$$R_t = \sum_i r_i d_i \Delta t - c_{\text{gas}} - c_{\text{bridge}} - L_t \qquad (3)$$

where L_t captures exploit losses.

Partial observability: The agent cannot directly observe protocol solvency or oracle manipulation.

Worked Example: Yield Agent Decision

Setup

Agent observes: Aave USDC 2.0%, Compound 4.5%, Morpho 5.1%.
Portfolio: \$50K in Aave. Gas: ETH \$3, Arbitrum \$0.05, Base \$0.02.

Belief state update:

- $b(\text{Morpho safe}) = 0.95$ (audited, high TVL)
- $b(\text{rate stable}) = 0.80$ (rates can shift)
- Expected improvement:
  $(5.1\% - 2.0\%) \times \$50\text{K} = \$1{,}550/\text{yr}$

Decision under uncertainty:

$$\begin{aligned}\mathbb{E}[R|\text{switch}] &= 0.95 \times 0.80 \times \$1{,}550 - \$3.52 \\ &= \$1{,}174 \\ \mathbb{E}[R|\text{hold}] &= \$0\end{aligned}$$

$\Rightarrow$ Switch. Breakeven $<$ 1 day.

Why POMDP, not MDP?

- Agent cannot *see* true Morpho solvency
- Oracle feed may be stale (5-min delay)
- Gas estimate $\pm 20\%$ uncertainty
- Rate may change before tx confirms

Oracle Problem

Can the agent trust the data it perceives? If oracle feeds are manipulated, the belief update is **adversarially corrupted**:

$$b_{t+1}(s') \propto O(\tilde{o}|s', a)\, b_t(s)$$

Agent Architectures: ReAct Framework

Definition 3 ([ ])

ReAct alternates between:

- **Thought:** reasoning step (natural language)
- **Action:** tool/API call (structured)
- **Observation:** environment response

This grounds LLM reasoning in real-world feedback.

Formal Model

Define the ReAct trace as a sequence $\tau = (t_1, a_1, o_1, t_2, a_2, o_2, \ldots)$ where $t_i \in \mathcal{L}$ (language space), $a_i \in \mathcal{A}$, $o_i \in \mathcal{O}$.
The policy:

$$\pi_{\text{ReAct}}(a_i | t_i) = \text{LLM}(a_i | t_1, a_1, o_1, \ldots, t_i) \tag{4}$$

Token cost: $|\tau| = \sum_i (|t_i| + |a_i| + |o_i|)$.

ReAct vs. pure planning (plan-then-execute): ReAct is more robust but 2–5$\times$ more tokens per task [5].

ReAct Trace: DeFi Risk Management Agent

Context: Agent manages leveraged ETH on Aave. Threshold: HF > 1.3.

Thought 1: ETH dropped 8%. Check health factor.

Action 1: `aave.getAccountData("0x...")`

Obs 1: `healthFactor: 1.15` < 1.3

Thought 2: HF below threshold. Need to repay debt. Check USDC balance.

Action 2: `usdc.balanceOf("0x...")`

Obs 2: `5200 USDC available`

Thought 3: Repaying 2000 USDC: new HF $= 5.0 \times 0.825/3.13 = 1.32 > 1.3$. Proceed.

Action 3: `aave.repay(USDC, 2000)`

Obs 3: `Confirmed. HF: 1.32`

Thought 4: Position secured. Set alert for further drops.

Action 4: `alert.set(ETH, below, 2600)`

Obs 4: `Alert configured.`

Key Property

Each observation *grounds* the next thought. The agent verifies its actions against real on-chain state—not hallucinated assumptions.

Architecture Comparison: Plan-Execute vs. ReAct vs. Reflexive

	Plan-Execute	ReAct	Reflexive
Planning	Full upfront	Interleaved	None
Grounding	None (open loop)	Per-step	None
Robustness	Low (plan fragile)	High	Low
Token cost	Low	2–5×	Minimal
Error recovery	None	Per-step	None
Best for DeFi:			
Use case	Static batch	Interactive	Simple trigger

DeFi-Specific Failure Modes

- **Hallucinated tool calls:** LLM invents non-existent contract methods
- **Observation misinterpretation:** confuses revert message with success
- **Infinite loops:** agent retries failing tx indefinitely (gas drain)
- **Stale context:** reasoning about prices from 5 blocks ago

Tool-Use and Chain-of-Thought in Finance

Tool-Use [ ]

LLM generates structured API calls:

`getPrice(ETH/USD)` $\rightarrow$ \$3,000

`getGas(base)` $\rightarrow$ 0.02 gwei

`simulate(swap, 1000)` $\rightarrow$ slippage: 0.3%

The LLM is the "brain"; tools are the "hands."

Tool taxonomy for DeFi:

- Read: oracle, RPC, subgraph
- Compute: portfolio analytics, risk metrics
- Write: sign & submit transactions

Chain-of-Thought [ ]

Step-by-step reasoning before action:

1. Current position: 5 ETH collateral, 3.8 ETH debt
2. Health factor: $5 \times 0.825/3.8 = 1.086$
3. Below threshold 1.3 $\Rightarrow$ must act
4. Repay 0.67 ETH-equivalent: HF $\rightarrow$ 1.32
5. Execute repayment

CoT Improves Accuracy

- Without CoT: 62% correct DeFi actions
- With CoT: 84% correct (benchmark)
- Critical for multi-step transactions

Complexity of the DeFi Action Space

Action Space Combinatorics

For an agent operating across n protocols, m chains, k assets:

$$|\mathcal{A}| = \underbrace{n \cdot k}_{\text{deposit/withdraw}} \times \underbrace{\binom{k}{2}}_{\text{swap pairs}} \times \underbrace{m}_{\text{chains}} \times \underbrace{L}_{\text{leverage levels}} = \mathcal{O}(nk^2mL) \tag{5}$$

Typical DeFi agent: $n = 15$, $k = 20$, $m = 5$, $L = 10 \Rightarrow |\mathcal{A}| \approx 10^5$.

Curse of Dimensionality

- State space: gas, rates, prices, utilization across all protocols
- Continuous state $\Rightarrow$ function approximation needed
- Deep RL: promising but sample-inefficient

Practical Decomposition

Hierarchical agent architecture:

1. **Strategic layer:** asset allocation (slow, $\Delta t = 1\,\text{h}$)
2. **Tactical layer:** protocol selection (medium, $\Delta t = 5\,\text{min}$)

Multi-Agent Game Theory

Strategy Spaces · Nash Equilibria · Emergent Dynamics

Multi-Agent DeFi Systems

Definition 4

A **multi-agent system (MAS)** in DeFi consists of N autonomous agents $\{1, \ldots, N\}$ interacting in a shared environment (DEX, lending protocol). Each agent i has strategy $s_i \in S_i$ and payoff $u_i(s_1, \ldots, s_N)$.

Agent Types

Type	Strategy	Goal
LP	Range, depth	Max fees
Arb	Threshold	Risk-free π
MEV	Sandwich	Extract value
Liquidator	Monitor	Penalty fees

Interactions

- **Cooperative:** Arb $\rightarrow$ LP (generates fees)
- **Adversarial:** MEV $\rightarrow$ Arb (increases slippage)
- **Symbiotic:** LP $\rightarrow$ Arb (reduces slippage)
- **Parasitic:** MEV $\rightarrow$ LP (extracts value)

Formal Game: Three-Agent Uniswap Interaction

Normal-Form Game $\Gamma = (N, \{S_i\}, \{u_i\})$

Players: $N = \{\text{LP}, \text{Arb}, \text{MEV}\}$
Strategy spaces:

$$S_{\text{LP}} = \{(p_l, p_u, L) : p_l < p_u,\ L > 0\} \quad \text{(range, liquidity)} \tag{6}$$

$$S_{\text{Arb}} = \{\delta > 0 : \text{trade when spread} > \delta\} \quad \text{(threshold)} \tag{7}$$

$$S_{\text{MEV}} = \{(q_f, g) : q_f \geq 0,\ g > 0\} \quad \text{(front-run size, gas bid)} \tag{8}$$

Payoff Functions

$$u_{\text{LP}} = \text{Fees}(L, \delta) - \text{IL}(p_l, p_u, \sigma) \tag{9}$$

$$u_{\text{Arb}} = \sum_t \max\big(|P_{\text{DEX}} - P_{\text{CEX}}| - \delta - c_{\text{gas}},\ 0\big) \tag{10}$$

$$u_{\text{MEV}} = q_f \cdot \lambda q_{\text{victim}} - 2c_{\text{gas}}(g) \tag{11}$$

where λ is the price impact parameter and q_{victim} is the target order size.

Nash Equilibrium in the Three-Agent Game

Theorem 5 (Existence)

Under standard regularity conditions (compact strategy spaces, continuous payoffs), the game Γ admits a Nash equilibrium $(s^*_{LP}, s^*_{Arb}, s^*_{MEV})$ by Glicksberg's theorem (mixed strategies).

Equilibrium Characterization

At the NE:

- LP widens range to reduce MEV extraction: $p^*_u - p^*_l \uparrow$ as MEV activity $\uparrow$
- Arb threshold increases (less profitable): $\delta^* \uparrow$ as MEV extracts from arb trades
- MEV front-run size bounded by gas competition: $q^*_f = \frac{q_{\text{victim}}\lambda - c_{\text{gas}}}{2\lambda}$

Welfare Analysis

Is the NE Pareto optimal? **Generally no.**

- MEV extraction is a **deadweight loss**
- LP provides less liquidity $\Rightarrow$ higher slippage for all
- Total welfare: $W^* < W^{\text{no-MEV}}$

Mechanism design question:
Can protocol rules eliminate MEV equilibrium? $\Rightarrow$ batch auctions, encrypted mempools, MEV redistribution [2].

Best Response Dynamics and Convergence

Best Response Mapping

For each agent i, the best response to others' strategies s_{-i}:

$$\mathrm{BR}_i(s_{-i}) = \arg\max_{s_i \in S_i} u_i(s_i, s_{-i}) \tag{12}$$

A Nash equilibrium is a fixed point: $s_i^* = \mathrm{BR}_i(s_{-i}^*)$ for all i.

Fictitious Play

Agent i at round k:

1. Observe empirical frequency of others' play
2. Best-respond to empirical distribution
3. Update strategy

$$\sigma_i^{(k)} = \frac{1}{k} \sum_{t=1}^{k} \mathbb{1}[s_i^{(t)}] \tag{13}$$

RL-Based Convergence

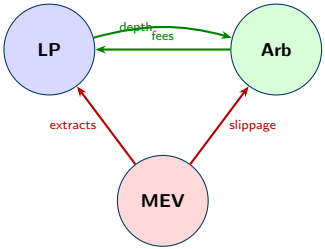
Each agent uses independent Q-learning:

$$Q_i(s,a) \leftarrow Q_i(s,a) + \alpha \left[r + \gamma \max_{a'} Q_i(s',a') - Q_i(s,a) \right] \tag{14}$$

Non-Stationarity

Each agent's environment is non-stationary (other

Emergent Dynamics: Feedback Loops



Green: cooperative. Red: adversarial.

Cascade Scenario

1. MEV increases $\Rightarrow$ Arb less profitable
2. Arb widens threshold $\Rightarrow$ less trading
3. Less trading $\Rightarrow$ LP fees decrease
4. LP withdraws liquidity $\Rightarrow$ higher slippage
5. Higher slippage $\Rightarrow$ MEV *more* profitable
6. **Positive feedback loop** $\Rightarrow$ liquidity spiral

Fragility

During stress (crash), all agents may simultaneously withdraw $\Rightarrow$ **coordination failure**. No agent internalizes the externality of withdrawal.

Game Theory of Liquidation Cascades

Liquidation as a Coordination Game

N agents each hold leveraged positions. Agent i with collateral c_i and debt d_i is liquidated when:

$$\frac{c_i \cdot P_t}{d_i} < m_{\min} \tag{15}$$

Liquidation of agent i pushes price down by $\Delta P_i = -\lambda \cdot c_i$, potentially triggering liquidation of agent j.

Cascade Condition

A cascade occurs when:

$$\sum_{i \in \mathcal{L}(P)} \lambda c_i > P - P_{\text{next}} \tag{16}$$

where $\mathcal{L}(P)$ is the set of agents liquidated at price P and P_{next} is the next liquidation threshold.

Contagion Network

Model as directed graph $G = (V, E)$:

- V: leveraged positions
- E: edge (i, j) if liquidation of i triggers j
- Cascade size $\sim$ largest connected component
- Connection to Eisenberg–Noe [1]

Mechanism Design for Multi-Agent DeFi

Design Goal

Choose protocol rules $\mathcal{M}$ such that the induced Nash equilibrium maximizes social welfare:

$$\mathcal{M}^* = \arg\max_{\mathcal{M}} W(s^*(\mathcal{M})) = \arg\max_{\mathcal{M}} \sum_{i=1}^{N} u_i\big(s_i^*(\mathcal{M}), s_{-i}^*(\mathcal{M})\big) \tag{17}$$

Approaches

1. **Batch auctions:** eliminate MEV by executing all orders simultaneously
2. **Encrypted mempools:** hide orders until committed
3. **MEV redistribution:** return extracted value to users/LPs
4. **Priority fees:** incentive-compatible ordering

Impossibility Result

Theorem 6 (Informal, [ ])

No transaction ordering mechanism can simultaneously be:

1. *Incentive compatible*
2. *MEV-free*
3. *Computationally efficient*

Information Aggregation with AI Agents

Market Efficiency and AI

If N AI agents each observe private signal $\theta_i = \theta + \epsilon_i$ about fundamental value θ, the market price aggregates information:

$$P = \mathbb{E}[\theta|\theta_1, \ldots, \theta_N] = \frac{\sum_{i=1}^{N} \frac{\theta_i}{\sigma_i^2}}{\sum_{i=1}^{N} \frac{1}{\sigma_i^2}} \tag{18}$$

where $\sigma_i^2 = \mathrm{Var}(\epsilon_i)$ is agent i's noise variance.

AI Agents Improve Aggregation

- Lower σ_i^2 (better signal processing)
- Faster incorporation (24/7, millisecond)
- More signals per agent (multi-source)

Risks to Aggregation

- Correlated signals: all agents use same LLM $\Rightarrow \mathrm{Corr}(\epsilon_i, \epsilon_j) \to 1$
- Herding: agents learn from each other's actions $\Rightarrow$ information cascades
- Homogeneity reduces effective N

Correlated AI Agents: Systemic Fragility

Proposition 7 (Effective Diversity)

If N agents use the same LLM backbone with correlation ρ between signal errors, the effective number of independent signals is:

$$N_{eff} = \frac{N}{1 + (N-1)\rho} \tag{19}$$

As $\rho \to 1$: $N_{eff} \to 1$ regardless of N.

Numerical Example

$N = 100$ agents, all GPT-4 based, $\rho = 0.8$:

$$N_{\text{eff}} = \frac{100}{1 + 99 \times 0.8} = \frac{100}{80.2} \approx 1.25$$

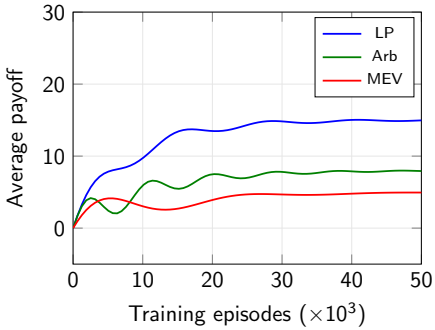
100 "independent" agents provide the diversity of $\sim$1.25 truly independent agents.

Flash Crash Scenario

1. LLM misinterprets news headline
2. All 100 agents sell simultaneously
3. Correlated selling $\Rightarrow$ cascading liquidations
4. Market crash amplified by agent homogeneity
5. No human in the loop to intervene

Implication: Model diversity (different LLMs, different training data) is a *systemic stability* requirement, not

Simulation: Convergence to Equilibrium Under RL



Stylized convergence of multi-agent RL to approximate NE.

Observations

1. LP converges fastest (simple strategy)
2. MEV is most volatile (adversarial landscape)
3. Arb payoff squeezed between LP needs and MEV extraction
4. Convergence takes $\sim$ 30K episodes (non-trivial)

Key Result

Independent RL does **not** converge to social optimum. Total welfare at NE: $W^* \approx 28$ vs. social optimum $W^{\text{opt}} \approx 35$.
Price of anarchy: PoA $= 35/28 = 1.25$.

LLM Oracles & Security

Calibration · Adversarial Attacks · Account Abstraction

LLM-Based Oracles: Architecture and Resolution

Definition 8

An **LLM oracle** uses a large language model to resolve natural-language questions that traditional price-feed oracles cannot answer: subjective, contextual, or judgment-requiring queries.

Resolution Pipeline (UMA/Polymarket)

1. Proposer asserts outcome + \$10K bond
2. 48 h challenge period
3. If disputed: LLM oracle invoked
4. LLM searches authoritative sources
5. Resolution + confidence score
6. Losing party forfeits bond

Oracle Taxonomy

	Chainlink	**LLM Oracle**
Input	Numerical	Natural lang.
Query	"ETH price?"	"Did EU approve X?"
Source	Exchanges	News, docs
Speed	Seconds	Minutes
Attack	Price manip.	Info. manip.

LLM Oracle Calibration Analysis

Definition 9

An oracle is **well-calibrated** if its stated confidence matches observed accuracy:

$$\mathbb{P}[\text{correct}|\text{confidence} = c] = c \qquad \forall\, c \in [0,1] \tag{20}$$

Brier Score

$$\text{BS} = \frac{1}{n}\sum_{i=1}^{n}(f_i - o_i)^2 \tag{21}$$

where f_i is forecast probability and $o_i \in \{0,1\}$ is the outcome.

Decomposition:

$$\text{BS} = \underbrace{\text{REL}}_{\text{calibration}} - \underbrace{\text{RES}}_{\text{resolution}} + \underbrace{\text{UNC}}_{\text{uncertainty}} \tag{22}$$

Empirical Results (Polymarket, 2024–2025)

Confidence bin	Accuracy
$[0.5, 0.6)$	0.52
$[0.6, 0.7)$	0.61
$[0.7, 0.8)$	0.73
$[0.8, 0.9)$	0.82
$[0.9, 1.0]$	0.93
Overall BS	0.14

Adversarial Manipulation: Game-Theoretic Model

Bayesian Persuasion Game

Players: Adversary (sender), LLM Oracle (receiver).
The adversary controls the LLM's information set by planting articles/sources:

$$\text{Adversary:} \quad \max_{\pi \in \Delta(\mathcal{I})} \sum_{\theta} \mu(\theta) \sum_{i \in \mathcal{I}} \pi(i|\theta)\, u_A\big(a^*(i)\big) \tag{23}$$

where π is the information structure, θ is the true state, μ is the prior, $\mathcal{I}$ is the set of possible signal realizations, and a^* is the LLM's optimal action given signal i.

Attack Cost Model

Cost to plant k fake articles:

$$c(k) = c_0 + c_1 k^{\alpha}, \quad \alpha > 1 \tag{24}$$

Probability of flipping LLM decision:

Practical Attack Vectors

1. **SEO poisoning:** rank fake articles in LLM's search results
2. **Source impersonation:** mimic authoritative domains
3. **Timing attack:** publish false info just before

Defense Mechanisms for LLM Oracles

Multi-Source Verification

Use M independent sources with majority vote:

$$\hat{\theta} = \begin{cases} 1 & \text{if } \sum_{j=1}^{M} \mathbb{1}[\text{source}_j = 1] > M/2 \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

Attack requires corrupting $> M/2$ sources.
Cost to attack: $c_{\text{defense}} = c(k^*) \cdot \lceil M/2 \rceil$ (linear scaling of defense cost).

Additional Defenses

1. **Temporal consistency:** require information to persist across multiple time windows
2. **Source reputation:** weight by historical reliability

$$w_j = \frac{\text{correct}_j}{\text{total}_j} \tag{27}$$

3. **Economic bonds:** proposer stakes collateral
4. **Human escalation:** disputes above \$X go to human jury

Open Problem

Optimal mechanism design for LLM oracles: balance accuracy, cost, speed, and manipulation resistance. No known solution achieves all four.

AI Agent Risks: Formal Taxonomy

Hallucination

LLM generates false output:

- Fake contract address
- Misinterpreted function
- Invented token symbol

Model:

$\mathbb{P}[\text{halluc.}] = h(\text{context})$

Typically $h \in [0.02, 0.15]$

Alignment Failure

Agent optimizes wrong objective:

- Maximizes yield, ignores SC risk
- Chases APY into rug pull
- Ignores user constraints

Model:

$u_{\text{agent}} \neq u_{\text{user}}$

Misalignment gap: $\|u_a - u_u\|$

Adversarial Attack

External manipulation:

- Oracle price manipulation
- Prompt injection
- Fake token honeypots

Model:

$\max_\pi \min_\delta \mathbb{E}[R(\pi, s + \delta)]$

Robust optimization

Unified Risk Framework

Total loss: $L = L_{\text{halluc}} + L_{\text{align}} + L_{\text{adv}}$. Expected loss per agent per year:

$$\mathbb{E}[L] = h \cdot \bar{L}_h + p_{\text{align}} \cdot \bar{L}_a + \lambda_{\text{attack}} \cdot \bar{L}_{\text{adv}} \tag{28}$$

Robust Agent Design: Min-Max Formulation

Adversarial Robustness

The agent solves a **robust MDP**:

$$\pi^* = \arg\max_{\pi} \min_{\delta \in \mathcal{D}} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t + \delta_t, a_t)\right] \tag{29}$$

where $\mathcal{D}$ is the adversary's perturbation set.

Perturbation Budget

- Price manipulation: $|\delta_P| \leq \epsilon_P$
- Oracle delay: $\delta_t \leq \tau_{\max}$
- Gas spike: $\delta_g \leq g_{\max}$
- Combined: $\|\delta\|_\infty \leq \epsilon$

Cost of Robustness

$$\text{Robustness gap} = V^*(\text{nominal}) - V^*(\text{robust}) \tag{30}$$

Empirical estimates: 5–15% reduction in expected return for agents operating in adversarial environments.

Trade-off: More robust agents are more conservative

Account Abstraction: ERC-4337 and EIP-7702

Definition 10

Account abstraction unifies EOAs and smart contract accounts, enabling programmable wallet logic: batch transactions, gas abstraction, social recovery, and delegated execution via **session keys**.

Before AA (EOA)

- 3 actions $\Rightarrow$ 3 transactions
- 3 gas payments (in ETH only)
- 3 manual signatures
- No delegation possible
- Cost: $\sim$\$15

After AA (Smart Wallet)

- 3 actions $\Rightarrow$ 1 UserOperation
- Gas in any token (Paymaster)
- 1 signature
- Session keys for AI agents
- Cost: $\sim$\$5 (batching savings)

Critical for AI Agents

Session keys enable **safe delegation**: agent operates within defined bounds without accessing the master private key.

Formal Security Model: Session Keys and Delegation

Definition 11

A **session key** κ is characterized by a permission tuple:

$$\kappa = (A_{\text{allowed}},\ L_{\max},\ T_{\text{exp}},\ C_{\text{contracts}}) \tag{31}$$

where $A_{\text{allowed}} \subseteq \mathcal{A}$ is the allowed action set, $L_{\max}$ is the spending limit, T_{exp} is the expiration time, and $C_{\text{contracts}}$ is the set of permitted target contracts.

Principal-Agent Framework

Owner (principal) delegates to agent with policy π subject to constraints:

$$\pi : \mathcal{O} \to A_{\text{allowed}} \tag{32}$$

$$\sum_t c(a_t) \leq L_{\max} \tag{33}$$

$$t \leq T_{\text{exp}} \tag{34}$$

Minimal Permission Design

Theorem 12 (Principle of Least Privilege)

The optimal permission set κ^* for a DeFi strategy σ satisfies:

$$\kappa^* = \arg\min_{|\kappa|} \{\kappa : u_{agent}(\sigma, \kappa) \geq u_{agent}(\sigma, \mathcal{A}) - \epsilon\} \tag{36}$$

Session Key Design: Practical Examples

Yield Optimization Agent

Permission	Value
Allowed actions	`deposit`, `withdraw`
Allowed contracts	Aave, Compound, Morpho
Spending limit	\$1,000/day
Expiration	30 days
NOT allowed	Transfer to external addresses

Security Properties

- **Containment:** max loss $\leq$ \$1,000 per day
- **Revocability:** owner can revoke instantly
- **Auditability:** all actions logged on-chain
- **Non-transferability:** key cannot delegate further

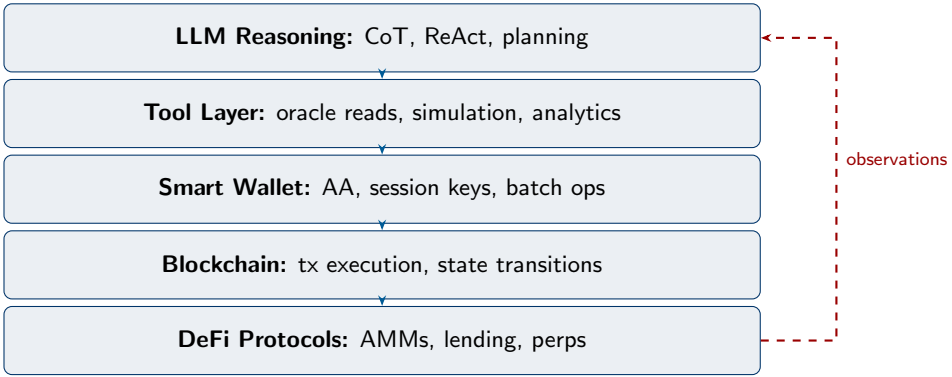
Attack Surface Analysis

- Agent compromise: max \$1K loss (contained)
- Key extraction: bounded by $L_{\max}$
- Replay attack: nonce prevents
- Privilege escalation: smart contract enforces

Synthesis & Research Frontiers

Open Problems · Regulatory Implications · Future

The Full Stack: From LLM Reasoning to On-Chain Execution



Each Layer Introduces Risk

LLM hallucination → tool misuse → wallet exploit → chain congestion → protocol bug. **End-to-end formal verification** remains an open problem.

Regulatory Implications of Autonomous Finance

Key Questions

1. **Liability:** Who is responsible when an AI agent causes market manipulation? The developer? The user? The agent?
2. **Fiduciary duty:** Does an AI agent owe fiduciary duty to its principal (wallet owner)?
3. **Market manipulation:** If 1000 agents independently front-run the same trade, is it coordinated manipulation?

Emerging Frameworks

- EU AI Act: high-risk classification for financial AI
- SEC guidance: algorithmic trading rules apply to agents
- MiCA: unclear on autonomous smart contract actors
- Proposal: "Agent registration" on-chain (identity layer)

Fundamental Challenge

Traditional regulation assumes **human actors**. AI agents are neither persons nor corporations. Legal frameworks need new categories.

Open Research Problems

Agent Theory

1. POMDP solutions for high-dimensional DeFi states
2. Sample-efficient RL for on-chain environments
3. Hierarchical agent architectures: optimal layer decomposition
4. Belief update under adversarial observations

Multi-Agent Systems

1. Nash equilibrium computation in continuous DeFi games
2. Price of anarchy in AMM-based markets
3. Mechanism design for MEV-resistant

LLM Oracles

1. Calibration improvement under distribution shift
2. Cost-optimal defense against info manipulation
3. Hybrid human-LLM oracle mechanisms
4. Cross-verification protocols

PhD Project Ideas

- Empirical: measure N_{eff} for LLM-based agents on Polymarket
- Theoretical: optimal session key design under bounded rationality
- Applied: multi-agent simulation of Uniswap

Key Equations: Day 7 Summary

Agent Models

$$b_{t+1}(s') = \frac{O(o|s',a)\sum_s T(s'|s,a)\, b_t(s)}{\mathbb{P}(o|b,a)} \qquad \text{(belief update)} \qquad (37)$$

$$V^*(b) = \max_a \left[R(b,a) + \gamma \sum_o \mathbb{P}(o|b,a)\, V^*(\tau(b,a,o)) \right] \qquad \text{(POMDP value)} \qquad (38)$$

Multi-Agent Game Theory

$$\text{BR}_i(s_{-i}) = \arg\max_{s_i} u_i(s_i, s_{-i}) \qquad \text{(best response)} \qquad (39)$$

$$N_{\text{eff}} = \frac{N}{1 + (N-1)\rho} \qquad \text{(effective diversity)} \qquad (40)$$

Security

Ethical Considerations: Autonomous Financial Agents

Access and Inequality

- Sophisticated agents available only to well-funded actors
- Retail users at informational disadvantage
- AI amplifies existing asymmetries
- "Arms race" dynamic: better agents $\Rightarrow$ more investment in agents

Transparency

- Should AI agents disclose their nature?
- On-chain: actions visible, reasoning hidden
- "Right to know" if counterparty is AI?

Autonomy and Control

- Human-in-the-loop vs. fully autonomous
- Kill switches and circuit breakers
- Graduated autonomy: limits scale with trust

The Alignment Problem in Finance

$$\max_{\pi} \mathbb{E}[u_{\text{user}}(\pi)] \quad \text{vs.} \quad \max_{\pi} \mathbb{E}[u_{\text{agent}}(\pi)] \tag{43}$$

When $u_{\text{user}} \neq u_{\text{agent}}$: the agent may pursue strategies that maximize *its* reward metric while harming the user's true objectives.

The Future of Autonomous Finance: Three Scenarios

Optimistic

- AI agents improve market efficiency
- Lower costs for retail
- 24/7 risk management
- Financial inclusion via agent-assisted DeFi
- Diverse agent ecosystem prevents fragility

Moderate

- Niche adoption for sophisticated strategies
- Regulatory guardrails constrain autonomy
- Hybrid human-AI operation
- Some efficiency gains, some new risks
- Gradual integration

Pessimistic

- Correlated agents cause flash crashes
- MEV extraction intensifies
- Adversarial manipulation widespread
- Regulatory ban on autonomous trading
- Concentration of power in AI operators

"The question is not whether AI agents will trade.
The question is whether we can design the rules they trade under."

Looking Ahead: Day 8

Tomorrow: Privacy, Zero-Knowledge Proofs, and Scaling

- ZK proofs: completeness, soundness, zero-knowledge
- zk-SNARKs and zk-STARKs: trusted setup vs. transparency
- Privacy-preserving DeFi: Tornado Cash, Zcash
- Layer-2 scaling: rollups, validity proofs
- ZK and regulation: selective disclosure

"Today we asked: can AI agents be trusted?
Tomorrow: can they be private AND accountable?"

Thank You

Questions, Discussion, Research Ideas

Prof. Dr. Jörg Osterrieder
joerg.osterrieder@...

References I

[1] Larry Eisenberg and Thomas H. Noe. “Systemic Risk in Financial Systems”. In: *Management Science* 47.2 (2001), pp. 236–249.

[2] Tim Roughgarden. “Transaction Fee Mechanism Design”. In: *ACM SIGecom Exchanges* 19.1 (2021).

[3] Timo Schick et al. *Toolformer: Language Models Can Teach Themselves to Use Tools*. 2024. arXiv: 2302.04761 [cs.CL].

[4] Jason Wei et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. 2022. arXiv: 2201.11903 [cs.CL].

[5] Shunyu Yao et al. *ReAct: Synergizing Reasoning and Acting in Language Models*. 2023. arXiv: 2210.03629 [cs.CL].